Distributed Collaborative Software Development Process Improvement Using Criticality Analysis

Marius VETRICI[1], Cosmin TOMOZEI[2]
[1]Bucharest Academy of Economic Studies,
[2]University of Bacău
mariusvetrici@softmentor.ro, cosmin.tomozei@ub.ro

Collaborative software proves to be a key-element in achieving common goals in many areas of activity, such as education or healthcare. The aim of this article is to provide a reliable framework for building collaborative portals for education. These portals represent virtual places for student cooperation and community build-up in order to create reliable and efficient projects, guided by the teachers from the Computer Science Departments and optimized using Criticality Analysis. This paper presents a new software project duration estimation model for enhanced project risk assessment and quality control in the field of distributed collaborative project development. The Extended Metrix model is a stochastic model for software project duration estimation and risk analysis using Monte Carlo simulation over an activity graph. The model is endowed with Criticality Analysis features for enhanced control of the project development process.
Keywords: *collaborative software, distributed, analysis, process development*

1 Preliminary Considerations regarding Project Duration Assessment

The vast majority of software development projects are known to be unsuccessful in terms of project delivery time and project budget. A wide range of surveys performed during the last 15 years reveal a relatively poor performance in delivering successful software projects. Most of the projects hit schedule and budget overruns of 25% to 100% and sometimes even more [1], [2], [3]. The adoption of Agile and Iterative development methods provided us with a raise in project success rate from one project in three [2] to two successful projects from three [4]. Still, there is place for improvement.

This paper will be focusing on timely delivery of the Collaborative Software Projects as a measure of project success. The precondition for defining an accurate project delivery date is a precise estimation of the project duration. Existing estimation models are rather imprecise because the forecasted value is to a certain extent distant from the real one. The large divergence between the estimated duration and the actual schedule of the ongoing projects prematurely ended them in order to prevent further damages and losses [5], [6]. Given this, it is imperative to look for new methods that will aid software project managers in forecasting and controlling project duration and, hence, the project quality [7].

The aim of this research is to bridge the gap between the forecasted software project duration and the actual project duration.

Earlier research [8] proposes a stochastic user friendly model called Metrix model suited for software project duration estimation using Monte Carlo simulation over an activity graph. This paper will continue the research by embedding project risk analysis steps into the existing Metrix model. The new resulting model is called Extended Metrix model and will provide project manager both with project duration probability distribution and with extra information regarding the criticality of every project task.

2 Collaborative Systems in Education

Collaborative software appears as an entity that supports people that are usually placed in different geographic areas, but with similar concerns about a specific domain. Collaborative work proved to offer efficiency and success in achieving difficult tasks, reuniting the experience of specialists, gathering their knowledge and expertise in a very short period of time. Our concern is to examine and implement collaborative work in higher education, with the aim of supporting students or young researchers with precious information for their teaching, learning and research activities. Our desire is to create and maintain a virtual community between professors from universities, young researchers and students.

The concept of collaborative systems is based on *computer – supported collaboration*, which presumed the evolution from the traditional cooperative work study and the support of the people collaboration in work activities and relationships. In [9] we have the concept of *computer supported collaborative work* (CSCW) as a multidisciplinary research field involving the way in which the collaborative activities are to be accomplished and in which way their coordination can be supported by means of computer systems. Our main idea is to analyze, make comparisons, validate and exemplify how collaborative software affects the quality of the teaching and learning process in higher education.

On the other hand, distributed systems provide the appropriate framework for the building and development of collaborative systems. Even though in this case the common goal or the objectives of the users, also referred to as agents, is the most important factor, the framework on which the collaborative software application is built and running cannot be ignored. As a consequence, the distribution of data and software becomes key elements in application development and data management. Still, another important aspect is related to the software quality, meaning that information has to be found very easily and with minimum errors and redundancy.

