Data Mining Applied to the Improvement of Project Management

Joaquin Villanueva Balsera, Vicente Rodriguez Montequin, Francisco Ortega Fernandez and Carlos Alba González-Fanjul

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/46830

1. Introduction

Most of the professional activities are developed as projects. Project management is a complex process involving many elements interrelated and dependent on external agents that may complicate its control. Projects, as defined by PMBOK of Project Management Institute (PMI, 2009), are designed to solve a problem or need, have a temporary effect and is unique in time and not repeatable in the same circumstances. Uncertainty is a key factor associated with project management. This factor affects the consumption of resources, the estimation of time and money as well as the impact of risk and quality.

In fact, risks, uncertainty or estimation are the key words to pursue for a project-oriented organization. And its complexity and difficulty are so significant that the deliveries are clearly unacceptable. CHAOS and other independent reports show success rates under 32% with a deviation in time and cost several times higher than initial estimation (Standish Group, 2010).

Traditional project management, regardless of sector, identifies a series of phases such as:

1. Initiation, where needs are identified and evaluated to know whether it is possible to carry out the project at this stage. Uncertainty is very high due to lack of accurate information, which means that the possibility of error in the estimate is high.
2. Planning, this aims to develop a solution in greater detail, by breaking down the problem into more detailed activities. This reduces uncertainty and makes estimates and forecasts. You must also define the tasks and calendar and estimate the time and money needed to undertake the project.
3. Execution. Once tasks are clearly defined, the implementation phase of the project can begin with the use of monitoring techniques and adjustments to planning, in order to

maintain control on the project. At this stage reducing uncertainty is critical, the risk of an incorrect estimate and the impact of this is much higher because there is no time to solve the deviations.

4. Closure. Finally there is the closing stage of the project in which results are checked to determine if the project satisfies the needs for which arose, as well as collecting information on the problems detected, weakness or strength of the team. This is called lessons learned and has to be a source of information that is stored to be the basis on which decisions are made in future projects.

Project managers have to deal with those problems with a limited set of tools but it has been proven that better estimation levels at any stage and correct post-mortem analysis are the most influencing techniques for a continuous improvement. And that is only possible using analysis of data from previous projects. During the development of projects, very different sources of data can provide information about what is happening, including delays, overloads, etc. That data could be used for immediate analysis and correction but it can be extraordinarily useful for a global better performance of the rest of projects in which the organization is involved. A structured repository of the data will be an optimal source of key information for future success. The dataset is a snapshot that defines the behavior of the portfolio to be used for postmortem analysis to analyze trends and generate models that define the behavior of certain critical factors in the projects as estimating the expected risk or effort.

The information to be collected must come from every phase in the project: initiation, planning, execution and closure. Some problems may arise in the data collection phase since every project is unique by definition, therefore the types of data, fields or indicators to be stored may be different depending on the project, thus generating a very heterogeneous but less constant data set.

However, the phase which will benefit more from the implementation of data mining techniques is the initial planning phase. Since at this stage there is not much detailed information on the outcome of the project, the project manager may make bigger mistakes in the estimations about costs, efforts, time or risk probability.

All these issues are common to every type of projects; however they have a greater impact on software projects, since they possess several components which influence directly on estimation problems. These projects are developed in a constantly-changing technological environment. Cost estimation is linked directly to resources effort, which entails that proper budget estimation depends on the effort estimation measured in person-hour necessary to develop a product. Another determining factor in software projects is meeting the requirements with a series of quality assurances, to avoid nonconformities or defects.

Unfortunately those data cannot be easily processed manually due to their heterogeneity and that is the perfect environment for Data Mining, which performs the processing of information extraction from raw data in order to extract useful conclusions. Data mining is capable of examining the data creating rules or conclusions providing the organization with tools devoted to decrease the risks and uncertainty in the decision making process.

Data mining can be helpful in all stages and fields: estimating better costs, optimizing the bids, evaluating the risks, decreasing the uncertainty in the duration of tasks, etc.

The chapter presents in a learn-by examples way how data mining is contributing to improve the field of project management, including also guides and tips about how to approach your own problem.

When the organization has not enough information from its own data it can collect information from external databases. Currently there are databases that collect information from the project closure. For the analysis of these cases we used International Software Benchmarking Standards Group (ISBSG, 2012), which is a dataset that includes data compiled from information extracted from the implementation of projects of different nationalities and sizes.

This dataset consists of information systems projects. Such projects are characterized by the deliverable is a product with no physical existence itself and whose main cost lies in the development or design (not materials), it is logical to assume the cost of production is dominated by personnel costs, measured so their effort in person-months or man-hours. Here it highlights the importance of a good estimate of the effort since it is the main cost factor.

One of the most important objectives in the management of information technology projects has been focused on the correct estimation of effort and risk detection of defects or quality nonconformities in the implementation of the system.

There are many empirical models, which are conventional methods that are commonly used, based on a formula that fits the physical behavior of this attribute, but information systems evolve very quickly and are much better to have a method that allows us to adjust the model to the way we work and the technological environment for projects that develop our organization.

This need arises to model for effort and the risk impact measured as defects in the project. Procedure must be considered as a data mining project, for which it has been followed the phases given by CRISP-DM methodology (Chapman *et al.*, 2000). The major steps to be followed begin by data understanding; this stage is a study of available data. The next step is to perform the data preparation to modeling, since each technique needs a particular type of data.

After the application of the data mining analysis described, it can be obtained a reduction of uncertainty implicit in the projects, particularly in information technology systems. In addition, you can get more results as, for instance, relationships between attributes, identification of variables that provide information or are more meaningful to estimate the effort or to identify the level of risk o quality assurance.

2. Understanding the software project environment

As a case study, it has been chosen the information technology projects and it has been considered the need to identify, from the initial phase of a project, the estimation of effort and risk in a software project taking as reference a dataset compiled from previous projects within an organization. Therefore, first it must be learned the generic and common process of effort estimation and the significance of classification of potential risk of product development.

Ever since the beginning of information technology projects was first developed back in 1968, serious problems have been detected when it comes to fulfill the main premises of a project: meet the quality goals within a predefined time and cost, as well as it occurs in any other sector. This is commonly known as "software crisis", and it is defined as the existing complexity in building error-free programs. The reasons that provoke a software crisis may be, among others, the complexity of programming and the constant process of changing that an application must endure in order to be adapted to users' requirements.

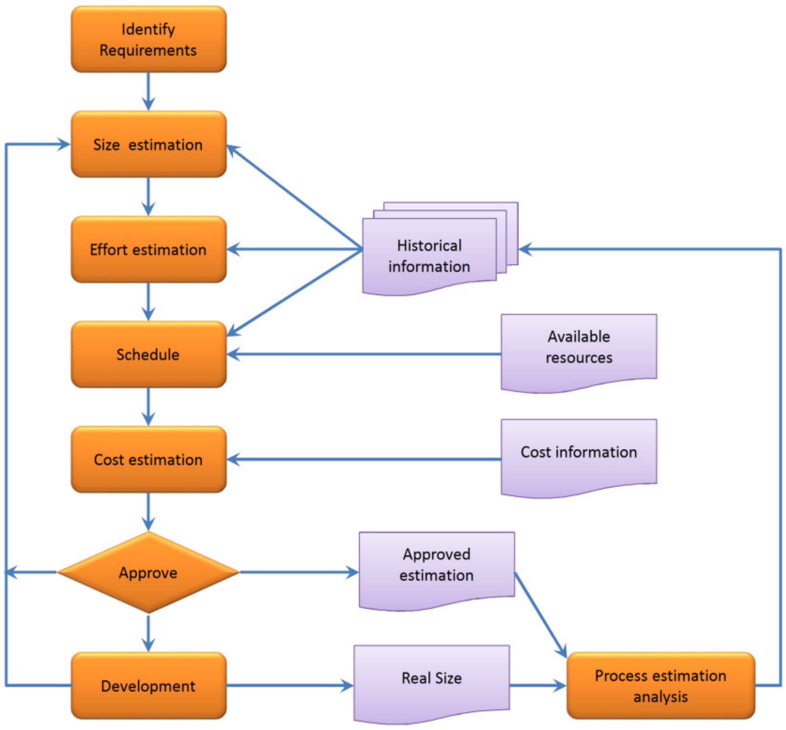

Figure 1. General stages in effort estimation.

