Proceedings of the 5th WSEAS/IASME Int. Conf. on SYSTEMS THEORY and SCIENTIFIC COMPUTATION, Malta, September 15-17, 2005 (pp289-293)

Software project cost estimation using AI techniques

Rodríguez Montequín, V.; Villanueva Balsera, J.; Alba González, C.; Martínez Huerta, G.
Project Management Area
University of Oviedo
C/Independencia 4, 33004 Oviedo
SPAIN
http://www.api.uniovi.es

Abstract: - The software cost estimation is an important task within projects. It determines the success or failure of a project. In order to improve the estimation, it is very important to identify and study the most relevant factors and variables. This paper describes a method to perform this estimation based on AI techniques and using Data Mining methodologies.

Key-Words: - Software Cost Estimation, Artificial Intelligent, Data Mining

1 Introduction

The software cost estimation for Information Systems is process used by an organization in order to forecast the cost for the development of a software project. The estimation of resources and time needed is very important for all the projects, but specially within Information Systems, where budget and schedule are usually overcome.

All the estimation methods have to take as reference a software size metric. [1][2][3].

The software project estimation present special difficulties, compared with other sectors. The existing methods are highly dependant of the available information for the project. When the project steps, the estimation is more accurate because there is more information and it is more reliable. The estimation process should be a continuous process, including the new information.

This work establish a method for software cost estimation based on Artificial Intelligent (AI) techniques, identifying the most influencing factors and variables over the software cost. The work is done based on a historical dataset of projects, taking the data provided by the International Software Benchmarking Standards Group (ISBSG), gathering information from more than 2000 projects. This dataset contains numerical values as well as categorical data. Within this dataset there are a high percentage of missing values. Due to this, the data mining techniques are used for preprocessing the information.

2 Problem Formulation

Usually, the process for effort estimation within Information Systems has been noted as cost estimation, although cost is just only the result derived from the estimation of size, effort and schedule. The size estimation is the measuring of the project size, usually in lines of code or equivalent.

Since software is a product without physical presence and the main cost is the design and development of the product, the cost is dominated by the cost of the human resources, measuring this effort in man-months. Finally, the schedule estimation is the amount of time needed to accomplish the estimated effort, considering the organizational restrictions and the parallelism between project tasks. At the end of the process, we can get an economical value for the project cost, multiplying the number of man-month estimated by unitary cost. So the project estimation is a forecast of the expected effort to develop a project and the scheduled needed to accomplish it.

Because the complexity and variety of factors influencing over the accuracy of the effort estimation, we need to develop analytical models that take consideration of every factor.

The base of the software cost estimation was established by Lawrence H. Putnam and Ann Fitzsimmons [4], although the first approaches were carried out during the sixties. The more important progresses were performed within the big companies of the epoch. So, Frank Freiman, from RCA, developed the concept of parametric estimation with his tool named PRICE. Norman Peter, from IBM, developed a model based on adjusted curves [5].

During the seventies the number of software projects and its size suffered a big increment. Most projects performed during this epoch failed. Due to this, more people focused on project estimation. Using statistic techniques (mainly correlations), people researched about the factors influencing over project effort. In this way, the more emblematic model, COCOMO, was developed by Barry W. Boehm [6]. Most of these models consider the effort (E) as result of an equation based on:

$$E = a \cdot S^b \quad (1)$$

Where E is the effort, S is the project size in code lines, a reflects the productivity and b is an scale economy factor. The result is adjusted with a set of drivers representing the development environment and the project features (15 drivers for COCOMO model).

The main issue of these models (and even the main issue of present models) is considering the size as a free variable, when the size is unknown until the end of the project. The size must be estimated before the project start. During this epoch, Albreacht and Gaffney [7][8] replaced the lines of code by the Function Points (FP) as unit for measuring the project size. The Function Points measure the size of the software independently of the technology and the language used to code the programs. It involves the change from the size oriented metrics to the functionality oriented metrics. The productivity of develop will be countered as FP per man-month.

During the seventies Putnam [9] developed other popular model, SLIM, based on Rayleigh curve, adjusted using data from 50 projects.

