Characteristics for Software Optimization Projects

Cătălin BOJA, Marius POPA, Iulian NIȚESCU
Academy of Economic Studies, Bucharest

The increasing of the software systems complexity imposes the identification and implementation of some methods and techniques in order to manage it. The software optimization project is a way in which the software complexity is controlled. The software optimization project must face to the organization need to earn profit. The software optimization project is an integrated part of the application cycle because share same resources, depends on other stages and influences next phases. The optimization project has some particularities because it works on an finished product around its quality. The process is quality and performance oriented and it assumes that the product life cycle is almost finished.
Keywords*: optimization, software, project management, quality, performance.*

1 Optimization of software applications

The evolution of software and hardware technologies permits for complex software applications in the present, but also with great requirements for processing speed and memory usage. Software applications included in this category are operating systems, entertainment applications and multimedia applications.

With all this waste of resources, transparent to the user, the developer gives particular importance to the optimization process, looking to maximize the performance of the final product.

Another category of software products is constrained from the start to be efficient with respect to the system resources used. In this category are included antivirus applications, drivers, viruses, applications implemented in microcontrollers or smartcards, function libraries, applications for mobile devices.

The objective of software optimization is to obtain a new product or a new version of an existing product, which presents a higher quality level. This grade is worked out based on the levels obtained from the set of software characteristics or the established optimum criteria. These are well defined and are followed throughout the whole process, directly measuring the obtained levels. By direct comparison to the base levels or by determining aggregated values based on the way multi-criteria models are composed, the level of improvement is obtained. One such model, [IVAN06], permits determining an aggregated quality level which measures the effects of multi-criteria optimization.

The optimization process implements techniques and methods used in:

- problem analysis – implies that a lot of the problems in software optimization are generated in stages before the development of the source code; the implementation of an inefficient analysis leads to defining a solution that isn't characterized by a required quality level; the definition and analysis of a data model that doesn't take into account all the aspects of the problem to be solved leads to memory wastage or a raised processing effort in getting the data to a desired format;
- source code – if this is based on a bad implementation of an algorithm, it will lead to obtaining inefficient results in most situations, even if the complexity of the source code may be reduced, or if it is of high quality; the main cause of problems at source code level lies in a low level of its experience, and last but not least in the mistakes it makes; the primary methods, [IVAN06], used at this level are based on the elimination of repeating sub-expressions, instructions without any meaning, sequences in which instructions with opposite effect appear, invariations, by substituting complex reference expressions with simpler ones and by regrouping control structures;
- compiler – this component is responsible for transforming the source code, that has the form associated with the high level language, into machine code; as this form is directly

processed by the microprocessor, it greatly influences the way in which resources are used and especially the total processing time; a good complier contains a lot of techniques for optimizing memory usage and processing speed; it leads to the optimization of the application without the need for any other additional effort; the second solution for obtaining optimized machine code is to write the source code in assembler languages; analyzing the efficiency of using an optimization routine for the compiler or writing the entire application in assembler languages there are defined two approaches; in the case of routines, with a high level of importance for application performance, the programmer can generate machine code more efficiently than the compiler; the solution of developing the whole application in machine code is not viable because the effort and time resources are too great.

The stages carried out for software optimization are executed after finishing a stage from the development process and the result is redoing that stage with the necessary modifications or confirming the objective is reached and validating the transition to the next stage.

The overall image of the development process for a software application makes the impact of available costs and resources upon all activities stand out. For this reason, the stages of the optimization process have to be tied to the same restrictions with regards to the available resources. Through the prism of limited development time, delivery stages, limited number of personnel and last but not least of available financial resources, the optimization of the software application has also got to be carried out in optimized conditions. The following situations have to be avoided:

- software optimization is carried out infinitely, continuously defining new objectives;
- the objectives imperil the development cycle of the software application;
- the resources implicated are greater than allocated quantities.

It is very important that the decision for software optimization is taken based on an objective and detailed analysis of the functional software product. Also, the cost-benefit analysis has to highlight the raised importance of results in proportion to the financial effort and implicated time for beginning the software optimization stage.

At the software application level, objectives are based upon the degree to which each component positively or negatively affects performance and qualitative level of the entire application. Through testing, the modules that have a high impact on quality characteristics are identified, and the optimization process will be geared towards those modules. This approach permits reduction of costs through concatenation of effort on a small part of the application and will contribute to obtaining a better software product.