Effort estimation is characterized by its complexity and its lack of reliability and high failure levels. (Lock, 2007), given that:

- Estimations *per se* require a great amount of time effort,
- Estimations are hastily performed,
- Previous experience is required,
- Experts' estimations are subjective,
- Information from previous projects is not stored as it is not time and cost efficient,

- Estimations are adjusted to available resources, it is known as "Parkinson's law"(Parkinson, 1955),
- Other characteristics of these projects are: the difficulty in finding similar projects, vague and changing requirements, etc.

The four basic steps to perform a more plausible estimation of a information technology project (Chemuturi, 2009) are the following ones:

1. Estimate the product development size.
2. Estimate effort in person-months.
3. Estimate total time.
4. Estimate final cost.

The first step in any estimation is the understanding and defining of the system to estimate. Software projects are intangible, thus the difficulty in understanding a product which cannot be seen or touched. Size estimation can be based on source lines of code or function points. (Albrecht & Gaffney, 1983). In order to estimate the schedule with resources, there are a series of rules depending on the type of project to perform, and they may be of help in a first estimation, although this process keeps improving as it is adjusted to a more detailed schedule. From this schedule, a distribution of necessary resources is extracted, which will have to be adjusted to the available resources to, eventually, provide the final schedule.

In this phase all the aspects of an initial estimation are considered, even calculating the project cost in terms of effort. Nevertheless, this value clearly will not be the final effort of the project, since there will be a series of efforts or costs to add, which stem from the needs and interests of the organization, such as general costs, profits, risks, technological environment, etc.

In 1986, Alfred Spector, president of Transarc Corporation and vice president of research and special initiatives at Google since 2007, published a paper comparing bridge building with software development; he reasoned that unlike the software, the bridges are built on budget, on time and do not fall, and if something is wrong it must analyzed why it happened (Spector & Gifford, 1986). The reason for such difference is an extremely detailed design. The design of a bridge stays permanent and admits no alteration; therefore the building contractor has little or no chance at all to modify the specifications. Another difference lies in the fact that, when a bridge collapses, causes are investigated and gathered as feedback to apply in future constructions, whereas in software development failures are hidden, so the benefits of lessons learned are not obtained.

After this first approach, a more rigorous study of problems and solutions was undergone. A number of institutions emerged, producing reports and statistical analysis, such as GAO (Government Account Office) – analyzing software development projects for the American Government, or ESPITI, studying the main problems of software development on European scale, which results are quite similar to the obtained in one of the more accepted reports, CHAOS (Standish Group, 1995), which indicates that most of the problems are related to specifications, management and documentation of projects.

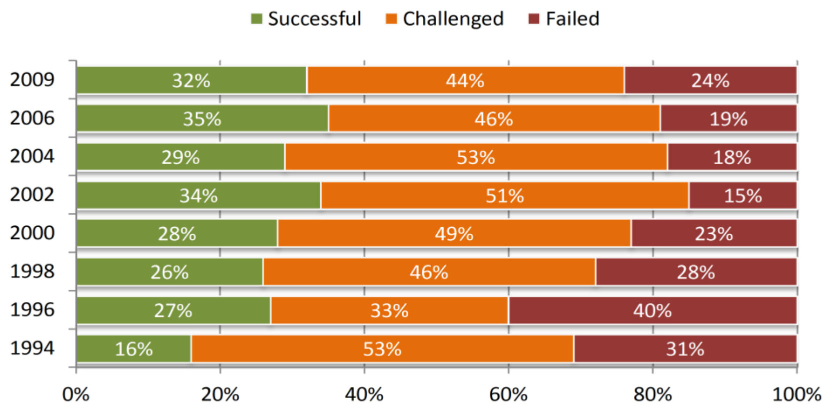

Figure 2. Evolution of the Software Project success. Source CHAOS (Standish Group, 2010)

The report shows that in 2009 software projects were 32% successful, thus a reduction as compared to 2006. 24% of the rest of projects have failed, understanding "fail" as cancelled or never used, and 44% of projects were changed, that is, delayed, over-budgeted or with less functionalities than the required.

From the previous considerations it is inferred that the complexity and variety of agents that affect correct effort estimation, illustrate the need to develop analytic methods which consider every agent which influence the development of the project and product, with specifications to be applied in an organization.

Besides, knowing, preventing and trying to mitigate any risk that a software project may be subjected to is a complex task, mainly as a result of the large amount of threats of diverse nature which are present since the beginning of the project.

The estimation of level risk is another key factor. Risk is reflected upon every main target of a project: time, cost and quality. Regarding time and cost, one could perform an assessment of the grade of certainty with which the models provide their previsions and, as a consequence, extend this time and cost. As regards the quality variable, specific models can be developed using similar data mining as the one used for time and cost estimation but to predict this resulting quality, which in the case of software projects it is usually related to the number of defects. Therefore, risk level can be estimated through the prediction of defects to be produced in the project.

Making a model or knowing the agents influencing the risk, as well as their severity, will allow the managers of software projects to facilitate project management starting process.

So far, the models have improved the effort estimation process; however they have not met the demands of software projects.

Given the high nonlinearity of the process and its dependence on non-quantifiable parameters, the case study is framed within the problems in which the application of intelligent data mining techniques has given successful results, with the introduction of significant control advances.

Therefore, when it comes to developing the models, the main target will be the estimation of risk and effort needed to develop the product, taking the available information on size, resources and other variables that may affect the development environment as a starting point.

3. Related works and assess situation

Software estimation process foundations were first established by Putnam (Putnam LH & Ann Fitzsimmons, 1979), who also developed another reputed method, Software Lifecycle Model (SLIM) (Putnam, 1978), based on Rayleigh curve adjusted to empiric results.

The most representative model of effort estimation is COCOMO (B. W. Boehm, 1981), developed by Barry W. Boehm. These types of models have their estimation based on a general equation as:

$$E=a\cdot S^{b} \tag{1}$$

where *E* stands for *effort*, *S* stands for the software size measured in source lines of code or function points, *a* and *b* are factors adjusted according to the type of project. One of the problems of these methods is their use of parameters such as source lines of code. It is quite complex to estimate the values of those variables before the beginning of the project, therefore, these models are difficult to apply before considering the requirements and design of the project. Afterwards, this method was revised to be adapted to the new software characteristics (B. Boehm *et al.*, 1995).

One of the key advances was undergone by Allan Albrecht and John Gaffney, who developed the Function Point Analysis (FPA) (Albrecht & Gaffney, 1983) as a method to estimate the functional size of an information system software.

This method meant a substantial progress if compared to the traditional kilo-Source lines of code (KSLOC); it abandoned size-oriented metrics, to start using functionality-oriented metrics. First, the system components are identified, quantified and classified as inputs, outputs, inquiries, logical internal files and external interface files. Once the unadjusted function points are calculated using a weighting table, it is adjusted with a complexity adjustment factor. This factor is calculated by the weighting of the influence of environmental agents on the project, which is: data communications, distributed processing, complex processing, end-user and configuration load among others. The development team productivity is then calculated in a more aseptic way, by using Function Points per Person-Month.

Different surveys as (B. Boehm *et al.*, 2000), compile the different traditional software estimation techniques.

