Efficiency Criteria in Software Project Management

Cecilia CIOLOCA, Mihai GEORGESCU, Mihai CURTEANU
Academy of Economics Studies, Bucharest, Romania
cecilia_cioloca@yahoo.com, mihai.georgescu@europe.com, mike@synirgy.com

The modeling and development cycle of a software project is presented. Each phase of the modeling and development cycle is detailed. The activities and involved actors are identified for each phase. Software project management notions are presented. Potential risks pertaining to the software project are identified. The risks affecting the software projects are classified and each risk category is detailed. Risk avoidance and reduction strategies are identified for the identified risks. A new method is presented that defines the global efficiency for a project based on the efficiency defined for each type of actor. Global efficiency is defined for modeling information systems.
Keywords: *Efficiency, Project Management, Risk, Indicator, Software*

1 Introduction

Project management comprises the organizational structures necessary for the development of projects at organization level, the internal organization of the project and the correlation between them. It is also an integrated leadership concept. Project management implies consciously applying a coherent set of principles, rules, knowledge, methods, techniques, and instruments used in planning, organizing, leading and controlling a project.

As a result of the information sector's dynamism and of the technological evolution there is a need to integrate information systems in the management of projects pertaining to all domains and especially IT development projects. This lead to the development of tools, models and methods for the management of projects based on software systems.

A project is defined, according to [4], as a temporary effort that is undertaken in order to create a unique product, service or result. The temporary nature of projects infers a clear start and finish. The end is reached when the project's objectives have been realized or if the project is canceled because its objectives will not or cannot be accomplished, or when the project is no longer necessary.

In order for a software project to be considered a success it must be efficient. Defining the efficiency of the project is a delicate matter and several aspects of the project must be taken into consideration. Defining the efficiency is a very important phase in the modeling and development of the software product because it will represent the vision and the focus point of the entire project. This definition will be used in all the modeling and development phases of the information product. For these reasons defining an incomplete efficiency by omitting important efficiency criteria can have negative consequences for the software projects and can lead up to the failure of the entire project.

The project manager is the person who participates in and supervises the project's progress and is responsible with defining the efficiency of the project, identifying all the criteria that must be taken into consideration and making sure that they are respected throughout the whole duration of the project's modeling and development cycle. He is the one that sets the project's target. In defining the global software efficiency concept the manager must consider a multitude of factors and actors that participate and interact in the project's development process. He is also the one that has to identify the potential risks that can appear during the product's modeling and development cycle and has is in charge with elaborating damage control and safeguard strategies for the threats that these risks pose to the final product.

2 The Modeling and Development Cycle of a Software Project

The modeling and development cycle of a software project is composed of several phases. Without going through these phases it is impossible to obtain a software product that satisfies the requirements of the beneficiaries. In the domain literature, according to [1] and [2], the following stages of the modeling and development cycle have been identified:

- planning;
- system analysis ;
- system design;
- implementation and installation;
- testing and validation;
- system maintenance.

Figure 1 shows the schematic of the product modeling and development cycle's stages:

