Supporting the decision making process in the software projects risk assessment

Abstract

The processes of planning and running software projects should be placed under systematic process of risk management. Its main stages are risk assessment and risk control. Risk assessment allow to choose appropriate steps, according to identified alternatives of project realization and risk connected with those alternatives. Information generated during risk assessment makes it possible to prepare project teams for changes occurring in phase of planning or project realization. Therefore, one can be ready to control the risk i.e. to develop possible plans of actions in case of risk occurrence. The article presents a conception of DSS of multidimensional software projects risk evaluation, where multidimensional approach focuses on project’s four facets: its scope, duration, costs and effects. Multiclass objects’ classification applied to software project risk allow to predict the most important parameters of analyzed projects. As a result, the knowledge from knowledge-base connected with decision-maker’s experience may improve decision making process in project management.

Keywords: Decision Support Systems, software projects, risk assessment, data mining

Introduction

Most software projects are derived from, or constitute another variety of previously executed projects. The result of such a state is a possibility to obtain valuable information related to different aspects of project execution. The aforementioned aspects should include the analysis of successes, failures, problems, and solutions of similar projects. Knowledge and experience acquired in that way are likely to facilitate the potential project risk identification and help in working out the response strategy.[12] Thus, these approaches are becoming more and more popular, which are based on an assumption that any project, even the most innovative and unique, is not a completely new system.[9] It is vital to underline that knowledge of any particular project management, should not be reserved solely for the top management. One of
basic requirements of project work is to prepare a project team properly in line with the claim that they cannot accomplish their tasks without being equipped with adequate knowledge and skills.[4] The proper conduct strategy of the project decision-makers will primarily concern the ability to introduce changes to project requirements, to make decisions on project continuation or termination, as well as on necessity to take some measures to additionally protect the project. Therefore, it is necessary to select these pieces of information which reflect the facts related to software project planning and execution.

The role of information in software project risk assessment
The level of software project risk which is acceptable for a project team is one of key factors contributing to business success and inseparable element of technical activities.[13] It is recommended that the key role in this process should be held by the project managers responsible for the risk factors identification and control [14], because skillful use of information obtained from the environment improves the effectiveness of software project management. However, the project practice indicates that the difficulties connected with including activities related to risk management in project budgets results in their cessation. That is why historic information is so important in software project planning and execution. In order to employ historic information in risk assessment-related issues it is suggested to introduce a term, namely object condition diagnostics, involving the ability to assess the real condition of the examined objects and to determine the sources of damage or disturbance occurring in the examined system (the symptoms of the system’s behaviour).[6] In the submitted applications, the symptoms of the system’s proper functioning or malfunction may be either observed directly or deduced on the basis of other variables observed while running the system.[11] In the case of software project, at the stage of planning and execution it is possible to distinguish and use both soft and hard data. Hard data constitutes a valuable source of information on risk symptoms. Soft data, on the other hand, informs of the project status. It may take a form of crisis situation signal, enabling early detection of future problems at the same time.[8] However, in a diagnostic process it is possible to identify the existence of two major types of knowledge: procedural and declarative one. Declarative knowledge describes some objects, situations and the relationships between the objects. It is also referred to as the skills and constitutes a strategy of reaching the targets. The field experts are usually the source of procedural knowledge, whereas declarative one is obtained from the databases.

Declarative knowledge exploring is usually aided by automatic knowledge discovery tools

Procedural knowledge obtained from the experts, enriches the diagnostic process with the information on problem environment i.e. object descriptions, classifications and relationships between them in particular. Despite the fundamental difference between the two types of knowledge, it is essential to combine the database knowledge with the field expert knowledge skillfully.[10]

Combining declarative and procedural knowledge is possible owing to algorithmic tools for obtaining knowledge hidden in databases and thus it completes knowledge of particular object or phenomenon. Such acquisition is referred to as knowledge discovery in databases. The process of knowledge discovery in databases consists of four major elements, i.e. data choice and pre-selection, searching for data interrelation and interpretation of results. An essential element of knowledge discovery in databases is data mining, referred to in the literature as the analysis of observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.[7]