Recently, it has been noted a trend to use computational intelligence techniques based on data to project management problems and these techniques were proved able to detect complex relationships. Classic approaches, which have been shown above, involve analytical or statistical equations (B. Boehm *et al.*, 2000). Usually, intelligent models utilize neural networks (Tadayon, 2005), fuzzy logic (Xu & Khoshgoftaar, 2004), tree decision (Andreou & Papatheocharous, 2008) and evolutionary algorithms (Dolado, 2001) for performing improved effort estimations.

In addition, research studies propose various data mining techniques to extract knowledge. Authors such as,(Briand *et al.*, 1992) utilized regression-based models combined with classification trees techniques applied on historical datasets in order to estimate the project cost.

Other techniques have used the concept of fuzzy logic; it was used to integrate the concept of risk uncertainty using fuzzy decision trees (Huang *et al.*, n.d.).

4. Application process and guidelines

Given the case of study is presented as a data mining application process, it has been considered the use of one of the more wide-spread methodologies: Cross Industry Standard Process for Data Mining CRISP-DM (Chapman *et al.*, 2000). This has already been used in order to solve similar problems (Montequín *et al.*, 2005).

This methodology defines the data mining life cycle process; it consists of 6 phases, this is a global process which is performed by process iterations, in addition, the phases interact with each other throughout the development process.

The initial phase is defined as Business Understanding and aims to identify the objective, which will be defined from a business perspective, which also has to assess the situation and design a plan of data mining project.

The next step is defined as Data Understanding and its aims are to collect and review data; this begins with an initial dataset that is processed to get familiar with the data, performing the first contact with the problem, discovering data quality problems, identifying the first hypothesis and defining initial relationships.

When the Understanding step is complete, CRISP-DM proposes a new step for preparing data for subsequent modeling. The data preparation phase has all the activities necessary for building the final dataset, its goal is to select and clean the data. This phase can be performed several times. This task includes the selection of rows and attributes and data cleaning to conform to requirements of used modeling tools. It should be borne in mind that each modeling technique requires a particular data type or a preparation adapted to its needs. Therefore it has to perform transformations on the attributes, such as converting numerical values to nominal or otherwise, processing missing values, identifying outliers, reducing the size of variables or samples, etc. This phase is closely related to the following modeling and there is much interaction between them

The next phase is the modeling, at this stage the modeling technique that best fits to study requirements should be selected and its parameters are calibrated to optimal value. Therefore, stepping back to the data preparation stage could be often necessary.

After the modeling phase, it has to perform the evaluation phase. The confidence degree that ensure as valid model has been set from the beginning. It must determine whether the business issue has been sufficiently resolved.

A data mining study is not completed in the evaluation phase, but it has to continue with a deployment plan and subsequent monitoring and maintenance of model results.

This whole process is iterative, since it generates a cycle that is repeated until the criteria of success is met, i.e. if the objectives are not met at the evaluation phase, it has to do another cycle, for which you have to develop a new data set or to define new initial objectives.

In the methodology phase, targets and strategies of data collection must be fixed. It has been chosen to use an existing dataset, whose collection is approved by an international prestigious organization as ISBSG, International Software Benchmarking Standards Group. Release 10 has been used as case of study.

5. Application case

To achieve the fixed objectives and analyzing the available attributes of the dataset, it is considered as target a set of models which predict the development effort and three predictive models capable of estimate the number of potential defects that will be included in the information system to be developed (minor, major and extreme). Once the models are developed, one can analyze which parameters provide more information and are more sensitive to changes in the problem to be predicted.

Once the targets are defined, the following phases of the methodology are applied, which entails data examination.

5.1. Data understanding

This section has examined the data to identify quality and relationships that define a first hypothesis. ISBSG database contains information recorded on the projects, which is what an organization wants to record its own repository.

The dataset is formed by variables such as:

- Rating (associated with data quality as rated by ISBSG organization),
- Size measurement metrics (Function points, SLOC: Source Lines of Code),
- Product environment and architecture (development, documents, language, hardware, operating system, methodology),
- Quality (number of minor, major and extreme defects).

The amount of information that will be used is 4106 projects and 106 attributes. Data is collected from projects that come from different countries, as well as a large number of different types of organizations. It can be seen in the attribute "Organization type" that has projects in the following sectors: Engineering (38%), Services (21%), Insurance (17%), Public (12%) and Banking (10%).

In order to analyze the data quality for modeling, some basic statistics are performed to identify ranges in which numeric variables move and the dispersion of the cases of categorical variables.

After an initial analysis of the dataset, the first observed characteristics are that there are a lot of categorical variables, which complicates their use by some methods that do not support these types of variables.

As projects are heterogeneous, the cases do not collect information on all attributes, therefore there are plenty of missing values (Acock, 2005).

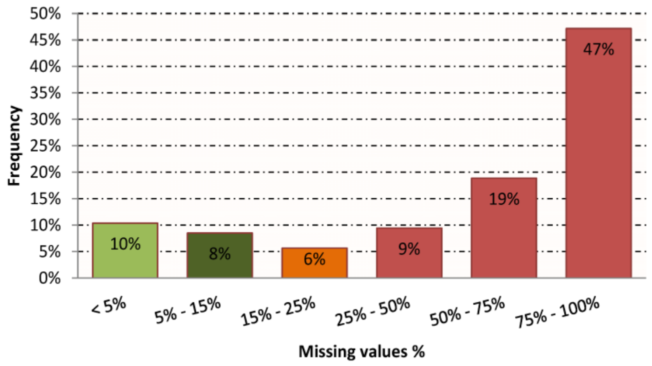

Figure 3. Frequency of missing values in dataset

Only 10% of the attributes have an amount of missing values below 5%; depending on the importance of some of the variables, it could be used up to 18% of the attributes of the dataset. However, as the figure shows, more than 75% of the dataset variables have missed over 75% of information, this means that, at the most, only 25% of the projects would have information available in these variables, which implies a great reduction of the dataset to be used for modeling. As a result, over 80% of the attributes of the dataset would not be used for modeling.

In order to identify the first hypothesis and check the consistency of the data, linear correlations have been searched.

As regards the attributes that are the targets for this study, you can choose between the three variables that reflect effort and productivity (as measured by effort per functional size) which have a strong linear relationship, so the study will take one of each as a representative. There is also a relationship between "Summary Work Effort" and variables that represent the breakdown effort into phases, especially between Specify, Build and Test phases. In the figure, one also can observe a direct linear relationship between planning and design phase, and between specify and test phase. It follows a trend that projects with much effort in the planning phase also make great efforts in the design phase, and those with great effort in the specify phase, do also have it in the test phase.