The eighties is a transition period and the best methods (like COCOMO and SLIM) are consolidated. Caper Jones [10] improved the Function Points method to consider complex algorithms, and KPMG developed the MARK II [11], other improvement method for measuring FP.

In the nineties, Boehm developed a new version of COCOMO, named COCOMO 2.0 [12], adapted to the new circumstances of the software (object oriented, transactions, software reusing, etc.) Until the nineties, most of the improvement efforts were address to disaggregate the components of the models and proceed to adjust the parameters using regressions. Other approaches were also used, in example rule systems were used by Mukhopadhyay, [13], or decision trees by Porter [14][15]. But the results were not satisfactory and the application of these techniques presented some problems. With the explosion of the AI techniques in the beginning of the nineties, new approaches were used: Fuzzy Logic, Genetic Algorithms, Neural Networks and so on. The new modelling techniques allow a most suitable selection of variables and the study and work with more representative datasets. Additionally, the use of these techniques is useful combining the knowledge of the domain (those information we have about the problem) with the processing of large data information. But this links with other of the existing problems, the lack of reliable datasets. Using these techniques, we can analyze large quantity of information. Due to this, during the last years there were some tries to establish repositories with information about software projects. One of this approaches were performed by the ISBSG [16], the dataset used in this work. This repository contains information about:

- Size metrics
- Efforts
- Data quality
- Type and quality of the product: information relative to the development, the platform, the language, the type of application, organization, number of defects, etc.
- CASE tools utilization
- Team size and characteristics
- Schedule information
- Effort ratios

Although the existing methods have improved significantly the way in which estimation is performed, they don't reach the required accuracy. The limitations of the existing models are derived from the difficult to quantify the factors, as well as the simplifications done in the models. The datasets used to adjust the models shall be representative. Finally, considering the non-linear of the process and the dependencies of non quantified parameters, the problem is suitable to be studied under the framework of the AI techniques.

4 Technical and Methodology

Data Mining techniques can help in data analysis, modelling and optimization. The software estimation process is influenced by a lot of variables. In order to get a successful model a work methodology must be use for Data Mining projects. CRISP-DM [1] is one of the most usual process models. It divides life cycle for Data Mining projects in six phases.

The methodology CRISP-DM [8] constructs the cycle of life of a project of data mining in six phases, which interact between them on iterative form during the development of the project

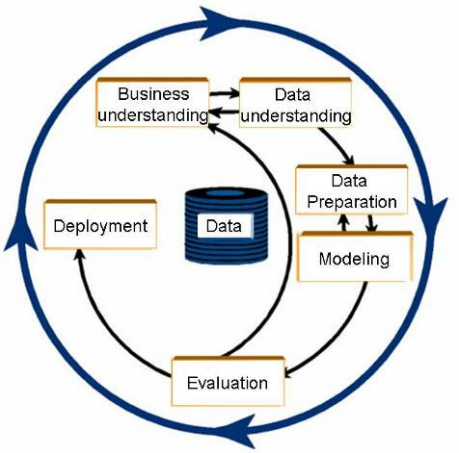

Fig.1 Phases of the Modelling process of methodology CRISP-DM.

The first phase, business understanding is an analysis of the problem, includes the understanding of the objectives and requirements of the project from a managerial perspective, in order to turn them into technical objectives and plans.
The second phase, data understanding is an analysis of data includes the initial compilation of information, to establish the first contact with the problem, identifying the quality of the information and establishing the most evident relations that allow establishing the first hypotheses.
Once, realized the analysis of information, the methodology establishes that one proceeds to the data preparation, in such a way that they could be treated by the modelling technologies. The preparation of information includes the general tasks of data selection to which the modelling technology is going to be applied (variables and samples), data cleanliness, generation of additional variables, integration of different data origins and format changes.
The phase of data preparation, it is more related to the modelling phase, since depending on the modelling technology that is going to be used, the data need to be processed in different forms. Therefore the phases of preparation and modelling interact between then.
In the modelling phase the technologies more adapted for the specific project of data mining are selected. Before proceeding to data modelling, it must establish a design of the evaluation method of the models, which allows establishing the confidence degree of the models. Once realized these generic tasks one proceeds to the generation and evaluation of the model. The parameters used in the generation of the model depend on the data characteristics.
In the evaluation phase, the model is evaluated in that degree they are fulfilled of the success criteria of the problem. If the generated model is valid depending on the success criteria established in the first phase, one proceeds to the development of the model.
Normally the projects of data mining do not end in the model implantation but it is necessary to document and present the results of an understandable way to achieve an increase of the knowledge. In addition, in the development phase it is necessary to assure the maintenance of the application and the possible diffusion of the results [3].