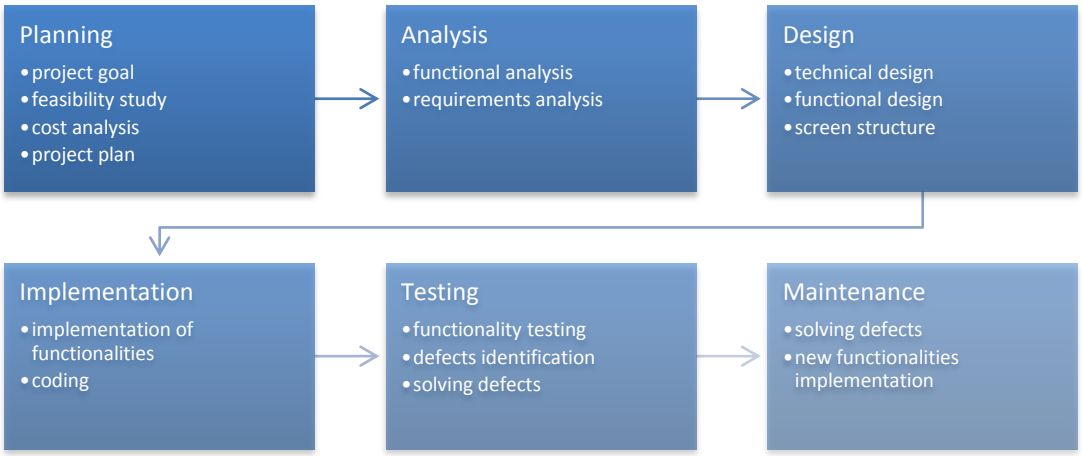

Fig. 1. Modeling and development cycle's stages for a software product

The *planning* phase is the first phase of the modeling and development process of a software product and it involves activities for defining the main requirements of the information system. During this stage a feasibility study is conducted, a risk management plan is laid out and a cost analysis is realized in order to set the budget. The most important activity of this stage is the creation of the management plan for the project. Based on this plan the resources necessary for the implementation and delivery of the software product will be acquired.

The *system analysis* phase is defined by requirement detailing activities. While in the planning phase only the major requirements of the information system were identified, requirements that define only the basic functionalities of the system, in the analysis phase all the auxiliary requirements will also be detailed. Furthermore, a technical analysis of the system will be realized in this stage in order to identify the hardware necessary for the future product.

The *design* phase is the stage where technical documentation is being built. This documentation provides a general view of the future technical system. During this stage the functionalities of the system are also detailed, the resulting documents being used in future phases. The screens of the future application and the project's diagrams are drafted in the design phase.

Using the technical documents describing the functionalities of the product that were created during the design phase the software engineers go on and work on the *implementation and installation* stage. This is the part where the new product starts to take shape. In this phase the code is being written by the programmers.

During *testing and validation* phase the application is tested and the results are validated in order to check if the system corresponds with the beneficiaries' necessities. In this stage a multitude of dysfunctions and deviations from the project's requirements are corrected. This step is necessary for ensuring a high quality

of the product that is to be delivered to the clients. For this reason this phase must be treated very carefully and seriously.

Once the software product has been created and validated, it will be delivered to the end users. During the product's usage by the end user some errors and minor deviations from the initial requirements of the projects that have escaped the testing done by the engineers are discovered. These errors must be corrected and the fix delivered to the clients in the shortest possible time. This stage is the *maintenance* phase of the information system.

In the completion of the phases of the software systems' modeling and development cycle there are several type of actors are involved. In the planning phase both the beneficiary and the project manager participate, defining the main goal of the software product as well as drafting the project's realization plan. In the project analysis phase the main actors involved are the beneficiary, the system analyst and the software architect. The last two categories also participate in the design stage of the system by building the technical documentation and the functional documentation based on the requirements they get from the beneficiary. During implementation and installation as well as testing and validation, the main role is taken by software engineers that build the software product. During these 2 stages the analyst and the software architect play a greatly reduced role. They are nonetheless present in order to offer details that are necessary to the software engineers. The last phase, maintenance, is realized through the communication between the end users and the system engineers. For this last phase the analyst and the software architect are called upon only in case of new functional requirements. These phases are discussed in depth in [8].

Figure 2 shows the degree of involvement of the actors in each stage of the information product's modeling and development cycle.

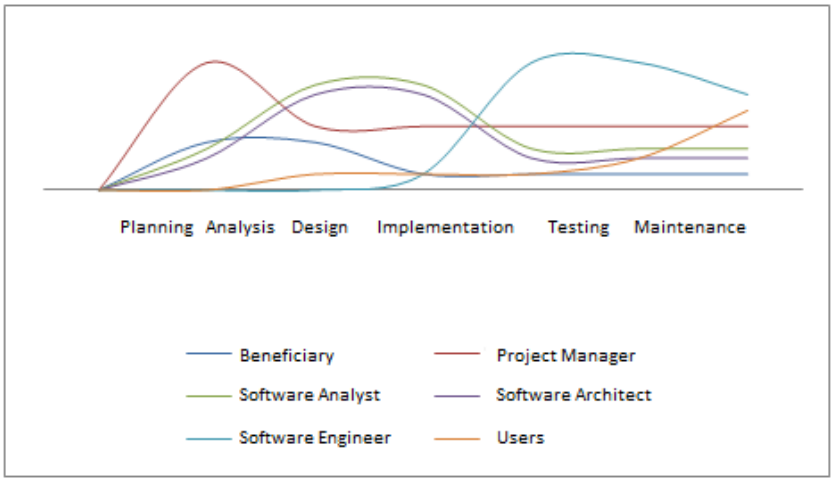

Fig. 2. The involvement of each type of actor during the modeling and development cycle of a software product

As can be seen, the only type of actor involved in every stage of the modeling and development cycle for the product is the project manager. His role in the planning phase is very intensive and he maintains a high profile in the next stages as he represents the beneficiary's interest and ensures that all stages are implemented according to the established plan. The project manager is the one that coordinates and helps the actors to achieve their established objectives by eliminating the obstacles that appear during the software product's modeling and development cycle.