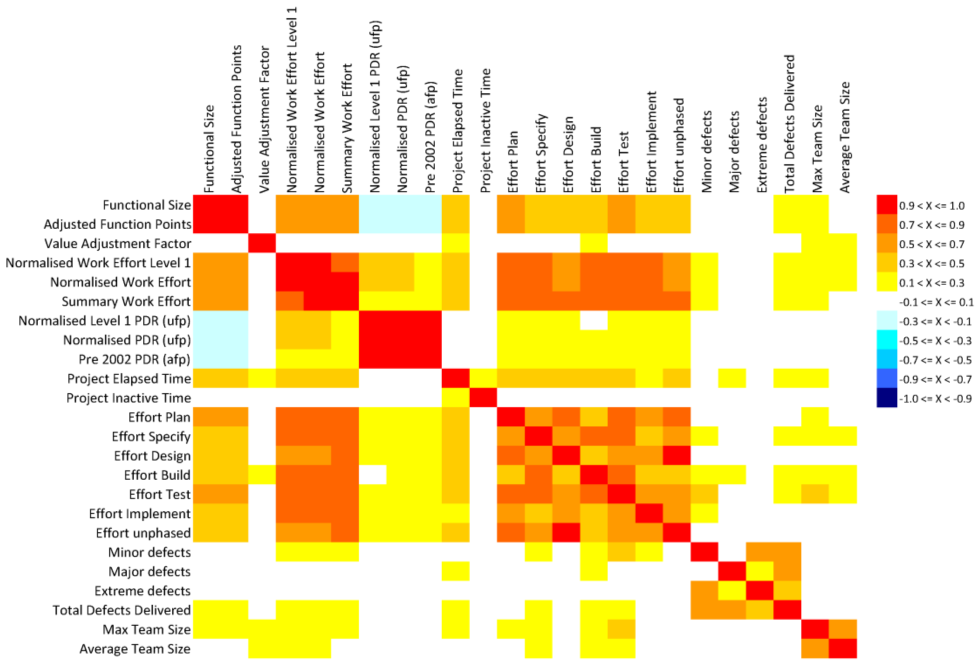

Figure 4. Linear correlations between numerical variables

In the case of defects, it is observed a direct linear relationship between the "Minor defects" and "Extreme defects"; it also appears a slight direct trend with the effort, thus it can be deduced that the high number of defects is associated with projects with strong development effort.

Another preliminary study which can be done is the multidimensional data visualization for which they have used self-organizing maps SOM (MacDonell, 2005). This visualization method allows us to identify groups of data for similar projects and to find nonlinear relationships within the variables set in exploration.

The SOM map shows the previously identified linear relationships between "Summary Work Effort" and the variables that break down the effort into phases, since high values of each of the variables are distributed in the same area of the map.

The extensive blue areas show the great amount of missing values identified above, in order to extract more information with this technique, the data should be filtered and the missing values be processed.

As a conclusion of this initial data exploration phase, it is observed that a high percentage of the dataset attributes will not be used due to the large amount of missing values; it also must be selected which categorical attributes can be transformed to obtain predictive models. This is a usual situation in this kind of environment.

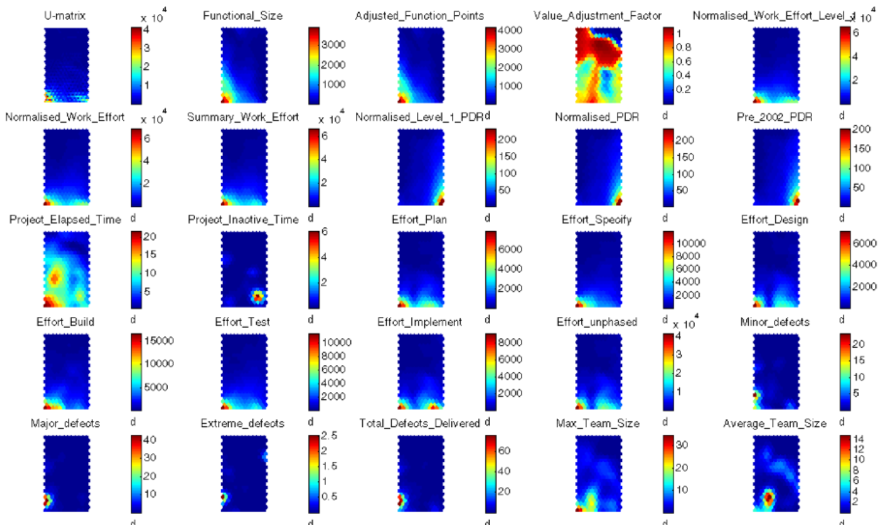

Figure 5. SOM map of the problem variables with the initial dataset

5.2. Data preparation

Next step is the preparation the data to be used for modeling technique; for this particular case it has been considered the use of predictive algorithm to generate models. To do so, the data are first prepared, filtering cases to form a homogeneous group, and clearing cases with low quality or that are outliers, but it has to preserve the cases that remain extreme but belong to a wider group.

Attributes that have a high percentage of missing values are removed from the study dataset, since they will not be able to be used.

"Normalized Work Effort Level" and "Level 1 Normalized PDR" variables have been selected as candidates for the target of effort estimation model, since they are two ways of expressing the effort, either through the reported effort by the team or the hours needed to develop a function point.

To ensure the reliability of the data that form the modeling dataset, projects that have a quality rating with little credibility are removed. Dataset will consist only of projects that have collected data about development team effort (e.g., project team, project management, project administration).

As seen in the introduction section, size estimation is a prerequisite for effort estimation, so this is an attribute that has to take part in the dataset used to generate the model. In the ISBSG repository "Count Approach" attribute is available, this is the functional size measurement method (FSM Method) used to measure the functional size (e.g. IFPUG (ISO/IEC 20926, 2003), MARK II (ISO/IEC 20968, 2002), NESMA (ISO/IEC 24750, 2005), FiSMA (ISO/IEC 29881, 2008), COSMIC-FFP (ISO/IEC 19761, 2003), etc.).

Group	Attribute	Type	Acronym	Effort M.	Defect M.
Size	Functional Size	Numeric	FS	In	In
	Adjusted Function Points	Numeric	AFP	In	In
	Value Adjustment Factor	Numeric	VAF	In	In
Effort	Normalized Work Effort Level 1	Numeric	NWE	Out	In
Productivity	Normalized Level 1 PDR	Numeric	PDR	-	In
Schedule	Project Elapsed Time	Numeric	PET	In	In
	Project Inactive Time	Numeric	PIT	In	In
	Effort Plan	Numeric	EP	-	In
	Effort Specify	Numeric	ES	-	In
	Effort Design	Numeric	ED	-	In
	Effort Build	Numeric	EB	-	In
	Effort Test	Numeric	ET	-	In
	Effort Implement	Numeric	EI	-	In
Defects	Minor defects	Numeric	MiD	-	Out
	Major defects	Numeric	MaD	-	Out
	Extreme defects	Numeric	ExD	-	Out
	Total Defects Delivered	Numeric	TD	-	Out
Project	Development Type	Categorical	DT	In	In
	Package Customization	Categorical	PC	In	In
	Language Type	Categorical	LT	In	In
	Programming Language	Categorical	PL	In	In
	CASE Tool Used	Categorical	CTU	In	In
	Used Methodology	Categorical	UM	In	In
Team	Max Team Size	Numeric	MaT	In	In
	Average Team Size	Numeric	AvT	In	In

Table 1. Data dictionary of candidate attributes for both modeling

Since the measurements performed with different methods are not comparable (Cuadrado-Gallego *et al.*, 2010), the dataset is filtered by "Count Approach" to provide uniformity with the measurements taken for functional size, in this case the projects that use IFPUG are taken, as this is the most used. This situation does not apply for an organization, as it will only use one type of measurement.

For the effort model there are certain attributes, such as the breakdown of effort phase (EP, ES, ED, EB, ET, EI and EU), that cannot be used because they are another way of expressing the effort. Other attributes are not known at the time of estimating the effort, as is the case of defects, which can only have that information after the product is developed. Other variables may be taken as model inputs to know its influence and then, if they are relevant, they can be estimated to predict the effort depending on its variation, as is the case of PET and PIT.

As the elements that cannot be involved in the effort model present a high number of missing values, it was decided to perform two separate datasets, one for each model.

Then, it is decided to take NWE "Normalized Work Effort" as target, because the productivity (PDR) is calculated as the ratio between NWE and FS. PDR was discarded since the use of normalized effort for development team only and unadjusted functional count should render the most comparable rates, but excludes the adjustment factor (VAF) that, in contrast, it does affect the effort.

For the processing of nominal variables two possible encodings have been considered, depending on the number of different values the variable may take.

In the case that a variable may take a low number of categories, as is the case of PC, CTU and UM, it has been created a number of dummy variables equal to the number of categories as shown in the table below. In this case, although the variables may take three categories, it was encoded as two new variables, and with the code "0 0" is represented no information as a state.