5 Modeling Method

Following the steps of the methodology, the data acquisition is realized, organized.
To begin the analysis of the data set, it get the historical set that there has provided ISBSG (International Software Benchmarking Standards Group).
Later it proceeds to make a data exploration and a monitoring of the quality. For which there are realized statistical basic technologies, to find the data properties. Since, given that there are more categorical variables we proceed to realize histograms with the occurrence frequencies.
In this point starts the data preparation phase. This phase has been very costly due to there are more missing values, on which there has been analyzed the use of diverse technologies to predict or to delete this hollow in the information.

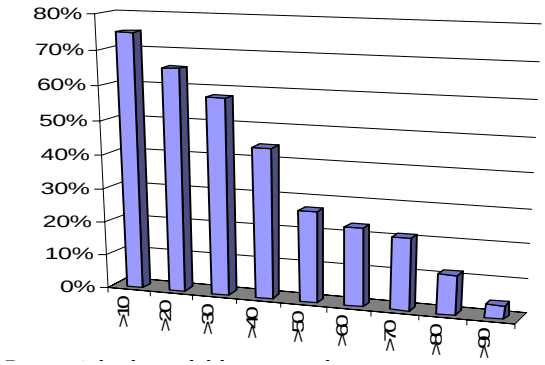

Fig. 2 Percentage of variables with missing values

Studies have been realized to verify if this missing values has some type of influence in the effort (person per time) needed to realize the project, this is the objective variable that has been identified as model goal.
Other one of the problems is the great presence of categorical variables of difficult processing for someone modeling methods.

They have been measured different technologies for the per-process and transformation of these variables. When the number of classes was reduced (lower than six), the process to the categorical information has created so many variables as classes. This way for example, if the categorical variable "platform of development", it was containing the values MR, MF and PC, then 3 variables have created like (1,0,0) if the value of the variable is MR, (0,1,0) if it is MF and (0,0,1) if it is PC.

When the number of classes of a variable was very high (Superior for six) has been transformed the value of the category directly to a numerical value.

For the process of the missing information has chosen to select a robust technology that allows the processed of this type of information, such as nets SOM, MARS [2] and MART [4].

For the results evaluation of the information has been cut in three separated sets from random way: one of them that contains 75 % of the data that it has been send for the construction of the model, 10 % for the model test and selection of the best model. The results have been tested by 15 % of remaining data.

Once has been generated the model it is possible observe that the variable that contributes with more information to the effort estimation, in this model, is the maximum team size. It is also important the Function Points and the value of adjustment factor.

Another important factor is the development platform used and the type of language that is used in the programming, it is relevant that the missing values give knowledge to the effort estimation model.

The model also considers if an adaptation of the code has been made, if planning has been used, as well as other variables related to the metric one used and the implication of the resources.

The relative importance of each variable in that model is analyzed, the variables that more importance contributes to the model are selected and they are added according to his importance while they improve the results.

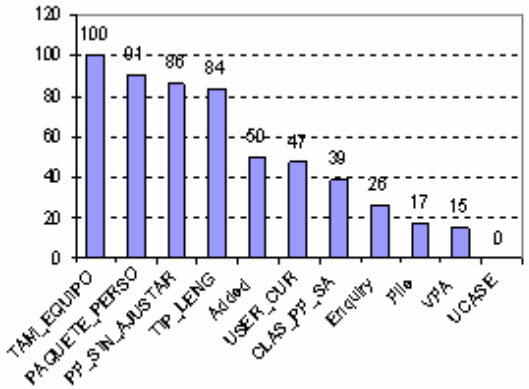

Fig 3 Relative importance of the model parameters

In the previous figure the relative importance of the variables of the best model can be seen since it has commented previously.