3 Software Project Management

Projects use a series of concepts like: *tasks*, *milestones*, *critical path*, *PERT* and *Gantt diagrams*, all elements that are used to

represent the development plan of the project.

The company's organizational structure influences the way the projects are being managed and how the team is controlled. In a traditional hierarchy found in an organization that is divided into departments, the project manager has the least authority. On the other end of the spectrum is the project oriented organization, where all resources are organized around projects and the project manager has the authority to make decisions and implement measures regarding the project. IT organizations are project oriented inherently because of to the nature of their activity as each new contract translates into a new project that ends with the realization of a unique product that satisfies the beneficiary's requirements.

Project management is closely linked to the informational tools used due to the multitude of information sources and the large volume of work data that lead to the development of IT solutions for the management of projects.

Thus, the existent applications have transposed and automated the classical instruments of project management like Gantt charts and WBS towards collaborative applications adding tools like messaging systems, file storage, etc.

This transformation has relieved some of the workload of project managers but did bring practical developments to the field as no new tools were invented; only the old tools have been used in an innovative way. This paper proposes the creation of new methods and tools for project management by collecting and analyzing data in real time and without human intervention using the system's capacity to track the activity of each team member and collect relevant data for it in order to grade each team member and establish a hierarchy for them.

The implementation of performance indicators in IT system is a new concept that serves to enhance the efficiency of the team by establishing new control mechanisms like the ones described in [3].

Presently the focus for project management in the IT area is described by web-based solutions [10]. This trend can be explained by the large number of tools available for the project manager and the team members by these new solutions that tend to link project management with virtual teams and cooperation concepts.

According to [5] the most important element in creating a virtual team is choosing the communication and cooperation tools.

The existing web-based project management systems combine the project management part with the cooperation part to create a coherent environment in which a virtual team is able to work. The current focus is set on the IT systems' ease of use, a fact that lead the existent applications to distance them from the formal aspect of project management in order to facilitate cooperation inside the team.

The next step is to implement passive monitoring systems for the activity of the team members as such systems that can offer necessary data to the project manager about each team member without requiring written reports from the respective member but only by monitoring each one's activity together with the performance prediction functions will lead to increased efficiency in the team. Indicators for measuring the productivity of the management process are presented in [6] and [9].

The failure of software projects is the result of the multiple risks inherent to the software development environment. Software development projects are collections of larger software products, with many interactions and interdependencies. These imply the creation of something new that has not been done before, although the development processes are similar with other projects. As a result software development projects have an alarming rate of budged overruns and time delays as well as quality problems and ease of use issues. A multiple criteria decision system for software project which aids in offering options in choosing the best solution is presented in [7].

Most software projects fail partially or completely as only a small number of projects satisfy all the requirements. These

requirements relate to costs, schedule, quality and goals. The failure rate of software projects is very high, nearing 70% [IBM06].
According to [HOO09] risks are classified as follows:

- **Requirements related risks** – not reporting requirements, ambiguity, inadequate requirements, invalid requirements;
- **Cost related risks** – malfunctioning hardware, lack of testing, architecture complexity, extending or changing requirements, staff changes, changing technologies, lack or reevaluation of the management cycle;
- **Schedule related risks** – extending or changing requirements, inadequate knowledge of the tools used, long training period for the staff, lack of sufficient qualification of staff, difficulties in implementation, lack of tools, inadequate tools, lack of agreement between client and developer, lack of a good coordinator;
- **Quality related risks** – inadequate documentation, lack of development documentation, human errors, extending or changing requirements, lack of sufficient qualification of staff, faulty project estimation, lack of managerial experience, inadequate knowledge of the programming language, malfunctioning hardware, malfunctioning tools, high staff attrition, weak management, disagreements between members, architecture complexity, unclear definition of roles and responsibilities.

These are just a part of the risks that appear during a software development project and they must be handled accordingly. In order to do that it is necessary to develop metrics to be used to monitor the project's status and detect the chance of risks triggering.