Package Customization (PC)	PC_Y	PC_N
No	0	1
Yes	1	0
Empty (missing value)	0	0

Table 2. Process de nominal variable PC for two binary variables

For variables that take a larger number of categories, as is the case LT, DT and PL, it has been transformed into a single numerical variable, so as not to generate so many binary variables. Since it cannot be assumed that there is the same distance among categories, therefore it cannot generate an encoding in consecutive numbers because the algorithm based on data could be misleading. For encoding these variables, each category has been replaced by the average effort associated with each category; thereby the distance between categories is preserved

Using techniques such as box-plots and percentile thresholds to remove outliers, cases that seem extreme in one variable can be lost, however they can coincide with the values taken in the other project attributes. Therefore, it was decided to make a projection of the n-dimensional space to identify those that are located far away from the rest.

There has been a data projection using the Principal Component Analysis PCA technique (Pearson, 1901), it allows us to pass from an n-dimensions space to two dimensions in this case by analyzing the principal components.

The figure shows a bubble chart with the dataset projection for which it has been used functional size (FS) as bubble size and effort (NWE) as fill color.

There is a series of clusters that appear due to the addition of nominal variables to the dataset, which generates layers in the projection. In this case, the group at the bottom of the chart is associated with "Yes" values in the package customization variable (PC).

Three projects associated with medium and high values of effort (NWE) are identified and located far from the normal behavior of the other variables in the group, especially the red point to the right that corresponds to very high values of effort (NWE) and associated with a low value of the functional size variable (FS), which would be associated with low functional complexity projects, but with an stronger effort (NWE) than the rest. It can also be remarked, that the red point is associated a value "Yes" in PC variable, which means it should be closer to the group at the bottom.

For all these reasons, it can be deduced that this project is an outlier in the dataset.

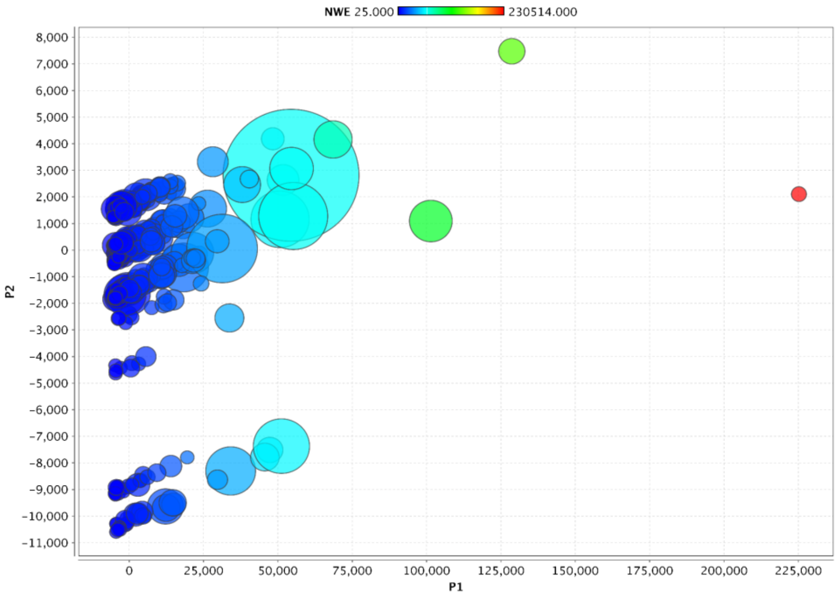

Figure 6. PCA projection dataset analyzed

The three projects that are outside normal behavior groups have been removed from the dataset, since these are projects that may be actually developed under these conditions, but they are not representative for this dataset.

PIT variable has been taken out of the dataset due to the large amount of missing values presented, which would greatly reduce the dataset. After cleaning missing values, filtering outliers and creating dummy variables, the dataset for effort model contains 409 projects and 16 attributes.

The same filters as for the above variables shall be applied on the dataset for defect models. In the case of the efforts for each phase, it shows a lot of missing values, but the total project effort has been collected.

The effort profile that a project may present over its life cycle, depends on the type of tasks it contains. This distribution of effort throughout tasks could be grouped into one of the following profiles: uniform, increasing, decreasing, bell, initial peak, final peak, and two peaks among others. Since the projects stored in the database are very different from each other, a clustering is made to identify groups using a SOM network and using the k-means clustering technique (Kanungo *et al.*, 2002).

The minimum number of clusters that produces less error in classification is three, as seen on the left side of the figure. Clusters labeled as Cluster1 and Cluster3 are a very small group of projects, for the Cluster1 the effort of the design phase is very high, while in the Cluster3 the effort of design phase is almost nonexistent. The cluster labeled as Cluster2 contains 94% of the projects studied, so that its distribution throughout the project life cycle is representative.

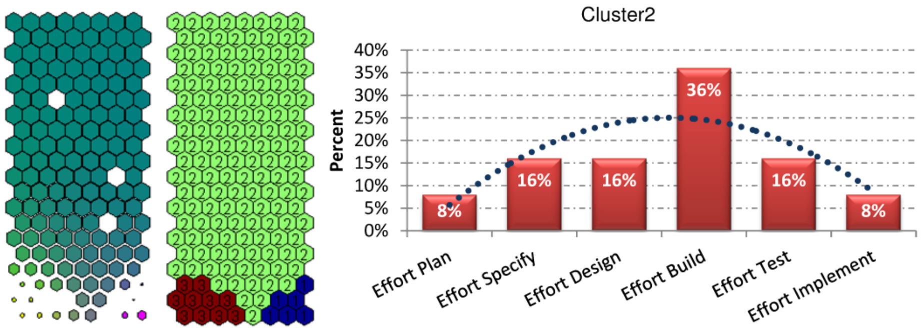

Figure 7. Identification of the load distribution of effort in the majority cluster

On the right side of the figure it can be seen the average performance of the projects that are labeled Cluster2. As it is shown, it follows a bell distribution, focusing efforts on building phase, which is standard in software projects. This distribution will be used to generate the missing values of attributes EP, ES, ED, EB, ET and IE.

Deleting cases with missing values has serious effects on the target variables defects, which are chosen to define different data sets to ensure that the target variables are representative. Thus, datasets were generated for MiD model with 413 projects, MaD model with 333 projects and ExD model with 466 projects.

5.3. Modeling effort and predicting quality through forecasted defects

Several models are built to identify factors that affect the effort and software quality, measured as defects. Afterwards, the influence of the attributes in the target variables is analyzed.

It has been decided to use Adaptive Regression Splines Multivariate MARS (Friedman, 1991) as a technique for modeling, since it is adapted to the problem using its unique characteristics and it is a promising technique in modeling complex nonlinear problems.

The characteristics that have been decisive to select MARS algorithm are, among others: robustness against collinearity of the input variables, their ability to select relevant variables and to identify the interactions between them.

Another reason for the selection of MARS algorithm was its achieving good results with dataset that do not have an abundant number of cases, provided that these are representative of the target. This dataset has been reduced, since the phases of cleaning and data filtering have reduced 10% of the initial dataset.

MARS has been used for the selection of input variables and to perform an analysis of relevant factors to be taken into consideration for estimating the effort and risk in software projects.

The process followed for training is a set of steps that begin by defining two dataset. The cases for each dataset have been selected at random. One will have 85% of the initial dataset and will be used to the model training and the remaining cases are allocated to the test dataset, whose purpose is to test the generalization of training performed. This is done to verify the model results obtained with a set of projects that have not been used for training.

In the training phase of the MARS algorithm models, it has been taken different options in algorithm configuration parameters. Different parameter settings have been tested for basic functions parameter (Friedman's nk), i.e., maximum number of model terms before pruning, as well as other parameters such as the degree of interaction (Friedman's me).