The optimization process applies to a functioning product because it needs to permit the direct measurement of an initial quality level, which will become a comparison base for evaluating the effects of optimization. Tackling optimization through simulation is based on defining complex methods that take into account the multitude of factors that influence directly or indirectly the quality level of the application. Simulation methods are used for filtering or to show possible solutions for future improvement. The directions for continuing the process are implemented in the development stage, leading to one or more functional versions. The results for evaluating result quality are totally based upon levels directly measured in real testing conditions. In case the final objectives have not been met, or the development process continues with new components, the result from the current optimization stage becomes and entry point for the next stage.

2. The software optimization project

Within an organization that develops activities in any field, the Information Technology (IT) is implemented for strategic reasons. Information Technology has to improve the efficiency and the control and productivity for the processes of the organization.

In [STEW07], there is depicted the process for IT project life cycle management process.

The described framework has the following modules:
- IT project selection – there have been numerous IT projects that failed; the reason is a limited understanding between the IT investments into an organization and organizational performance; a good structured IT project selection ensure that it will be selected those IT projects that support the organizational needs and identified the risks before the investment and implementation;
- Strategic IT implementation and monitoring – the implementation supposes a very good planning in order to reduce the gap between the outputs and expectation from IT investments; in order to improve the rate output-expectations, it is needed a monitoring framework;
- IT performance evaluation – the IT investments produce benefits that are hard to be identify and measure; in [STEW07], there are presented some perspectives regarding an empirical framework for benefits of the IT investments.
In figure 1, the process of the IT project life cycle management is depicted, as it is shown in [STEW07].
Based on the life cycle diagram described in figure 1, the place of the optimization project main core is in the EvaluateIT management phase. The figure 2 describes the stages of the process being focused on:

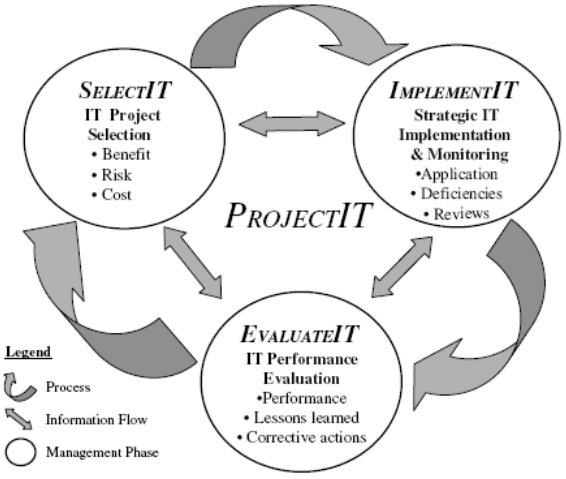

Fig.1. The process of the IT project life cycle management [STEW07]

- performance evaluation; the application is analyzed and measured form the viewpoint of selected software quality characteristics;
- problem analysis; the results of the previous stage are analyzed in order to give the most efficient solutions;
- testing; each proposed solution is implemented and is tested measuring the new quality levels;
- solution selection; based on an empirical assessment approach of the optimization process there are identified the solutions that best resolves the problem.

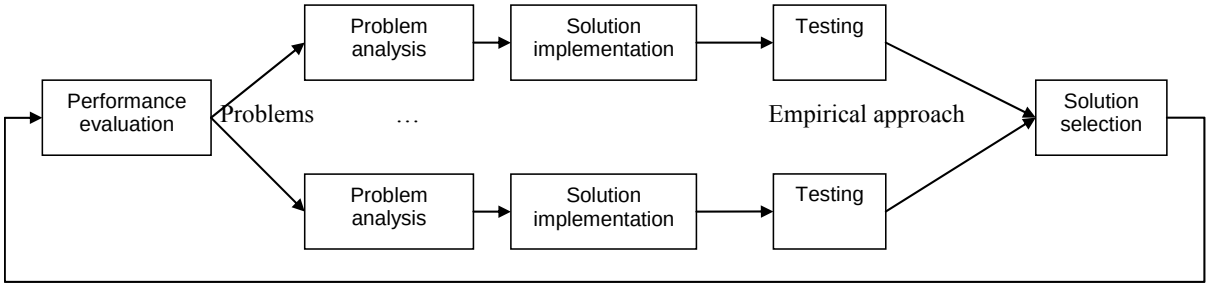

Fig.2. The process of the IT optimization project

As far as project management is concerned, the activities associated with the optimization process have as a final objective a functional software product with a higher quality level that satisfies the specifications defined by the user or producer. The entrance in this process is determined by an initial application that solves the problem for which it was written, but not within the specified parameters.
The project interacts with:
- the developers of the software application; these define the project objectives, the limits

