

# Dynamic Estimation Model for the early stages of a software project

Isabel Ramos Román  
Universidad de Sevilla  
Facultad de Informática y Estadística  
Avda. Reina Mercedes, s/n  
Sevilla - 41.012 (Spain)  
e-mail: isabel.ramos@lsi.us.es

Mercedes Ruiz Carreira  
Universidad de Cádiz  
Escuela Superior de Ingeniería  
C/ Chile, n°1  
Cádiz - 11.003 (Spain)  
e-mail: mercedes.ruiz@uca.es

## 1 Introduction

The publication, in 1991, of a dynamic model for managing software development projects by Abdel-Hamid and Madnick, lead to a new field which is providing a better understanding of the different variables to be considered, and the complex relations that are produced between them during the process of a software project. The multiple applications that dynamic simulations have and the new available simulation environments (like Estella, Vensim, Ithink, etc.) have lead to the opening of new working lines inside software development projects (SDP), where significant advances have not been produced in the past few years and where the traditionally used methods and tools were becoming obsolete for the complexity and magnitude of the actual software projects.

The objectives of this paper are, firstly, to justify the development of a Reduced Dynamic Model (RDM) which should be easy to learn and understand and can be used in initial project phases where the available or known information about the project is little. The second objective is to present the results provided by this RDM.

The RDM supports the same internal structure that Abdel-Hamid and Madnick's dynamic model and has been implemented in Vensim which is a flexible simulation environment as well as easy to use. We present a brief description of a traditional static cost estimation model like COCOMO and dynamic cost estimation models in order to compare them and obtain the gaps that should be covered with the development of new dynamic cost estimation models.

## 2 Cost Estimation Models

### 2.1 Static Cost Estimation Models

Among these models we would like to point out the ones most used, which are empirical univariable static estimations models. These models use empirically obtained formulas (from a small sample of projects) to predict the required data. These formulas are represented in the following equation:

$$\text{Resource} = a \cdot (\text{estimated feature})^b$$

where *Resource* can be the effort, development time, human resource level or technical documentation lines; *a* and *b* are empirical constants derived from the calibration of the model; "estimated feature" can be the source code lines, the function points or another previously estimated software pattern. Between the traditional static cost estimation models, COCOMO (actually COCOMO II) (COConstructive COSt MOdel), developed by Boehm in 1981, is considered as the most complete and used model. It has been formulated as a hierarchy of models. The following gives a brief summary of each one of the COCOMO models, detailing the aspects which are interesting for this paper:

- Basic COCOMO is a cost estimation model for estimating the effort of software projects from the product size in source code lines. This model is applicable in early project phases because of the little information it requires and follows a similar relation to type (1).
- Intermediate COCOMO calculates the effort from a similar relation to (1) multiplied by the EAF (Effort Adjust Factor). In order to obtain the EAF it is necessary to evaluate 15 cost drivers. This model can be used when we have not only the number of code lines but when we also have a better understanding about the project.

- Advanced COCOMO is a model that contains all the characteristics of the previous models but makes an evaluation of each of the cost drives over each phase of the project (analysis/design, coding and test). Therefore, it is applicable when you have enough specific information about each of the basic phases of the software project.

## 2.2 Dynamic Cost Estimation Models

The application of System Dynamics to SDP allows us to consider them as complex social-technological dynamic systems where the temporal evolution will also come determined by its internal structure as well as the relations established between the people who work in the project. This allows the development of multivariable dynamic models to describe the mental process followed by project managers in decision making. As we know, decision making has been traditionally based on the manager's experience. With the simulation of an available dynamic model, the project managers find the answers to their "What if ..." questions.

In order to simulate a dynamic model, it is necessary to know the initial estimations of the project and a series of initially defined values (parameters and function). These parameters and function help us to define the characteristics of the development project environment and of the environment of the organisation that develops the project.

## 2.3 Comparing Static and Dynamic Cost Estimation Models

Comparing static and dynamic cost estimation models has made us consider that actual dynamic models should be placed at the intermediate level proposed by Boehm. That is, actual dynamic models need to know an important number of project and organisation attributes before being simulated. In theTable 1, illustrates the equivalence we found between static and dynamic cost estimation models.