4 Risk Management Strategies

Requirements Related Risks

Regarding requirements related risks, the main risk factor is that the project is lacking a solid base. Among the additional factors are situations like unclear objectives, increasing application domain, more and more requirements are introduced while the project is already in development.
Requirements related risks result from a bad situation analysis of the project, a poor understanding of the projects utility for the organization or the lack of a unitary concept inside the team and among the project's sponsors. The greatest risk of this kind is that the visions of the sponsors, of the project manager and of the team regarding the final product to be delivered are different and that this situation is identified after the project starts.

Risk avoidance strategies:

- conducting a feasibility study for the project in order to determine the expected business gain;
- associating the business case with the business standards that are presented in the functional requirements;
- validation of the business case by the sponsors, prioritizing the project based on the value of the business;
- resetting the expectations and treating the project as a prototype;
- risk analysis for fulfilling the requirements.

Risk reduction strategies:

- forcing the reevaluation of the project by the top management;
- stopping the project until a clear business case is obtained and a solid understanding of the project's benefits is gained;
- implementing an alternative defined in the risk management plan;
- analyzing the impact of the change on costs, timetable, and application domain;
- project cancellation.

Schedule and cost related risks

The budgeting and scheduling of the project contains risks like financing the whole project at the beginning thus leading to underfunding towards the end of the project, underfunding the development component, using artificial deadlines, sub financing the maintenance activities, budget deviations, etc. The main risk is represented by the fact

that the whole project must be budgeted from the start.

The strategies employed in order to avoid this problem insist that the entire budget must be approved at the beginning of the project. These include using an alternative approach for planning and budgeting projects, adding adequate emergency funds, and understanding that the initial budgets are not very accurate. The strategies for reducing the effects of underfunding a project include periodical plans and budgets reevaluation, using a change management process, continuing the risk analysis, clearly communicating the budget and schedule risks, obtaining the client's permission for additional funds.

Risk avoidance strategies:

- planning the projects in stages. Detailing only the current phase;
- laying out the plan as the project progresses;
- budgeting each phase of the project;
- planning the financing based on schedule;
- projects must not be estimated partially for each stage;
- elaborating a resource allocation plan that supports the project plan;
- understanding clearly the initial budget.

Risk reduction strategies:

- reevaluating the project by considering the costs vs. gains analysis and other options;
- verifying the financing for each phase of the project to determine the financial requirements for the next phases;
- using the change management process to approve additional financing;
- conducting risk evaluations and communicating the results to the stakeholders;
- communicating the budget and planning for the risks;
- explaining the lack of financing to sponsors and clients and asking for additional financing.

Staff Related Risks

Staff related risks include the lack of adequately skilled personnel for the project, the lack of necessary knowledge / competencies, the lack of available qualified staff, the lack of some professionals from the project team, rare meetings between the project team members and excessive use of external consultants, the most first being the most important risk factor.

Strategies used to insure the availability of sufficient adequately skilled people include setting the resource requirements, evaluating the competencies of the already allocated resources, recruiting to fill the gaps inside the project team, using roles and responsibilities to identify the areas where issues appear. In the case of an already ongoing project where problems arise with human resources, strategies include reducing the application domain if sufficient personnel cannot be added to the team. Another method would be to replace or reassign people inside the project team and to design with them ways to overcome the crisis.

Risk avoidance strategies:

- determining the project's needs for each position. Revising the already allocated competencies. Replacing the ones that are not needed;
- validating the resource requirements at the beginning of the project and then evaluating the available set of competencies;
- recruiting to fill the gaps;
- using a roles and responsibilities matrix to identify areas with issues;
- hiring persons with the required skills;
- taking in consideration qualifications requirements while evaluating the risks.

Risk reduction strategies:

- reducing requirements or having personnel delivers results;
- choosing people with experience in several types of projects;
- replacing team members or if there is enough time re-qualifying them;
- reallocating people;
- discussing with the project team how to compensate for deficits.

Quality Related Risks

The requirements for quality related risks to ensue include the existence of a disagreement regarding objectives, the lack of understanding of end user needs, not identifying risks and not drafting emergency plans, lack of use of adequate control, unclear communication lines, faulty project change management. The experts consider the last factor as being the most important.

Following technology instead of satisfying legitimate requirements can be avoided by employing strategies like conducting feasibility studies to determine the expected gain, validating the business case for the project and its owner, rejecting unrealistic projects from the start, testing new technologies before they are incorporated in the project and defining alternative approaches for satisfying the legitimate needs of the organization.