It aims to achieve a model with minimum prediction error and the least number of input attributes. In order to reduce the dimension (the number of attributes involved in the model) attributes are removed from the dataset that do not add value or have a void relative importance.

Also different combinations of variables were tested, which in theory could provide the same information to the model. An example of this combination of attributes is the case of using Functional Size (FS) next to Value Adjustment Factor (VAF) and Adjusted Function Points (AFP) in another model, since the latter is the product of two factors above, something similar is the case of PDR, NWE and FS.

In order to compare the quality of the models, it has been used Root Mean Square (RMS), which is the square root of Mean Squared Error (MSE), defined as the average difference between predicted and actual values squared.

In the case of NWE model, dimensions have been reduced to seven parameters, which are those that have obtained significant importance.

The table shows the percentage of success rate achieved by the model in each error range, both in train and in test; it also shows the results obtained by the reference model for the same interval. The column "Relative error" is the percentage of value related to the range in which effort varies, i.e. an error of less than 300 hours is around 1% of the range in which the effort is varied.

Relative error	Absolute error	NWE Train	NWE Test	Reference
1%	300	29%	26%	16%
3%	700	54%	55%	25%
7%	1500	80%	75%	44%
15%	3000	95%	89%	60%
25%	5000	97%	90%	62%
50%	10000	99%	93%	76%
75%	15000	100%	95%	80%

Table 3. Success of reference model and NWE model in training and test

To test the model results, a reference indicator on effort has been searched. Other authors have defined models for the effort but have used the variable transformations, and models were optimized for another dataset release that contains different attributes (Angelis *et al.*, 2001). Since values for the project elapsed time (measured on months) and the average team size are known in the dataset, effort is the product of time by team. It has been assumed an average of 160 hours monthly work to calculate the expected effort for the project regardless of the functional size and other factors. This way it can be checked if the model is able to extract information from the rest of attributes to provide better results in prediction. This is the reference indicator that is taken in the table.

The prediction model of effort (NWE) has an RMS of 1509 in test and the reference model has an RMS of 7212.

The figure shows the attributes that are part of the model and the relative importance of the variables for the model. The most important attribute for prediction of the effort is Average Team Size (AVT). Also, Adjusted Function Points (AFP), Project Elapsed Time (PET) and Development Type (DT) provide significant information; DT attribute describes whether the development was a new development, enhancement or re-development. Also it was involved the value "Yes" in the use of CASE tools (CTU_Y), the value "Yes" in using Methodology (UM_Y) and whether the type of project is a custom package (PC_Y).

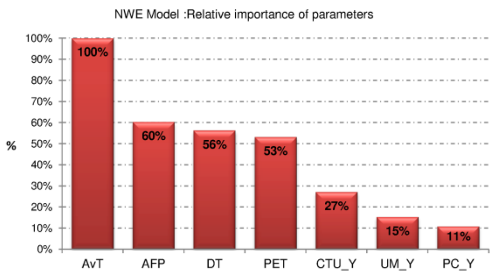

Figure 8. Relative importance of parameters in the model of NWE

For the prediction of potential software defects and therefore the prediction of the product quality that will generate, three models have been developed to perform predictions on:

Minor Defects (MiD), Major Defects (MaD) and Extreme Defects (ExD). The same methodology as for the previous model has been used.

The table shows the results of the three models for both train and test. Since there is no reference model on the prediction of possible defects in the software product, it was decided to take as reference what an organization involved in project management will do. Next, information on historical data available are collected and compiled in the dataset, and then it was taken as reference an average of defects reported in previous projects.

MiD				MaD				ExD			
Error	Train	Test	Ref.	Error	Train	Test	Ref.	Error	Train	Test	Ref.
0	46%	40%	0%	0	20%	17%	0%	0	94%	94%	0%
1	60%	57%	78%	1	44%	43%	2%	1	99%	98%	98%
2	73%	62%	82%	2	61%	59%	3%	2	99%	98%	98%
3	85%	71%	85%	3	72%	72%	8%	3	100%	100%	100%
4	91%	81%	88%	4	79%	79%	13%	4	100%	100%	100%
5	93%	83%	91%	5	82%	80%	19%	5	100%	100%	100%
10	98%	90%	95%	10	92%	87%	94%				
15	99%	95%	97%	15	97%	96%	95%				
20	99%	99%	98%	20	99%	99%	96%				

Table 4. Success of reference model and Defects models in training and test

The number of defects has been distributed with an increase in one unit until case of 5 defects and, from this point onwards, the increase has grown, because the more number of defects, the more uncertainty it presents.

The model predicting the number of minor defects has a RMS of 3.4 in test and the reference model has a RMS of 10.1. The model which presents a major number of defects has a RMS of 8.4 in test and the reference model has a RMS of 28.4. The prediction of extreme defects has a RMS of 0.3 in test and the reference model has a RMS of 0.5.

The figure shows the attributes that are part of each of the models as well as the relative importance of the variables for each model. The dimension of each problem has been diminished, in the case of MiD there are eleven attributes that define the model, eight are needed to MaD and ExD is defined by five variables.

Each model selects the attributes that best define your target attribute, but there are some attributes common to all but with different relevance.

The functional size, whether expressed as FS and VAF or AFP, appears in all models. In the case of the breakdown of the effort phases of the project life cycle, it has retained all the attributes so that the graphics are more comparable, although some do not provide information to some models.

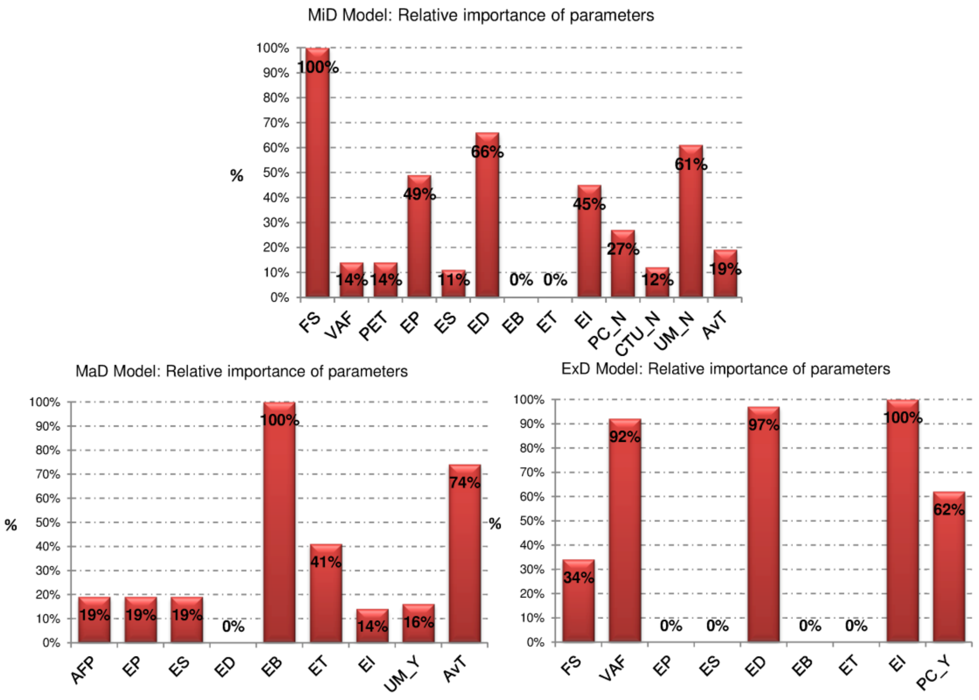

Figure 9. Relative importance of parameters in the models of defects

Other attributes as PC, CTU and UM varies depending on the model, for example, the attribute UM in the MiD model, affects the negation of the use of methodology, while MaD model affects the affirmation. Another variable that affects two models is AVT.