STATIC COST ESTIMATION MODELS (COCOMO)	DYNAMIC COST ESTIMATION MODELS
BASIC COCOMO	<b>RDM</b> (Reduced Dynamic Model)
INTERMEDIATE COCOMO	Abdel-Hamid and Madnick's, Draper Laboratory, SEPS, etc.
ADVANCED COCOMO	

*Table 1. Comparison between COCOMO models and actual dynamic models*

The corresponding level to advanced COCOMO has intentionally been left empty (Table 1) because we consider that there is actually no dynamic model that can support a set of dynamic models for each of the software process phases. The goal of a model of this kind should be double: on one hand, it should provide the capability of make a separate analysis of the singularities of each software process phase and its effects over the other project phases and, on the other hand, it should supply the linking of the results obtained from one phase to the following one in order to study their influence over the project phases that are left. The corresponding box to dynamic models which are applicable in the early stages when we have little project information but we need to have a rough idea of the fundamental variables evolution, will be occupied by the Reduced Dynamic Model (RDM). This is so because the RDM has applied a strong reduction to Abdel-Hamid and Madnick's model so that the amount of information required to simulate the RDM is approximately half the amount necessary to simulate the Abdel-Hamid and Madnick's one (hereinafter called the Extended Model).

### 3 Reduced Dynamic Model (RDM)

The two principal factors that lead us to develop the RDM were:

- The existing hierarchy of estimation models such as COCOMO (static and univariable) gave us the possibility of creating a similar hierarchy of dynamic estimation models (multivariables) depending on the knowledge we have about the project.
- The requirement of knowing an important number of project and process attributes in order to simulate the dynamic models (over 26 functions and another 25 parameters in the Extended Model). Because these parameters and functions are difficult to obtain in the beginning of the project.

Moreover, the development of a reduced model is also justified by the existence of a common causal diagram for R&D and software projects and the creation of reduced models from other extended models in other known areas employing the work of Eberlein on dynamic models simplification. The simplification proposed by Eberlein consists in obtaining a reduced model from an extended one, eliminating the feedback that are not considered essential for the behaviour wanted to be analysed. To be brief, the development of the RDM has the objectives of obtaining a simplified dynamic model with a minimum number of function and parameters, the basic feedback to model a small or medium SDP and which is able to generate the behaviour of the Extended Model. The principal application of this RDM is placed in the early estimations when the information and knowledge over the project is little.

The main reduction assumptions that have been employed are basically the following three:

- No difference between expert and non expert personnel. Both types of personnel have been included as a single level variable called *Personnel*, which contains the total number of personnel working in the project.
- No difference between active and passive errors. Both kinds of errors have been included as a single variable called *Errors*.
- Being that one of the measures of the project size comes with the number of tasks to know the progress of the project, we consider that in the early stages of the project it is very important to know, in each moment, the work done (tasks finished). This must be done without distinguishing between the tasks developed and the tasks tested. So, in the RDM two types of tasks “do tasks” and “check these tasks” have been considered. The latter has been the one which has had most influence in the reduction of the Extended Model. Employing the former assumptions, the size of the RDM is approximately the half of the Extended Model size (see Table 2).

	RDM	Extended Model	% of Reduction
Variables	67	138	more than 50
Functions	12	25	more than 50
Tables	16	26	38,5
Total	95	189	50

Table 2. Obtained reductions of the RDM compared to the Extended Model

In order to validate the RDM, the Example Case from Abdel-Hamid and Madnick and a real project have been used, verifying that the RDM is able to reproduce the same evolution patterns as the Extended Model

### 4 Conclusions

The RDM is an useful tool which can also be used to make estimations and analyse the SDP behaviour, especially in the following cases:

- When there is little information about the SDP, which is usually in the first client contacts.
- When the development organisational entity does not have a historical project base or this has not been completed enough to define an important number of project and development process attributes in early project phases.