If the problems appear when a project team ignores a requirement, the top management should reevaluate the project, stop the project in case a powerful business case cannot be developed, use an alternative approach or shut down the project, if necessary.

In the case where a project team has started developing a technology that cannot support the business case, but looks promising, the top management can choose to roll this technology into a new project that targets a market where a business case can be justified.

Risk avoidance strategies:

- running preparation and organization exercises before a major project;
- allocating an experienced project manager specializing in change management and progress monitoring;
- setting a clear, concise schedule before starting work;
- setting a clear change control process;
- setting a control mechanism for the project responsible;
- requesting approval for changes from the financier;
- requesting approval for major changes from the project team;
- using and tracking the value measurements obtained following the project status evaluation.

Risk reduction strategies:

- communication with stakeholders;
- respect the rules, procedures as well as processes for change management;
- get to know the project sponsor in order to review and approve the change management plan;
- evaluating the impact of change requests on schedule, budget, ROI;
- reviewing the requirements and obtaining approvals;
- making sure that the manager and the entire team understand the process;
- adding project management resources and remaking the inadequate tasks.

Once the project is in the development phase, the agreed rules related to the management change must be followed and in case a change appear it have to be presented to the project sponsor.

5 Defining the Efficiency of IT Systems

Defining efficiency for IT systems is a complex problem and many aspects must be taken into consideration. For an IT system to be efficiency oriented it must satisfy various needs requested by various types of actors that participate in the development process and various types of users.

The concept thus defined will represent a generic efficiency that the project must attain. Each type of actor has its own expectations from an IT system based on software architecture. In [11] there are presented several strategies for obtaining an efficient architecture from the technical aspect. It is not enough for a software architecture to be efficient financially and integration wise, if the end product performances are not meeting to the end user's expectations. End users want an ergonomic IT system, easy to use and with fast results. As such, the efficiency concept must reflect the needs of different type of actors. The concept thus defined represents the central focus point of each phase of the IT system's modeling and development

process and needs to be reflected in the final product that reaches the end users.

As presented at the beginning of the paper, the actors participating in the modeling and development process of IT systems are: the project's beneficiary, the project manager, the system analyst, the software architect, software engineers, and end users. The IT system efficiency is defined below according to each type of actor's point of view.

The project's beneficiary seeks a financial efficiency of the project. He is directly influenced by the project's budget and wants the best possible quality/price ration. Also the financial efficiency is represented by the ease of use by the end users. The beneficiary wants a quality product, feasible, that makes work easy for the users, which is ergonomic and fits the budget. Another very important aspect for the project's beneficiary is the time it takes for the product to become available to the end users. In most cases the beneficiaries want the products to be available in the shortest amount of time possible. This often leads to lower product quality.

The project manager is considering the beneficiary's and end users' satisfaction as well as financial efficiency and the company's profit. From his point of view the project is efficient if the project goal is reached, if the product quality is high and if the delivery deadline is met.

From the point of view of **the system analyst** the IT system is efficient if it is an ergonomic product, easy to use for the end users and if the functionalities requested by the beneficiary are implemented.

The software architect tracks the software system's performances from a technical point of view. For him the software's efficiency is represented by scalability, feasibility, response time, security and IT system integrity.

For **software engineers** the code's reusability, the quality of the code, the use of design patterns that help write fast code, the systems extensibility are important factors. They are also interested in the ease of adding modifications to the project, thus requiring a solid documentation of the system and of the code.

End users are interested in the aspect of the graphical interface as well as the response time performance of the system. They want an application that has a pleasant design, intuitive interface, is easy to use and is capable of responding in the shortest possible time. Once these characteristics are satisfied they will be able to perform in their daily tasks.

In order to define the general IT system efficiency concept the previously defined partial efficiencies will be used. The global efficiency must reflect all the identified actors. Such a definition will be used in all phases of the modeling and development process for the software architecture and represents the final goal of the process. The activity of generalization and defining of the global generic efficiency is very delicate. There are situations in which conflicts arise between different types of actors. The efficiency for one type of actor conflicts with the efficiency of another type of actors. To solve this problem a value scale will be defined with the help of which each type of actor and its efficiency needs will be prioritized. Depending on the priority of the actor type and its efficiency requirements, the existing conflicts will be solved and the efficiency criteria that will be taken into consideration when defining the global concept of efficiency will be chosen.