**Evaluation of knowledge generalization ability in prediction models**

Providing fast information access is one of the most important factors likely to influence the assessment of threats associated with software project execution. However, using data mining methods allows for finding some unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the user. A choice of proper object condition classification method depends on the kind of data obtained at the stage of observation of diagnostic parameters. However it makes it possible to find projection of new data onto the previously defined data class. Such a projection helps to acquire knowledge of the system based upon the historic information. Project teams aided by automatic tools for use of historic information may obtain proper calculations of basic parameters of executed projects. In order to obtain information necessary for the decision process facilitating the software project risk assessment, several experiments were conducted aiming at selection of proper set of data mining methods for the needs of diagnostic objects classification. As a source of data on the examined objects, A Repository of Free/Libre/Open Source Software Research Data was used, as registered at SourceForge.net platform.[5] The basic premise for the decision of using a particular model to classify future data and predict the value of new examples, was the acceptance the coefficient of accuracy level of particular model. The coefficient of accuracy indicates the percent of correctly classified instances. More about the
The initial sets of information attributes pertaining to the diagnostic objects, appeared in the investigations, these were:

- **project scope** $Z(t)$ – the number of months of duration of particular project category, from the moment of decision of project initialization (project registration) till the last published presentation of the project effects, incorporated 39 attributes, inclusive of 8 attributes pertaining to the project field ($d_p$), 28 attributes pertaining to the project resources ($z_p$) and 3 attributes pertaining to project communication ($k_p$),

- **project time** $C(t)$ – task completion time expressed in terms of working hours spent on completing a particular task incorporated 11 attributes, inclusive of 6 attributes pertaining to general conditions of task completion ($w_t$), and 5 attributes pertaining to the resources of persons completing the task ($z_t$),

- **project cost** $K(t)$ – the average number of working hours spent by a particular project contractor on task completion incorporated 18 attributes, inclusive of 8 attributes pertaining to the participant competence ($z_u$) and 10 attributes pertaining to the participant activity ($a_u$)

- **project effects** $E(t)$ – the number of completed tasks as of the date of diagnosis incorporated 21 attributes, inclusive of 16 attributes pertaining to activity of project execution ($r_a$) and 5 attributes pertaining to communication activity related to project execution ($k_a$).

The abbreviation “A” in Table 1 indicates the number of attributes in a particular model.

Table 1. Comparison of prediction models for the software project risk assessment [3]

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Project scope (Accuracy)</th>
<th>Project time (Accuracy)</th>
<th>Project cos (Accuracy)</th>
<th>Project effects (Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>BBN</td>
<td>93.6129%</td>
<td>60.2400%</td>
<td>69.8030%</td>
<td>46.1851%</td>
</tr>
<tr>
<td>C4.5</td>
<td>97.3560%</td>
<td>67.6481%</td>
<td>78.2700%</td>
<td>55.3124%</td>
</tr>
<tr>
<td>MLP</td>
<td>44.4376%</td>
<td>34.5709%</td>
<td>38.6135%</td>
<td>38.5102%</td>
</tr>
<tr>
<td>RT</td>
<td>99.2803%</td>
<td>71.8745%</td>
<td>91.9815%</td>
<td>76.8332%</td>
</tr>
</tbody>
</table>

Comparison of the accuracy ratio of prediction models with the required no. of information attributes

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Project scope (Accuracy)</th>
<th>Project time (Accuracy)</th>
<th>Project cos (Accuracy)</th>
<th>Project effects (Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>BBN</td>
<td>91.3183%</td>
<td>58.4271%</td>
<td>65.5481%</td>
<td>44.1325%</td>
</tr>
<tr>
<td>C4.5</td>
<td>94.9654%</td>
<td>64.3034%</td>
<td>77.5000%</td>
<td>51.3104%</td>
</tr>
<tr>
<td>MLP</td>
<td>31.6062%</td>
<td>32.7989%</td>
<td>37.9305%</td>
<td>38.5102%</td>
</tr>
<tr>
<td>RT</td>
<td>97.1019%</td>
<td>71.8745%</td>
<td>91.1807%</td>
<td>67.1637%</td>
</tr>
</tbody>
</table>

