METHOD OF CALCULATING PROJECT EVALUATION INDICATORS FOR THE IMPLEMENTATION OF SOFTWARE SYSTEM REENGINEERING

Tymofieieva O., postgraduate student, Velykodniy S., PhD in Computer Science, Associate Professor, Zaitseva-Velykodna S., technician 1-st cat.
Odessa State Environmental University, Ukraine

The main objective for any project automation design system is cost reduction of designing a system which can be used in any industry. The presented study provides an assessment of design, or to be more specific, software system redesign.

When it comes to managing net cost for the project design it is generally achieved through reducing time and / or people necessary for software development. Software systems are used at work and in personal lives by many people and further civilization development makes it almost impossible to find an area of human life not affected by computers.

Operation, or more commonly use of software systems in each particular industry (information technology, production management, transport, telecommunications, scientific activities, education, etc.) has its own fundamental differences.

At the same time, all software systems has something in common – they age under the influence of time and other unavoidable factors of informatization, e.g. constant need to update operating systems, languages, principles of operation for distributed data processing systems, which is especially important in a field of information technology. This trend leads to performance deterioration for computer systems, such as slower processing speed, low quality graphic rendering, loss of information and communication, until its complete failure. That is why, the question arises of the need for enhancement, improvement and upgrade of software systems and all of abovementioned processes are components of the complex components of “re-engineering”.

Thus, the subject that involves compiling a method for calculating project appraisal indices while performing re-engineering of software system is relevant.

Therefore, re-engineering is a radical transformation (redesign) intended to improve the technical characteristics and remove negative qualities of the primary object.

In order to perform project re-engineering, it is essential to set up parameters essential to calculate the foreseen financial expenditure. As well as to evaluate and determine the complexity of re-engineering for existing system that needs to be changed over time. Assessment of the project complexity is necessary for further evaluation and determination of time required for re-engineering which includes following phases: redesign, testing, maintenance.

Many experts believe that accounting for expenditures based on Karner's method allows to get an estimate with an error of 20% relative to real cost. Therefore, this method will be used as a base for implementation of software system re-engineering, with its further development in relation to the re-engineering procedures.

The answer to this question is given by the Karner's method (Karner's Use Case Points Method). According to Karner's, premises points' variety are functions of the following arguments:

a) the point size of the software that accounts for the number and complexity of use cases;
b) the point size of the software that accounts for the number and complexity of actors;
c) variety of non-functional requirements such as labor efficiency, portability, which were not included into business use cases;
d) development environment (languages, actor's motivation).

The Karner's methodology offers a general assessment of the labor cost of the project but it does not allow to select or emphasize any of its phases. Moreover, this technique can only be used after all use case diagrams for the system will be developed.

The factor of technical complexity of the project is calculated on the basis of how essential is each technical factor to the system.

It is imperative to set up technical complexity and environments factors for the system before estimating the size of the project. In order to determine technical complexity factor (TCF) and environmental complexity factor (ECF) it is necessary to fill a list of factors that will affect the project indicators.

The first of these factors is the weight factor, which is determined by the Karner's method for project points although they could be adjusted according to the specific requirements of the project.

The next factor is the value that indicates the degree of importance of the determined factor on the project. Its value is assigned a score between “0” and “5”, where “0” – factor is irrelevant and “5” – factor is essential.

Before you evaluate the software system with the help of indicators using the functions that will be subject to re-engineering, you need to assign weight for each of the options to be used in future software system, based of project factors.

The appropriate weight should be assigned to the Use Case if there is at least one of the factors with a maximum score for the respective rank.

Most indicators require comments on how they were gathered. The description of the factors and coefficients is provided below. The point size of the software that accounts for the number and complexity of use cases is reflected in Unadjusted Use Case Points (UUCP). The Technical Complexity Factor (TCF) is calculated on the bases of several indicators.

The UTV (Unadjusted TCF Value) calculated based on a list of factors of technical complexity for the project and to which, in addition to weight, a score assigned, influenced by the degree of importance of the determined factor where “0” – factor is irrelevant and “5” – is factor is essential.