6. Discussion

During the data analysis, several problems associated with the great amount of nominal variables have been found (having too many levels), as well as a major presence of missing values in both nominal and numeric fields. This problem forced the rejection of many variables that could provide information to the analysis, but if missing values are filtered, it will greatly reduce the final dataset, leaving a non-representative dataset.

Next, we will proceed to analyze the results and the behavior of the most significant variables in the model to define the behavior of the effort.

The model error is less than the reference model, as the RMS of the model is much lower than the reference. Besides, the results of model tests are higher than the reference model used, therefore we can deduce that the model has been able to extract information from other attributes in addition to the team size and project time. When comparing the linear regression between model estimation and the real value of the effort we achieve a r-squared of 0.884, which is very similar to the linear adjustment obtained by other studies performing regressions (Angelis *et al.*, 2001). Considering this, it can be assumed that the model is valid to analyze and identify behaviors of the variables which give meaning to the effort. If we consider that in the dataset, the average project elapsed time is around 150 days and the average team size is about 7 resources, this implies that 300 hours of effort is equivalent to a week of work in the project. This amounts to a good estimate for an initial phase of the project, when it is important to learn about an effort in order to perform a budget as tight as possible.

We performed a sensitivity analysis to know the behavior of the model as regards the variation of one of the inputs. This is analyzed as it affects the increase or decrease of one variable with respect to the output, which is the effort.

We proceed to discuss the most significant changes that have been found in the variables that form the model of effort.

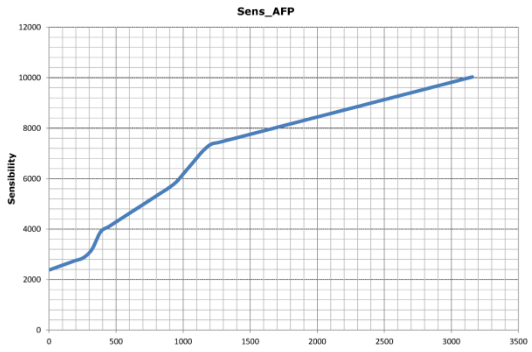

Figure 10. Sensibility of AFP in the NWE model

In some variables such as the affirmation of use of methodologies (UM_Y) it is implied that the effort is greater than in those where it is not used. However, this must not be understood as if the use of methodologies increases the effort, but that projects that often use a methodology are larger and, therefore, require more effort. Something similar occurs with the classification by the Development Type (DT); the effort increases, with the lowest type "Enhancement" followed by "New Development" and higher "Re-development". For the variable (AVT) there was an increase of the effort for average team sizes above 12.

In the case of Adjusted Function Points (AFP), the trend of the effort grows directly with AFP, but not linearly, since the slope is smooth in projects with a value of AFP higher than 1200. With elapsed time, the effort has a more stable growth for projects lasting less than one year and a half, while it increases for longer projects. This may be caused by a decrease in the number of large projects and an increase of long term uncertainty.

Predictive models of defects have a lower RMS than the reference models. Reference models are characterized by their inability to detect the absence of defects. This aspect is important since it allows us to identify the conditions that define a better quality project.

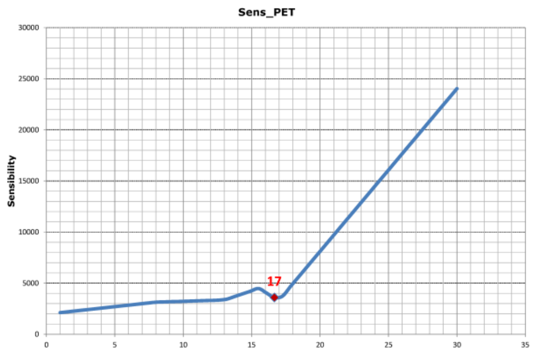

Figure 11. Sensibility of PET in the NWE model

The MiD model is able to separate in test 40% with a zero error range, and succeed 90% for a margin of 10 defects error in a range that varies 150. This model with a threshold of 2 defects achieves 90% of true positive, i.e., projects with zero minor defects are identified as such by the model.

In the case of the MaD model, it apparently reduces the scope for identification of projects without major defects and a range of 15 defects is needed in order to ensure a success of 96%, but although it seems worse than the previous model, it must be commented that the variation range is 285 defects, which implies that a margin of 15 defects is much less than that of MiD model. If we analyze only the projects that have zero major defects, this model guarantees 90% of success with a threshold of 5 defects.

After filtering, the dataset of ExD model has the variation range of 5 extreme defects. The model succeeds in nearly all cases and achieves 98% of true positives. Although this seems the best of the three when compared to the reference model, it does not improve much with the exception that it is able to guarantee 98% of true positives i.e., the model ensures zero defects in projects that really had zero extreme defects.

Analyzing the attributes that add value to the model, we observe that the functional size is always part of all models.

What makes variables PC, CTU and UM remarkable is that, for some models, it is significant the presence of dummy variable “Yes”, and for other models, it is significant the presence of dummy variable “No”. In the case of MiD model, the attributes PC_N, CTU_N and UM_N affect the increasing in the number of errors, so it follows that projects that do not use a methodology or do not use CASE tools, have a greater number of minor defects. For the model MaD, UM_Y, it has an indirect influence, i.e. their presence does reduce the number of major defects. For ExD model, if it is a customization package project, the number of extreme defects is reduced.

As efforts broken down in the project life cycle tend to have a direct tendency to the number of defects, with the exception of Design Effort (ED) and Implement Effort (EI).

The minor defects are related to efforts to design and implement (architecture and implementation phases) tending to reduce the defects, the major defects are directly related to efforts to build and test (code development phases), and extreme defect are directly related to efforts to design and implement (architecture and implementation phases).

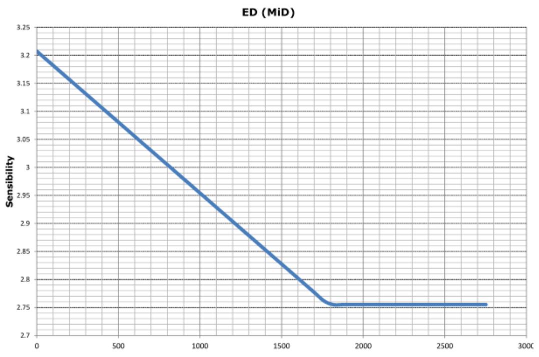

Figure 12. Sensibility of Effort Design (ED) in the MiD model

The final phase of a data mining project is the deployment of a model in an organization for which a solution is proposed. This model will be an ensemble model based on historical data from closure projects, which are provided for NWE model. Its output goes back to the repository, however at the same time it will also be an input to an NWE effort transformer into effort by phases (EP, ES, ED, EB, ET and IE; according to historical information or by type of project). Output transformation of NWE model will be the input for the three defects models, along with the information needed by each model. In this way, the four models will be linked to predict the effort and quality of the software product.

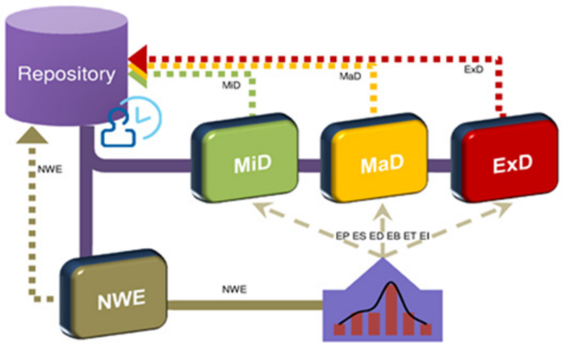

Figure 13. Ensemble model to estimate effort and defects

7. Conclusion

Software projects are involved in a changing environment, both in human and technological aspects, and do not allow generalization of models. On the contrary, data mining techniques are compatible with all project management methodologies, which aim to collect information and indicators from the project closure for its later post-mortem analysis with the aim of continuous improvement.