Based on the test results (Table 1) it may be observed that the best results were achieved for the Random Trees (RT), good results were also proved for such classifiers as: Decision Tree (C4.5) and Bayesian Belief Networks (BBN). That is why the use of the discussed methods is recommended for the issues related to software project risk assessment. However, the observed low level of classification represented by Multilayer Perceptron (MLP) may be caused by the use in the test of the majority of categorical variables not being the ordinal variables.

Knowledge of the abovementioned indexes enabled obtaining of the selected set of attributes pertaining to the examined objects. Selecting the reduced set of descriptors helped to limit the decision space. The main advantages of reducing the number of attributes are, above all, time and memory saving in classification of new examples. The selected sets of information attributes for the best prediction models (desired attributes) were:

- \[ Z(t) = \{ z_1, d_8, d_1, z_4, d_7, z_8, z_2, d_5, d_3 \} \]
- \[ C(t) = \{ w_1, w_7, w_5, w_2, z_5, w_4, z_4, z_3, z_2 \} \]
- \[ K(t) = \{ z_1, a_1, z_6, z_8, a_2, a_4 \} \]
- \[ E(t) = \{ r_2, r_{13}, r_{11}, r_{14}, r_{10}, r_9, r_8, r_{12}, k_5, k_4, r_7, k_1 \} \]

The selected sets of information attributes required for the prediction of diagnostic objects responses were:

- \[ Z(t) = \{ z_1, d_8, d_1, z_4 \} \]
- \[ C(t) = \{ w_1, w_7, w_5, w_2 \} \]
- \[ K(t) = \{ z_1, a_1 \} \]
- \[ E(t) = \{ r_2, r_{13}, r_{11} \} \]

Explanation of individual information attributes for a diagnostic parameter project scope \( Z(t) \):

- the average number of working hours spent on completion of the project tasks (\( z_1 \)),
- number of selected project tasks (\( d_8 \)),
- subject scope of the software (\( d_1 \)),
- number of project contractors (\( z_4 \)),
- number of selected subprojects (\( d_7 \)),
- number of project contractors at the position of a Developer (\( z_8 \)),
- number of unique time zones (\( z_2 \)),
- software language (\( d_5 \)),
- user interface (\( d_3 \)).
Explanation of individual information attributes for a diagnostic parameter project time $C(t)$:

- task completion time expressed as a number of days from the task initialization till its completion ($w_1$),
- number of tasks within a subproject ($w_7$),
- number of preceding tasks which a particular task depends on ($w_5$),
- percentage of task completion ($w_2$),
- skills of a project contractor completing the task ($z_5$),
- task status ($w_4$),
- role / position of a project contractor completing the task ($z_4$),
- role / position of a project contractor assigning the task ($z_3$),
- number of contractors assigned to complete the task ($z_2$).

Explanation of individual information attributes for a diagnostic parameter project cost $K(t)$:

- registered time of the project contractor (mths) ($z_1$),
- number of assigned tasks ($a_1$),
- number of unique skills of project contractor ($z_6$),
- the most frequently held role/ assigned position ($z_8$),
- number of projects executed by a particular contractor ($a_2$),
- number of posts sent by a particular contractors to different project – related Web forums ($a_4$).

Explanation of individual information attributes for a control parameter project effects $E(t)$:

- the average task completion status in percentage ($r_2$),
- number of tests in CVS version control system ($r_{13}$),
- number of open artifacts (additional artifacts in the project) ($r_{11}$), number of comments in CVS version control system ($r_{14}$),
- number of closed error reports ($r_{10}$),
- number of open error reports ($r_9$),
- number of requests for help ($r_8$),
- number of closed artifacts (additional artifacts in the project) ($r_{12}$),
- number of project mailing lists ($k_5$),
- number of threads on forums assigned to particular project ($k_4$),
- number of closed binary code modifications ($r_7$),

– number of project documentation groups \((k_1)\).