Weight coefficient TWF (TCF Weight Factor) is assigned by default as 0.01; although it can be corrected in theory. Project Constant TC (TCF Constant) 0,6 – is guaranteed minimum for TCF or from its theoretical minimum = 0,6 till its theoretical maximum = 1,25 and that shows both: positive impact on cost – decrease by 40% and negative – increase by 25%.

Since CAD/CAM/CAE is a consolidation of the tools which also includes staff, it is necessary to consider project participants in the calculation of the project indicators, that is, to include factor of developers (actors) to assess the complexity of software reengineering.

The Environmental Complexity Factor (ECF) is another factor applied to estimated size of the software in order to account for environmental consideration of the system. It is calculated based on several factors.

Unadjusted ECF Value (UEV) can be determined based on list of qualification of the developers working on reengineering project. It is determined by quantifying set of skills (assigning a score) of people working on a project. It ranges between “0” – no experience and “5” – expert.

Weight coefficient EWF (ECF Weight Factor) is assigned by default as (-0,03) although it could be corrected. Project Constant EC (ECF Constant) = 1,4 is guaranteed maximum for ECF. It could fall anywhere in between its minimum 0,4 and maximum 1,4 and that shows both: positive impact on expenses - reduction by 60% - and/or negative – increase by 40%.

After calculation of these factors you can proceed to the assessment of the project. The computed value for the UCP (Use Case Points) is the number obtained for a simple Use Case. The value is measured as so-called "project points".

DR (Duration) is the amount of time required for developing one USP's "design point", which is best chosen based on experience in similar project. DR is a crucial factor and although its default value equals 10 hours it can easily exceed 30 hours if the UEV value is unbalanced. When choosing DR it is necessary to take into account the qualifications of developers – for example for a new team you should assign DR = 20 hours.

While working with the subject matter of the article, going through numerous reviews and analysis of implementation of Karner's method, authors came to the following conclusions:

1) According to Karner's method it's better to exclude extended Use Case when calculating design points. The authors disagree with this and recommends to include all Use Case into consideration while evaluating the project, because if the abovementioned Use Case required functional changes, then there is always an extra effort involved to redesign them which should be taken into account.

2) The best way to fine-tune a redesigned project in your CAD/CAM/CAE is to consider Use Case that were successfully completed in similar projects. After analysing completed project and examining the report, available factors can be accurately adjusted to provide quite accurate estimate for actual hours involved in reengineering. Subsequently this data can be used as a base line for project life cycle.

3) In order to analyse whether it is possible to redesign a simple, medium or complex for use cases in a timely manner (including all stages of creation) an adequate performance test, based on data for average hours per use case, is necessary.

4) No one should expect an definite answer to the question "How much of re-engineering is worth?" or "How long it will last?", but it is necessary to carefully examine all obtained data, analyze successful projects implementation.

The authors hope that using this technique will result in a lesser degree of error (<20%) relative to actual expenditures compare to the Karner's method.

Thus, when compiling the methodology for estimating indicators for software system that will be subject to re-engineering, the following factors that influence the project resources were identified:

a) studying requirements model for software system;
b) number of steps required for reengineering of piece of software system;
c) technical difficulty of the project;
d) qualification level of developers' team;

Each variable used for re-engineering calculation is determined and calculated separately using:

a) measurements of distinctive parameters;
b) weight value;
c) constant constrains.

Project manager provides estimations for all parameters based on his/her experience dealing with projects similar by technical complexity and the capability of the team that will be re-engineering the software.

The establishment of coefficients and selection constants' values are based on numerous data and performed using A. Jacobson technology for the similar projects.

Therefore, as a scientific novelty, we can draw conclusions on the further development of the Karner's method of calculating design points with introduction of significant additions and extensions. As a result, forming a report with detailed analyses for project evaluation based on calculated indicators of evaluation. The reports become part of final design documentation imperative to predict successful implementation of software system redesign. Project documentation is compulsory part of the administrative support for automation software system redesign.