The general aim of this study consists in developing a method or system which facilitate the process of time and cost estimation, as well as estimate the quality of the final product by the potential risk of software defects. With the defined models, it has been achieved a process which allows us to analyze its behavior and develop perform effort estimations and estimations of risk of not meeting the quality requirements.

Given the amount of nominal variables, it is necessary to transform them so that they can provide information to the models. Most of the performed transformations have positively contributed to the improvement of several models, as in the case of the Development Type (DT) or the dummy variables of Used Methodology (UM_N, UN_Y) or CASE Tool Used (CTU_N, CTU_Y).

The key phases to model this problem have been: the selection of dataset, the stage of filtering, cleaning of outliers, as well as the use of projection techniques, whether with space dimension reduction or using self-organizing nets.

Bearing in mind that the other problem of dataset are the missing values, it can be highlighted the advantage of using clustering techniques which permit finding projects with similar characteristics, and substitute the cluster prototype where it belongs for the missing value.

The defined models allow us to make estimations since the initial phases of the project (with high level of uncertainty), with higher number of success that those of the reference models. The effort model extracts information of adjusted functional size to the technological environment where it is performed, the type of development, the use of CASE tools and the use of methodologies, among others. Defect models do not behave the same, since for instance, in the case of minor defects FS is de highest influence, for major defects the highest influence is EB and for extreme defect, it is EI and ED.

The use of databases as ISBSG, allows the contrast of process and models performed on similar datasets, although the different versions provide new fields and new projects that may make non-comparative results. These models developed from this type of database, may not be directly extrapolated to the situation of an specific organization, since the projects stored in the database have been developed under a much different work environment from the organization's. However, the applied methodology is indeed completely extrapolated, since the steps followed can be repeated on the specific data of another organization.

Finally, what has been exposed is an application process of data mining techniques to project management, and thus it may be also applied to a wider sector, it is not restricted just to the software projects.

Author details

Joaquin Villanueva Balsera, Vicente Rodriguez Montequin,
Francisco Ortega Fernandez and Carlos Alba González-Fanjul
Project Engineering Area, University of Oviedo, Spain

8. References

Acock, A. C. (2005). Working With Missing Values. *Journal of Marriage and Family*, *67*(4), 1012–1028. doi: 10.1111/j.1741-3737.2005.00191.x.

Albrecht, A. J. & Gaffney, J. E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, *SE-9*(6), 639– 648. doi: 10.1109/TSE.1983.235271.

Andreou, A. S. & Papatheocharous, E. (2008). Software cost estimation using fuzzy decision trees. *ASE 2008 - 23rd IEEE/ACM International Conference on Automated Software Engineering, Proceedings* (pp. 371–374).

Angelis, L., Stamelos, I. & Morisio, M. (2001). Building a software cost estimation model based on categorical data. *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International* (pp. 4 –15). doi: 10.1109/METRIC.2001.915511.

Boehm, B., Abts, C. & Chulani, S. (2000). *Software development cost estimation approaches – A survey*. Annals of Software Engineering.

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. & Selby, R. (1995). *Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*.

Boehm, B. W. (1981). *Software engineering economics*. Prentice-Hall.

Briand, L. C., Basili, V. R. & Thomas, W. M. (1992). A pattern recognition approach for software engineering data analysis. *Software Engineering, IEEE Transactions on*, *18*(11), 931 –942. doi: 10.1109/32.177363.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., *et al.* (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Retrieved March 27, 2012, from http://www.crisp-dm.org/CRISPWP-0800.pdf.

Chemuturi, M. (2009). *Software estimation best practices, tools & techniques: a complete guide for software project estimators*. J. Ross Publishing.

Cuadrado-Gallego, J. J., Buglione, L., Domínguez-Alda, M. J., Sevilla, M. F. d., Antonio Gutierrez de Mesa, J. & Demirors, O. (2010). An experimental study on the conversion between IFPUG and COSMIC functional size measurement units. *Information and Software Technology*, *52*(3), 347–357.

Dolado, J. . (2001). On the problem of the software cost function. *Information and Software Technology*, *43*(1), 61–72. doi: 10.1016/S0950-5849(00)00137-3.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, *19*(1), 1–67. Retrieved April 2, 2012, .

Huang, S.-J., Lin, C.-Y. & Chiu, N.-H. (n.d.). Fuzzy decision tree approach for embedding risk assessment information into software cost estimation model. *Journal of information science and engineering*, *22*(2), 297–313. Retrieved April 10, 2012, .

ISBSG. (2012). Estimation Techniques, Software Benchmarking, Software Estimation, Software Standards. *The International Software Benchmarking Standards Group Limited*. Retrieved from http://www.isbsg.org/.

ISO/IEC 19761. (2003). *Software Engineering COSMIC - Functional Size Measurement Method*. Genève: ISO, International Organization for Standardization.

ISO/IEC 20926. (2003). *Software Engineering IFPUG 4.1 Unadjusted Functional Size Measurement Method. Counting Practices Manual*. Genève: ISO, International Organization for Standardization.

ISO/IEC 20968. (2002). *Software Engineering Mk II Function Point Analysis. Counting Practices Manual*. Genève: ISO, International Organization for Standardization.

ISO/IEC 24750. (2005). *Software Engineering NESMA Functional Size Measurement Method, Version 2.1, Definitions and counting guidelines for the application of Function Point Analysis*. Genève: ISO, International Organization for Standardization.

ISO/IEC 29881. (2008). *Software Engineering, FiSMA Functional Size Measurement Method, Version 1.1*. Genève: ISO, International Organization for Standardization.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y. (2002). An efficient k-means clustering algorithm: analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *24*(7), 881 –892. doi: 10.1109/TPAMI.2002.1017616.

Lock, D. (2007). *Project Management*. Gower Publishing, Ltd.

MacDonell, S. G. (2005). Visualization and analysis of software engineering data using self-organizing maps. *2005 International Symposium on Empirical Software Engineering, 2005*. IEEE. doi: 10.1109/ISESE.2005.1541820.

Montequín, V. R., Balsera, J. V., González, C. A. & Huerta, G. M. (2005). Software project cost estimation using AI techniques. *Proceedings of the 5th WSEAS/IASME International Conference on Systems Theory and Scientific Computation*, ISTASC'05 (pp. 289–293). Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS). Retrieved April 22, 2012, from
http://dl.acm.org/citation.cfm?id=1373616.1373665.

Parkinson, C. N. (1955). Parkinson's Law. *The Economist*.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, *2*(6), 559–572.

PMI. (2009). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. PMI, Project Management Institute.

Putnam, L. H. (1978). A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *Software Engineering, IEEE Transactions on*, *SE-4*(4), 345 – 361. doi: 10.1109/TSE.1978.231521.

Putnam LH & Ann Fitzsimmons. (1979). Estimating software cost. *Datamation*.

Spector, A. & Gifford, D. (1986). A computer science perspective of bridge design. *Commun. ACM*, *29*(4), 267–283. doi: 10.1145/5684.6327.

Standish Group. (1995). The CHAOS Report (1994). *Group*, 11–13.

Standish Group. (2010). *CHAOS Report 2009*. Standish Group.

Tadayon, N. (2005). Neural network approach for software cost estimation. *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on* (Vol. 2, pp. 815 – 818 Vol. 2). doi: 10.1109/ITCC.2005.210.

Xu, Z. & Khoshgoftaar, T. M. (2004). Identification of fuzzy models of software cost estimation. *Fuzzy Sets and Systems*, *145*(1), 141–163. doi: 10.1016/j.fss.2003.10.008.