It is evident that there are efficiency criteria that are contradictory. Availability in the shortest amount of time possible can damage the IT system's quality. In the pursuit of a fast product in most cases the end product has poor quality and performances. Taking this into account, one of the contradicting criteria will be eliminated. A value scale from 1 to 5 will be used, where 5 represents maximum importance and 1 minimum importance.

Table 1 – Efficiency criteria priorities

Efficiency criteria	Priority
IT system quality	4
IT system performance	5

Quick product availability	2

Table 1 show that quality and performance have a higher priority for the IT system than the need for quick availability. Thus the least priority criteria will be eliminated and the availability in a short time for the product will be excluded from the definition of the global efficiency concept of the IT system.

Figure 3 presents schematically how to define the global efficiency by using the efficiencies defined for each type of actor:

Fig. 3. Global efficiency of IT systems

The global efficiency of an IT system is defined as the multitude of efficiency criteria that contains all the efficiency criteria of all types of actors. The efficiency thus defined contains the interests of all the actors and will be used during the whole modeling and development process for the IT system. The global efficiency is the target and focus of the entire software product.

6 Conclusions

Defining efficiency in software project management is a very important task that must be approached seriously. Identifying all the efficiency criteria that form the global efficiency has real benefits towards the success of the software project. Through the detailed analysis of all the actors and their necessities a superior quality software project is obtained. The satisfaction of users and beneficiaries brings a long term relationship and a rise in the company's profits. Also, satisfying the software specialists' efficiency criteria reduces the company's costs related to the initial functionalities' development and maintenance. The correct definition of the global efficiency results in a healthy relationship between actors and ensures the development of a robust and quality software product.

References

[1] K. Roebuck, *Software Development Life Cycle (SDLC)*, Tebbo Publishing, 2011, pg. 398, ISBN 978-1743045756

[2] J. Lewis, *SDLC 100 Success Secrets*,

Emereo Publishing, 2008, pg. 184, ISBN 978-1921523151

[3] E. W. N. Bernroider, M. Ivanov, "IT project management control and the Control Objectives for IT and related Technology (CobiT) framework," *International Journal of Project Management*, vol. 29, nr. 3, 2011, pg. 325-336, ISSN 02637863

[4] *A Guide to the Project Management Body of Knowledge (Pmbok Guide)*, Project Management Institute, 2008, pg. 459, ISBN 1933890517

[5] R. Ubell, *Virtual Teamwork: Mastering the Art and Practice of Online Learning and Corporate Collaboration*, Wiley, 2010, pg. 268, ISBN 0470449942

[6] P. Pocatilu, "IT Project Management Metrics," *Informatica Economica*, vol.11, no. 4, 2007, pg. 122-125, ISSN 1453-1305

[7] G. Marques, D. Gourc, M. Lauras, "Multi-criteria performance analysis for decision making in project Management," *International Journal of Project Management*, vol. 29, nr. 8, 2010, pg. 1057-1069, ISSN 02637863

[8] K. Schwalbe, *Information Technology Project Management, Revised, 6th Edition*, Course Technology, 2011, pg. 704, ISBN 1111221758

[9] P. F. Rad, G. Levin, *Metrics for Project Management: Formalized Approaches, Management Concepts*, 2005, pg. 384, ISBN 1567261663

[10] R. F. Grove, *Web-Based Application Development*, Jones & Bartlett Publishers, 2009, pg. 608, ISBN 0763759406

[11] I. Gorton, *Essential Software Architecture*, Springer, 2011, pg. 242, ISBN 3642191754

Cecilia CIOLOCA has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2008. She is currently conducting doctoral research in Economic Informatics at the Academy of Economic Studies. She is experienced in software development using open source technologies. She has successfully completed several projects using various programming languages and open source frameworks. She is proficient in using Java Enterprise Edition.

Mihai GEORGESCU has graduated the Faculty of Electronics, Telecommunications and Information Technology – Politehnica University Bucharest in 2005 and also completed a master program at the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2007. He has experience in the development of software products using various programming languages like C/C++, C# both Windows Forms and Web.

Mihai CURTEANU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2008. He obtained a Master's Degree in Project Management in 2010 with a paper on Online Project Management Applications. He is a PhD student of the Doctoral School of Bucharest Academy of Economic Studies in the field of Economic Informatics.