Collaborative software and collaborative work permit that users have *social interactions*, by using social conventions and rules, in order to communicate and build virtual teams. Social computing presumes that any kind of social behavior is supported by the computer system. In addition to social computing, social information processing is an activity that describes the way in which the human knowledge is organized by collective human actions. In the medical domain, such as surgery [10] knowledge exchange and common experience allow to specialists from remote places to cooperate and reduce the duration of procedures, by creating virtual teams and working together for helping the same patient concurrently.

In our case, the software application we created allows teachers to help the students, also by making virtual teams each of them having a common goal, to increase the efficiency and quality of higher education. Each student has a web account with username and password, as well as all the teachers from the department. Common tasks, such as project development become easier to accomplish by web-uploading each version of any project, which may be reviewed and modified by the other members of the team, depending of the reputation and credentials each member has. There are some restrictions, which require the member who modified any particular project to notify the person which initially posted it, and get his permission. Each user action is well monitored and if it is necessary, the system will reduce or diminish the access, rights or reputation the user has in the social computer – mediated community. However, this cases are very rare, the grand majority of the people who use in their daily activities the educational collaborative system being focused mainly on achieving the scientific goals.

3 Educational Portals as Collaborative Software Projects

As in all social systems [11], the virtual reality constructed thanks to *Computer Mediation*, different people interact sharing their knowledge and experience. In our situation, the educational portal built for the students and teachers in *The University of Bacau*, where the teachers place learning material including eBooks, seminar material, laboratory work and various examples for students to access from distributed geographic locations. Each student has a unique username and password in order to authenticate himself in the system. After the authentication process [12], the person is able to get in touch with the colleagues, to look for a specific material or place his own. All of them form a virtual community, which mirrors the school community, working in addition to the academic schedule, and try to provide a higher level of efficiency to the education process.

Portlets proved to be the most intelligent way of presenting information on education portal, due to the richness of the user interface tools. The portal may be found at http://cursfs.ub.ro URL. In figure 1 there is a screenshot of the 2nd year students section in educational portal, which is very easy to use and helpful.