in which the cost has to be in, and also the quality level of the final product;
- users of the application; based on the specifications defined in the analysis phase, the objectives of the process are constructed, but as parts of the application are written, their validation is necessary, or detailing them through identification of new requirements; the users' interest towards obtaining a functional application assists the developer in identifying characteristics for application functionality but also what efficiency means from a different point of view than that of the programmer.
- software products from the same category; the software product market is a free market in which trends are dictated by supply and demand, and companies that produce software for internal use are rare; existing software applications strongly influence optimum criteria for a new products as they establish the standard for the specifications' level.

Even though the development process of a software product is based on very strict rules, which are defined by the used programming language, analysis and design methods and techniques, hardware and software platform specifications, it complies with the rules for risk situation appearances.

If for risk situations generated by the technical side there are numerous backup solutions, for avoiding human error, rules need to be implemented that imply:
- code documentation; in the case of complex applications, the leave of a person from a project is not such a big problem as is training a new person to take their place; this implies delays in attaining the desired objectives, raised costs and in the worst case not following the requirements;
- clear procedures for evaluating implementation solutions; these have to be gradually applied so that the project is not permitted to stray in a wrong direction; for example, a mistake in the analysis phase will lead to great consequences in the development phase;
- procedures and methods for measuring the quality level of the application; this way results are evaluated and the ratio cost/efficiency is determined to see if the process should continue or not;
- procedures for evaluating the productivity and efficiency of programmers; the solution implemented by the optimization process is analyzed through this approach from the point of view of programmers effort that is needed in order to achieve objectives.

As in any business, the software development organizations want to increase the profit level and to reduce the risks of their activities [COST06]. In software projects, resources are represented by the software development teams, the hardware and software items, and the returns is represented by the profit expected to be earned.

This fact limits the duration of the optimization project because its costs affects the entire plan. This process has high levels of needed time and other resources because its target aims for a high level of software quality. This requires trained and experienced team members whose role is to improve something that is already developed. In addition, the testing phases need lot of time to perform and to analyze the results.

The risk management has increased in the last few years. In IT software projects, there are uncertainties faced by the IT software projects. These uncertainties must be taken into account when the software project is planned and controlled. The goal is to reduce the rate output-expectations in order to maximize the profit of the software development organization. Main risks regarding the optimization project are defined by the way goals are established. The most unfavorable situation is the one in which the project fails to reach its objectives or the costs to do that are too big. An experienced project manager must analyze the situation and see if the product may be optimized and what are the costs to do that.

In [COST06], a questionnaire for risk factors evaluation is presented, table 1.

Also, in [COST06] the risk questionnaire is presented.

For each question, the project manager must assign a number from 0 to 5. After the com-

pleting of the questionnaire, for each factor it is calculated the median value that it is mapped to [0; 1] interval. The resulted values are used by managers in their decision-making activities.

Table no. 1 Risk factors and their questions [COST06]

Factor	Number of questions
Analysis	28
Design	17
Coding	11
Testing	25
Planning	36
Control	17
Team	32
Policies and structure	8
Contracts	21
Clients	16

3. The characteristics for the process

Defining the characteristics for the optimization process of software applications is significant for successfully carrying out the management for such a process.

The level of resources needed in writing a software application needs a good administration of activities and a permanent evaluation of intermediate and final results.