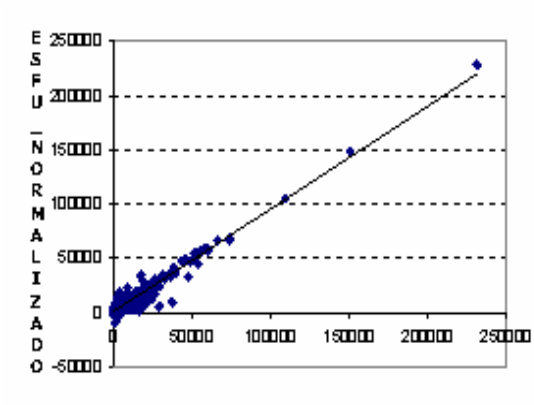

Fig.4 Real effort front of estimate effort

For the construction of model MARS it has been used the following parameters, interrelation between variables at level 3, base functions of second degree. The results are in the following table.

Absolute error	% old	% train success	% test success
396	22%	24%	15%
792	30%	45%	38%
1189	33%	58%	52%
1585	41%	67%	61%
1981	44%	73%	69%
2377	48%	80%	72%
3960	56%	83%	76%

Table 1 Model results

This is a significant improvement with respect to the reference old model that is a model based on analogies.

4 Conclusion

All the methodologies of project management make management of plan and costs in any type of project and in the projects of software.

The chosen system to make the estimations has to have the confidence of the project management and to allow to adapt again to the changing necessities of the software. The historical data summary in the end of the project is essential to update the data base of projects and so that the system can fit its parameters to the changing conditions of software.

References:

[1] Fenton, N. y Pfleeger, S.L. , *Software Metrics, A Rigurous & Practical Approch*. PWS Publishing Company 1997.

[2] Kitchenahm, B., Pfleeger, S.L. y Fenton, N.E., *Towards a framework for software measurement validation.* IEEE Transactions on software engineering, vol. 21, nº 12, 1995, pp. 929-944.

[3] Minguet Melián J.M. y Hernández Ballesteros J.F. , *La Calidad del Software y su Medida*. Centro de estudios Ramón Areces, S.A. 2003.

[4] Putnam LH, Ann Fitzsimmons. Estimating software cost, Datamation; 1979.

[5] Norden Peter V. Curve fitting for a model of applied research and development scheduling. IBM J Res Develop 1958;2(3).

[6] Barry W. Boehm, Software Engineering Economics, Prentice Hall PTR Prentice-Hall Inc., 1981.

[7] Albrecht, Allan J., "Measuring Application Development Productivity," Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Oct. 1417, 1979.

[8] Albrecht, Allan J., and John E. Gaftney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," IEEE Transactions on Software Engineering, Vol. 9, No. 2, November 1983.

[9] Putnam LH, Ann Fitzsimmons. Estimating software cost, Datamation; 1979.

[10] Jones, Capers, "The SPR Feature Point Method," Software Productivity Research, Inc., 1986.

[11] Symons, Charles, "Software Sizing Estimating: Mark II FPA," Wiley, 1991.

[12] Boehm BW, Abts C, Clark B, Devnani-Chulani S. COCOMO II model definition manual, The University of Southern California; 1997.

[13] T. Mukhopadhyay, S.S. Vicinanza, and M.J. Prietula, “Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation,” MIS Quarterly, vol. 16, pp. 155-171, June, 1992.

[14] A. Porter and R. Selby, “Empirically Guided Software Development Using Metric-Based Classification Trees,” IEEE Software, no. 7, pp. 46-54, 1990.

[15] A. Porter and R. Selby, “Evaluating Techniques for Generating Metric-Based Classification Trees,” J. Systems Software, vol. 12, pp. 209-218, 1990.

[16] ISBSG: International Software Benchmarking Standards Group. http://www.isbsg.org