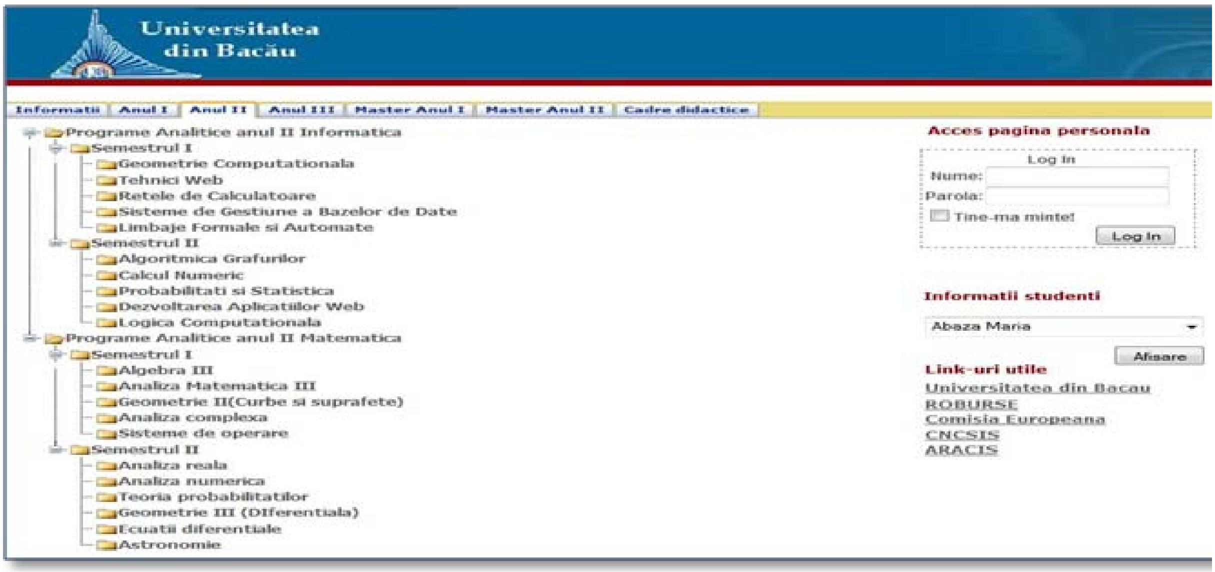

Fig. 1. 2nd year section in the Educational Portal

Despite the divergence of each member's individual interests and structure, the media as well as the Internet become new forms of social binding, and we believe they do not exclude the traditional ways of communication and interaction, but help people in their duties, in addition to the traditional, formal ways. In Figure 2 [13] there is a clear image about how each of the stakeholders interacts with the educational portal.

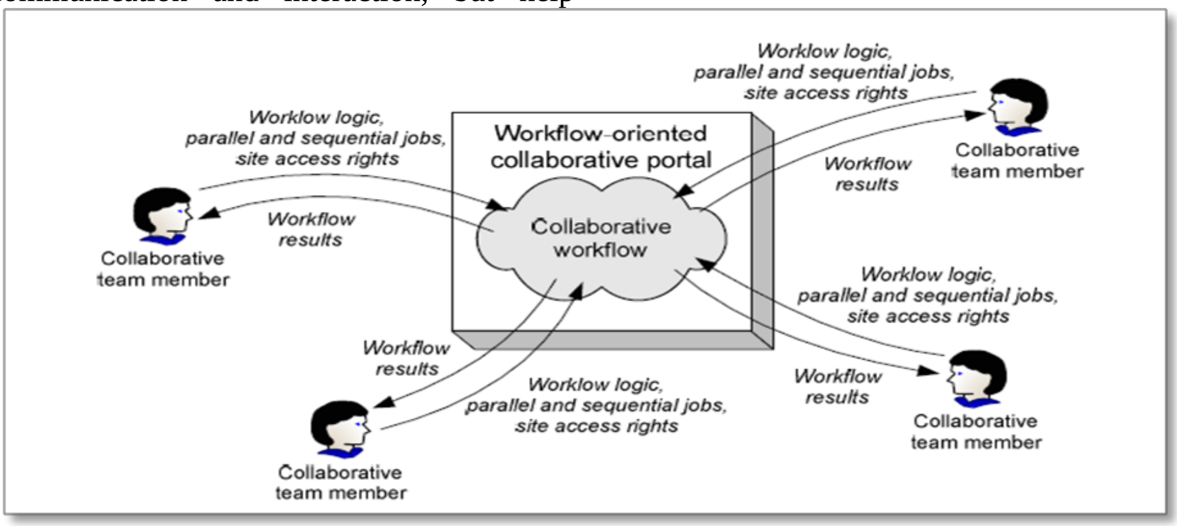

Fig. 2. Workflow in Collaborative Portals

As we have mentioned before, depending on the job, rights and reputation, every member of the collaborative virtual team interacts with the system. He is a beneficiary of the results provided by the system, which allows cooperation with his colleagues. Common tasks are greatly achieved and the duration [14] between the decision of realizing a task decreases significantly.

As a complementary theory alongside the software systems *constructivism* may be seen as a way to improve the efficiency of the education process, and guides the students, helped by the collaborative portal into building their knowledge with their own instruments.

4 Project Duration Estimation Prerequisites

A *project* is "a temporary endeavor undertaken to create a unique product, service, or result" [15]. A *software development project* is a temporary endeavor undertaken to create a unique piece of software [16]. The *duration of a project* is the time elapsed between the project start and the project delivery date. Project duration is only available after the project has been finished [17]. *Estimated project duration* is the calculated approximation of the project duration. Even though estimated project duration is determined using uncertain and incomplete information, however, this is an essential indicator that should be agreed upon with the stakeholders and thoroughly monitored, up to the project completion. *Project task* is a small division of a project representing an activity that needs to be accomplished within a predefined amount of time in order to meet project goals. *Project task duration* is the time elapsed between the starting and the ending moments of a task. Project task duration is only available after the project has been finished. *Estimated task duration* represents the calculated approximation of the duration of the project task. A *Project network* is a graph (flow chart) depicting the sequence in which project tasks are to be executed. See figure 1 for an example of project network. *Project Critical Path* represents the longest path in a project network, from start to end activities. The sequence of activities on the critical path adds up to the longest overall duration which represents the shortest time possible to complete the project. *Task Criticality Index* represents the ratio of the number of times an activity is on the critical path to the total number of Monte Carlo simulation trials [18], [19]:

$$TC = \frac{\sum_{i=1}^{N} TC_i}{N}$$, where: (1)

TC – task criticality, a number between 0 and 1 inclusively.

TC_i – equals 1 if task is on critical path at iteration i and 0 otherwise.

N – the total number of Monte Carlo simulations.

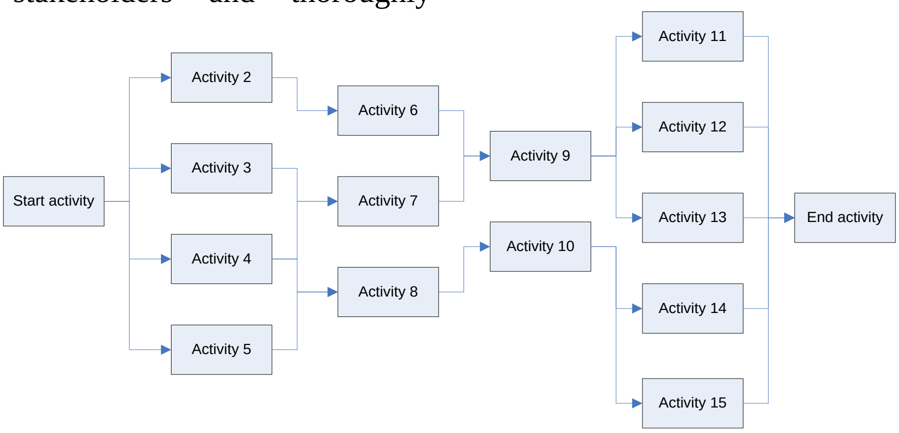

Fig. 3. Project network example

5 Determining task criticality using Extended Metrix Model

The Metrix model is a hybrid type of model for estimating the duration of software projects [20]. This is a stochastic model that addresses the project duration uncertainty by running Monte Carlo simulations over the activity graph. The Extended Metrix model incorporates project schedule overrun risk analysis by the means of task criticality analysis. More exactly, the Extended Metrix model counts the number of times every project task occurs on the critical path for every iteration of Monte Carlo simulation and then computes the task criticality index.

In the end, the user is presented with:

a) A list of project tasks, ordered in the descending order of the criticality index.
b) A probability distribution of the project duration estimation.

As follows, the Extended Metrix model structure and the steps it encompasses are presented in greater detail.

Individual task duration estimations and task interdependency represent the input data of the model. The model will also get the history of the duration estimations for the tasks of the projects finished earlier by the same development team.

The steps performed are described here under:

Step 1. The historical task duration estimations are collected for every developer. Both current project finished tasks and the tasks finished in other projects during the last 6 months will be considered.

Step 2. For every historical task duration estimation from step 1 we calculate the Estimation Accuracy Index (EAI) using the following formula:

$$EAI = \frac{AD}{ED} \quad (2)$$

, where:

ED – estimated task duration (in hours);
AD – actual, elapsed task duration;
EAI – Estimation Accuracy Index.

If EAI is greater than 1, then the task was overestimated, meanwhile if EAI is less than 1, then the task has been underestimated. Using the results above we calculate the discreet probability distribution for the EAI indexes for every developer part of the team.

Step 3. We build the activity graph using the task dependency and estimated task durations.

Step 4. We find the critical path through the graph and we calculate the deterministic duration of the software project.