The characteristics of the process constitute the basis for defining quality models that are used when evaluating or auditing the project for improving the software product:

- clearly stating the objective – this has to contain the purpose of the optimized application as well as the levels that need to be reached for the optimum criteria;
- stating the quality characteristics and the followed optimum criteria;
- defining the method for measuring each characteristic – metrics and models are implemented for aggregating the values of the multiple criteria considered;
- defining the correlation model for the characteristics – the separate variances recorded from the considered quality characteristics, the non-complementary character of the effects and the high degree of influence which they present one upon another impose the definition of a model on the basis of which one should be able to exactly establish whether or not improvements have been obtained or not;
- defining an indicator for the general quality of the application based upon the above model;
- the analysis of the software application based on the quality characteristics – the modules that have the highest degree of influence upon these characteristics have to be identified; this way the method and the effort needed for optimizing that particular component can be established; identifying the critical modules is done based on the producer's experience or empirically by running tests that measure the levels of the quality characteristics for the respective modules; the instruments used, profilers, are included in many development environments and offer a quantitative image of the quality level for the entire application or for its components;
- defining the optimization project;
- splitting the project into smaller projects dealing with the optimization of each characteristic – the quality level of the application is estimated from the considered model;
- optimization projects are based on a cyclic cycle of steps until the required level is attained or until the best level is attained based on available resources – also, the decision to continue or stop the process is based upon an empirical analysis of the result;
- using the cost-benefit analysis for each solution for verifying if the solution is feasible or efficient.

The complexity of the actual software projects imposes computer aided tools to plan the project development. To earned profit, the software development organizations must reduce the duration and the cost of the software projects.

4. Conclusions

The paper presented the main characteristics for the management of the software optimization project.

The complexity of the IT projects increased in the last few years. In order to reduce the

rate output-expectations, the project managers must take into account new methods and techniques to reduce or/and manage the complexity. They must re-engineer the traditional business processes.

In a very competitive software market that is characterized by:

- free flow of ideas, because people are migrating for one company to another and mainly because of the Internet, the biggest world wide billboard;
- wide access to new technologies; very few companies developed proprietary frameworks from zero; it is more efficient to developed around existing solutions like Java and .NET platform;
- great number of users;
- wide area of software types and categories that cover most of the social and economical domains;
- different producers for the same type of software applications;

the success if fully dependent on the quality and performance of the product.

From this point of view, the optimization project receives a new level of understanding. It becomes a secondary cycle in the product development cycle.

References

[BOJA05] Cătălin BOJA – *Software Multicriterial Optimization,* The Proceedings of the Seventh International Conference of Informatics in Economy, May 2005, Academy of Economic Studies, Bucharest, Romania, INFOREC Printing House, pp. 1068 – 1074, ISBN 973-8360-04-8

[BOJA05a] Cătălin BOJA, Ion IVAN – *Collaborative systems components empirical software optimization effect assessment*, The proceedings of The International Workshop on COLLABORATIVE SUPPORT SYSTEMS IN BUSINESS AND EDUCATION, October 28 – 29, 2005, Cluj-Napoca, Romania, Risoprint Printing House, Cluj-Napoca, pp 152 – 189, ISBN 973-651-008-9

[COST06] Helio COSTA, Marcio BARROS, Guilherme TRAVASSOS – *Evaluating Software Project Portofolio Risks*, The Journal of Systems and Software, no. 80, 2007, pp. 16 – 31

[IVAN05a] Ion IVAN, Cătălin BOJA – *Empirical Software Optimization*, Economic Informatics Journal, vol. IX, no. 2/2005, INFOREC Printing House, Bucharest, 2005, ISSN 1453 – 1305, pp 43 – 50

[IVAN05b] Ion IVAN, Cătălin BOJA – *Global Software Optimization*, pp. 205 – 214, Information Systems & Operations Management – ISOM no. 3 Workshop, April 20 – 21, 2005, Romanian American University Master of Economic Informatics, Academy of Economic Studies Master BRIE, Bucharest, ProUniversalis Printing House, ISBN 973-87166-8-3

[IVAN05c] Ion IVAN, Cătălin BOJA – *Managementul calităţii proiectelor TIC*, ASE Printing House, Bucharest, 2005, ISBN 973-594-558-4

[IVAN07] Ion IVAN, Cătălin BOJA – *Practica optimizării aplicaţiilor informatice*, ASE Printing House, 2007, ISBN 978-973-594-932-7, 483 pages.

[BOJA06] Cătălin BOJA, Ion IVAN– *Software optimization throughout simulation*, International Conference INFOBussiness ’06, Iasi

[IVAN96] Ion IVAN, Cristian CODREANU – *Optimizarea programelor assembler*, Bucharest, 1996, research report

[STEW07] Rodney STEWART – A *Framework for the Life Cycle Management of Information Technology Projects: ProjectIT*, International Journal of Project Management, 2007