The abovementioned attributes constitute a set of required / desired data for the diagnostic process aiming at software project risk assessment. Therefore, they also constitute components of base for prediction models which allows for multidimensional assessment of software project risk.

The concept of decision support system for multidimensional assessment of software project risk

The developed base of prediction models forms a background for a discussion on possibilities of multidimensional assessment of software project risk within a decision support system. Moreover, it enables to undertake the control measures aiming at identification of the actual status of project execution. Knowledge of values of basic parameters as well as knowledge of the type of project helps to determine the risk profile. The software project risk profile includes the conditions of the project execution. Both at the stage of planning as well as execution it is advised to identify following types of projects: [2]

– safe project – the project in which no substantial deviation is noted of real values of the decision attribute derived from the diagnostic object, from the prediction values as indicated by the prediction model, based upon the information obtained from the project

– cautious project – the project in which some substantial deviations were noted of real values of the decision attribute derived from the diagnostic object, from the prediction values as indicated by the prediction model, based upon the information obtained from the project, but – at the same time – some repair actions were undertaken aiming at identification of the location, degree and reason of discrepancy as well as project assumption modification

– aggressive project – the project in which some substantial deviations were noted of real values of the decision attribute derived from the diagnostic object, from the prediction values as indicated by the prediction model, based upon the information obtained from the project, and – at the same time – no repair action was undertaken aiming either at identification of the location, degree and reason of discrepancy or project assumption modification.
The profiles of a safe and an aggressive project are identified and referred to as the final status, whereas the cautious project profile is an intermediary status and equals the necessity to make decisions concerning future actions. Proper closing of diagnostic process occurs at the moment of execution of all the project assumptions and the project success. In the system proposal, as shown in Fig. 1, it is also possible to terminate the diagnostic system at the stage of project continuation, at the moment of the risk factor identification – the aggressive project. The detailed decision process is shown in Fig. 1.
Fig. 1 The concept of decision support system for multidimensional assessment of software project risk [3]

Risk profile identification – aggressive project occurs in the event of making the decision by the decision-maker on cessation of repair actions and project continuation based upon initial assumptions. Closing the diagnostic process may also occur in the event of termination of project works and project cessation. At the moment of closing the diagnostic process, each case is included in the database as a new record to be used for deducing in the next diagnostic process. The advantage of the suggested solution is connected with the small amount of information to be supplied to the decision support system in order to classify a particular project as belonging to one of the identified classes of decision attributes. Moreover, this approach enables early detection of abnormalities in project assumptions, thus indicating the modification area. It is a purposeful action not to perceive risk assessment in terms of such expressions as low/intermediate/high risk anymore, since these are strongly individualized terms. The decision-maker receives feedback information in a form of a message of conformity/nonconformity of the diagnostic parameter values with their assumed/real values for a particular project. The advantage of such approach to the risk assessment is the possibility to work out the software project schedule and budget consciously.

**Summary**
Possessing the information obtained during risk assessment makes it possible for the project teams to prepare well for the changes during project planning and execution, whereas the risk assessment itself helps them to choose proper conduct strategy, by means of identification of different options of project execution and the related risks. Therefore, the project team is equipped with the possibility to prepare well for the second phase of risk management, i.e. risk control including working out proper preventive measures against threats and the response strategy in the event of their occurrence. The suggested approach may be particularly useful in software project management, especially in supporting the project team works. Due to the fact that multidimensional assessment of software project risk does not require large investments, and its functionality helps to predict the project response, this tool may be employed by the institutions dealing with software project financing as well as by end-users of software systems.

**Acknowledgements**
This work was supported by grant no. N115 080 32/4229

Bibliography:

5. Free/Open Source Software (F/OSS) Development Phenomenon: Understanding Open Source Software Development. National Science Foundation, CISE/IIS-Digital Society & Technology, Grant No. 0222829
10. Moczulski W.: Knowledge Engineering and Diagnostics - Today and Tomorrow, Diagnostyka 30 tom 2, 2004