Step 5. We run the Monte Carlo simulation. The following operations are performed at each iteration i:

a) For every task we choose (using the probability distribution from step 2) an estimation error from the same developer's estimation error history. Then we adjust the actual estimated duration with this EAI.

b) We recalculate the critical path and the project duration.

c) We count every task appearance on the critical path, i.e. TC_i.

We repeat the simulation N times, where N is greater than 1000.

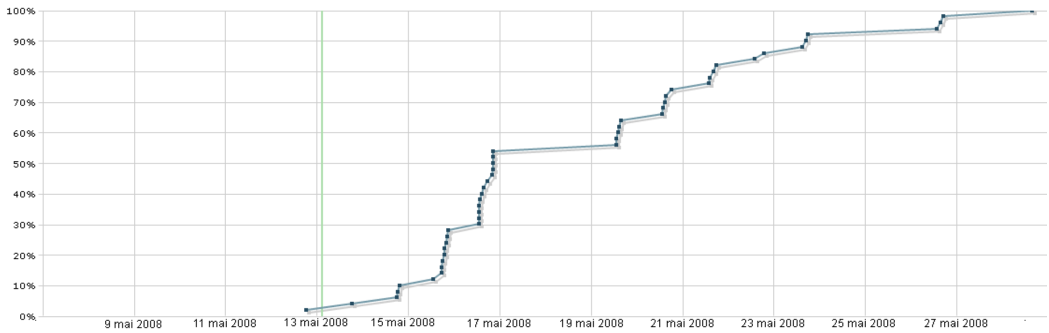

Fig. 4. Probability distribution for project duration and project deadline

Step 6. We calculate the project duration frequencies obtained as a result of Monte Carlo simulation. The project duration probability distribution can be seen in figure 2.

Step 7. For every task project, we apply formula (1) in order to obtain task criticality. The risk of delaying the whole project is directly proportional to the criticality index of every task. The higher the task criticality index is, the greater the risk that the entire project will be delayed if that task is delayed. See table 1 for example, listing top 5 tasks in order of their criticality.

Table 1. Top 5 tasks of a software project in descending order of their criticality

Task code	Task Criticality Index
15	0,91
2	0,87
7	0,85
21	0,84
6	0,82

6 Applying the Metrix Model in Collaborative Software Project Development

Our main contribution is the application of the Extended Metrix Model, developed during earlier research [21], [22] to Collaborative Software Platforms guided by the idea that the model becomes an appropriate way to get things done in a reasonable time and with an increase in efficiency. The successfulness of the distributed applications development is proved to be in relation with the Metrix Model.

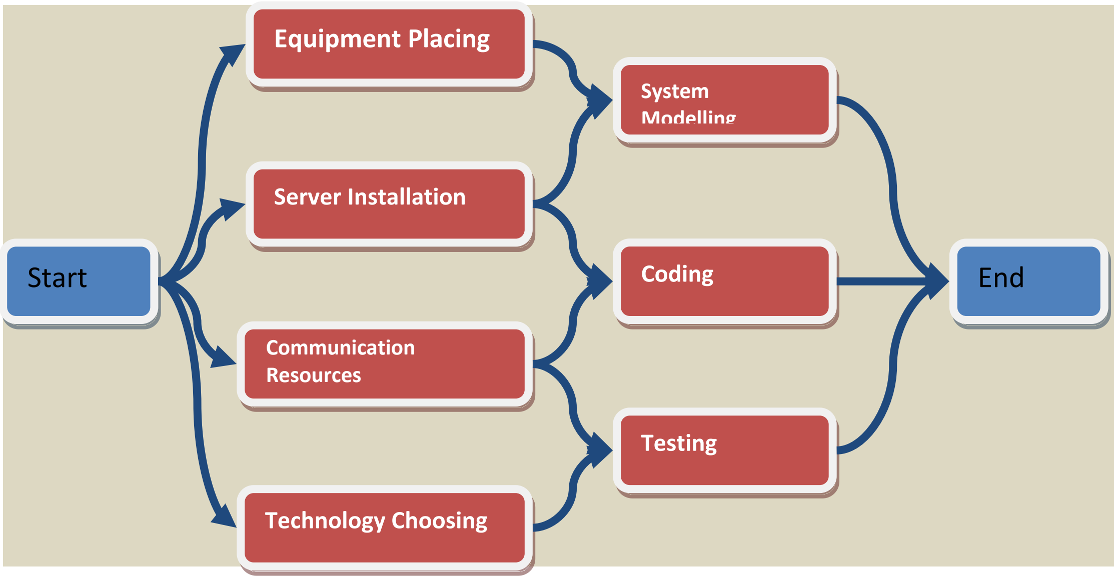

Fig. 5. Collaborative system project activity graph

The entire project has been divided in tasks and the development team has been assigned with tasks for each member. We determined the following categories of tasks:

- regarding the server side and operating systems administration;
- regarding the placement of the equipment;
- regarding the gathering and acquisition of the communication resources such as DNS, IP addresses;
- regarding the selection of the a appropriate development platforms, environments and Database Systems;
- regarding the domain analysis, modeling, and determination of diagrams for capturing the design of the software system, including the architecture;
- regarding coding and programming;
- regarding the future legacy and possibility of achieving new goals which will be determined in the future;

We calculated the EAI Index for each task, some proved to be overestimated and some underestimated. After that, we built the activity graph, which is described in figure 4.
Each activity from the activity graph may be divided into smaller activities, creating subrgaphs. We have just presented the entire project main activities, but each one of them represents a reunion of smaller activities. For example, the System Modeling presumes the construction of a sets of diagrams, both static and dynamic. Each class is represented by an entity from the class diagram and each activity is described as well in the activity diagram. Each process is presented as well as their order in the sequence diagram.
We finally applied Monte Carlo Simulation and obtained the distribution from the figure 5 and obtain each task criticality.

7 Conclusions

The first benefit of the Extended Metrix model is the project risk information associated to every task. The uniqueness of the proposed model is that it determines both the estimated duration of the project and the risks associated with delaying a task.
It is also very important that the Metrix model relies on the historic duration estimation of the team members. Similar models based on Monte Carlo simulations require a duration probability distribution function for every task. This requirement unfortunately set Monte Carlo simulations out of the practical domain into the academic universe. The innovation brought by the Metrix model is the elimination of the probability distribution functions requirement and the use of discreet probability distribution of the EAI. The EAI probability distribution can be easily determined using the historical estimation errors which are available to most software companies.
Further research will aim at implementing the Extended Metrix Model in the field of cloud computing software development, by minimizing the entire project duration and by calculating each task's criticality. The probability of success increases significantly and the risks are reduced to a minimum.

References

[1] A. W. Chow, B. D. Goodman, J. W. Rooney and C. D. Wyble, "Engaging a corporate community to manage technology and embrace innovation," *IBM Systems Journal*, Vol. 46, No. 4, 2007.
[2] ***, *The Chaos Report of IT Project Failure*, Standish Group, 2006.
[3] S. McConnell, *Rapid Development*, Microsoft Press, Washington, 1996.
[4] S. Ambler, "Software Development Project Success Survey 2008," *Dr. Dobbs Journal*, February 2009.
[5] I. Ivan and A. Sandu, "Projects Hierarchy Based on Duration and Complexity," *22nd IPMA World Congress Proceedings - "Project Management to Run"*, Italy 2008.
[6] I. Ivan, A. Visoiu, C. Ciurea and D. Palaghita, "Model bases and software quality metrics refinement," *The 4th International Conference "Economy and Transformation Management"* 2008, Timisoara, Romania.
[7] I. Ivan, C. Boja and D. Milodin, "Orthogonality level optimization for distributed informatics applications," *4th Intl. Conference on Applied Statistics*, Bucharest, 2008.
[8] M. Vetrici, "Project schedule using Monte Carlo simulation with discreet probability distribution," *Proceedings of the 4th Intl. Conference on Applied Statistics*, Bucharest, 2008.
[9] M. Szymanski, P. Aok and M. Grinter, "Sotto Voce: Facilitating Social Learning in a Historic House," *Computer Supported Collaborative Work*, Vol. 17, No. 1, February 2008, pp. 5 – 34.
[10] I. Ivan and C. Ciurea, "Quality Characteristics of Collaborative Systems," *The Second International Conference on Advances in Computer-Human Interfaces*, ACHI 2009, 1-7 February 2009, Cancun, Mexico, pg. 164-168.
[11] B. Becker and G. Mark, *Constructing*

Social Systems through Computer-Mediated Communication, Virtual Reality, No. 4, Springer-Verlag London, 1999, pp. 60-73.
[12] I. Ivan, M. Doinea and D. Palaghiță, "Optimization of authentication processes in distributed applications," Theoretical and Applied Economics, No. 6, June 2008, pp. 39 - 56.
[13] G. Sipos, G. J. Lewis, P. Kacsuk and V. N. Alexandrov, *Workflow-Oriented Collaborative Grid Portals*, P.M.A. Sloot et al. (Eds.): EGC 2005, LNCS 3470, pp. 434 – 443, 2005, Springer-Verlag Berlin Heidelberg 2005.
[14] M. Vetrici and C. Ionita, "Knowledge Based Project Duration Estimation for Workflow Based Document Management Software Projects," *Proceeding of Knowledge Management Conference*, Bucharest 2008.
[15] ***, *A Guide to Project Management Body of Knowledge Third Edition*, Project Management Institute, 2003.
[16] M. Vetrici, "Reducing Software Projects Duraton Using C#," *Informatica Economica Journal*, Vol. 7, No. 1, 2007, pp. 91-95.
[17] C. Tomozei, „Hypertext Entities Semantic Web-Oriented Reengineering," *Journal of Applied Quantitative Methods*, Vol. 3, No. 1, 2008, pp. 9-19.
[18] Y. H. Kwak and L. Ingall, "Exploring Monte Carlo Simulation Applications for Project Management," *Risk Management Palgrave Journals*, Vol. 9, 2007, pp. 44–57.
[19] T. Williams, "The contribution of mathematical modeling to the practice of proj. mangment," *IMA Journal of Management Mathematics*, Vol. 14, No. 1, 2003, pp. 3-30.
[20] M. Vetrici, "Software Project Duration Estimation Using Metrix Model," *Informatica Economic Journal*, Vol. 12, No. 2, 2008, pp. 87-91.
[21] M. Vetrici, "Improving software project quality using criticality analysis," *The IX-th International Conference on Informatics in Economy*, Bucharest, 2009.
[22] P. Pocatilu and M. Vetrici, "Schedule risk management for business M-applications development projects," *WSEAS Transactions on Computers*, Vol. 8, No. 4, April 2009, pp. 735-745.

Marius VETRICI is an independent researcher in the field of software project management. He is a PhD candidate since October 2007 at Economic Informatics Department from Bucharest Academy of Economic Studies. Mr. Vetrici holds a MSc degree in Project Management from Bucharest Academy of Economic Studies and a BSc degree in Economic Informatics from the Cybernetics, Statistics and Economic Informatics from the same university. His main research areas are: software project management and document management

Cosmin TOMOZEI is University Assistant - Lecturer at Mathematics and Computer Science Department from Faculty of Sciences of the University of Bacau. He is a PhD candidate from October 2007 at Economic Informatics Department from University of Economics, Bucharest. He holds a Master in Science - Databases- Business Support from University of Economics, Bucharest. He graduated in Economic Informatics at Faculty of Economic Cybernetics, Statistics and Informatics in 2006. His main research areas are: object oriented programming, functional programming in Lisp and F#, software reengineering and distributed applications development.