Teaching Requirements Analysis: A Student Project Framework to Bridge the Gap between Business Analysis and Software Engineering

Luisa Mich
Dept. of Industrial Engineering
University of Trento
Trento, Italy
{luisa.mich}@unitn.it

Abstract*—Teaching requirements analysis to computer science and information system students raises a number of challenges. One of the most critical is the gap between skills needed to deal with software requirements and those necessary to grasp the business problems. To bridge that gap in teaching requirements analysis students would have to carry out an assignment of analyzing requirements for a non-trivial, term-sized project. Here we analyze the gap and propose a framework for student projects which integrates a model of the computer based system as a solution to business challenges into a template for a business requirements document. The first model comes from information systems literature and the second from an object oriented analysis approach for business analysis. A CASE (computer aided software engineering) tool to support UML (unified modeling language) modeling is also used and we give some guidelines to reduce risks of premature requirements modeling due to students' tendency to start modeling, even if business analysis and requirements elicitation have just started. The proposed framework has been defined in many years of teaching and allowed to overcome some of the limitations of a traditional UML-focused course. Student projects of different academic terms – in different courses and different degrees – showed improved requirements models and better comprehension of the role of requirements in the later terms. Moreover, the students appeared to have greater interest and motivation towards this area of software engineering.

***Index Terms*—requirements elicitation, UML, business object-oriented modeling, business requirements**

I. Introduction

The term 'software engineering' dates back some forty years ago, and 'requirements engineering' is more recent, but for establishing a discipline in a university degree curriculum it could be a short time.

In industry, even if many studies have reported the critical role of requirements for the success of computer based systems, requirements analysis and modeling are not always part of the software development projects (see, e.g. the Standish-Group reports http://blog.standishgroup.com). Dedicated international conferences, such as the Requirements Engineering (RE) series, which started in 1993 (http://requirements-engineering.org), or the REFSQ working conferences (Requirements Engineering: Foundation for Software Quality, www.refsq.org), among others, do include in their programs industrial tracks and invited speeches of representative of companies. All these initiatives confirm the vitality of the discipline and a high interest for requirements from both industry and academia.

As regards the educational contexts and challenges as a *per se* subject, requirements engineering education and training is the focus of the REET (Requirements Engineering Education and Training) workshops (www.ics.uci.edu/~bpenzens/2014reet). Of the papers published in the CSEE&T proceedings (Conference on Software Engineering Education and Training, conferences.computer.org/cseet) only a limited number are about requirements – in average, one a year – and most of them analyze specific issues, e.g. security requirements [1] or the problem of prioritization of requirements [2]. Nakatani et al. [3], propose the design of a requirements engineering course for senior engineers. A comprehensive survey of the current research on requirements engineering education is given by Ouhbi et al. [4].

Almost all the methods of teaching computer science agree on the need to adopt an activity based approach, that is, what in [5] is called an 'active-learning-based teaching model'. In requirements engineering courses, it also implies the definition of a suitable student project. Many authors underline that a real industrial project would offer the best educational context. However, didactic and organizational constraints do not always allow proposing this kind of project in university courses. Other educational approaches, e.g., the one described by Fernandes et al. [6], introduce a curriculum for software development professionals with experience in developing real software solutions. But this is not the case for university students. Even students of a computer science degree do not always have experience in real projects. Another problem in teaching requirements analysis is due to a low interest in the subject: real projects require a comprehension of the company or organization domain and of their business challenges. For students in computer science and information systems, these are – for different reasons – not primary subjects of interest.

The purpose of this paper is to report an approach for student projects which has been defined in about twenty years of teaching in university courses to both software engineering and information systems students. The approach is based on a framework that allowed to overcome some of the limitations of a traditional UML (unified modeling language) focused course.

The project foresees the accomplishment of business and requirements analysis activities for a quasi-real project, that is for a project in a real context, whose characteristics satisfy time and other constraints of an academic course. Students groups have to realize and document the project following a template which integrates a model of the computer based system as a solution of business problems into a structured business requirements document. The first model comes from information systems literature [7] and the second from an object oriented analysis approach for business analysis [8].

In the rest of this paper, Section II describes the main challenges in teaching requirements engineering. Section III presents the template for a student project that allows addressing such challenges, gives some remarks and guidelines to apply the project framework and a preliminary validation referring to the guidelines illustrated in [4]. Section IV concludes the paper with an outlook on open questions and future work.

II. Challenges in Teaching Requirements Analysis

Teaching requirements analysis is challenging in many ways. The main positive and negative aspects, some of which have been mentioned in the introduction, can be illustrated with a SWOT (Strengths, Weaknesses, Opportunities, and Threats) matrix (Fig. 1). As is often the case, some of the strengths are such only if accepted as stimuli by the lecturer.

Firstly, requirements analysis is a multidisciplinary activity that employs a variety of methods and techniques at different stages of development of computer based systems. Teaching requirements engineering demands further efforts to take into account the business analyst's and the software engineer's points of views. In particular, the scope of a project can be correctly defined only if the business goals and challenges have been identified and analyzed and this requires skills beyond requirements modeling and specification [8], [9]. A multidisciplinary subject naturally induces to address problems in a rational way, promoting computational thinking [10] and group work to exploit cooperation among people with different expertise [11].

Another challenge in teaching requirements analysis is its recent development as an autonomous research and educational area, at least if compared to other areas of computer sciences. This implies that many graduation programs do not include a dedicated requirements analysis course. Being part of the software or information systems development process, requirements engineering is often taught (only) in those courses. In this way, usually a limited number of hours can be dedicated to requirements analysis, with the realization, if any, of toy projects. In turn, this fact sometimes implies that the assigned textbooks cover the information systems or software engineering content, whereas requirements are a small fraction of the content. Besides, while for students of computer science degrees requirements are perceived as not enough 'computer science', for students of information systems, requirements are too much 'computer science'. These views are some of the most critical experienced in about twenty years of teaching and could be due to the common view of the field as programming.

Strengths	**Weaknesses**
Multidisciplinary field Promote problem solving (computational thinking), and group work	Image problems Time constraints Students lack of interest Toy projects
Opportunities	**Threats**
Interest of companies for business analysts skills Integration of real projects Coordination with other courses	Not dedicated course Misconception of the field Lack of books for educators Tool driven analysis

Figure 1. SWOT matrix for teaching requirements analysis

This long-standing prejudice has been widely investigated [12]; authors in [5] underline that to overcome it, students have to be taught what computer science is, its nature and definition. An analogous factor that negatively impacts the quality of requirements engineering courses is the tendency of the lecturers themselves to emphasize specification on analysis, probably for the same reasons highlighted in industrial projects [9]. Also, in the author's experience, computer scientists somehow consider functional specification and software development as a kind of 'real' computer science, under-estimating business requirements analysis.

Another problem arises when looking for textbooks dedicated to 'Didactics of requirements engineering', or 'Didactics of software engineering'. You will not find many references and there are not many for the didactics of computer science either. Lecturers can design their requirements courses according to the corresponding subset of the software engineering model curricula, or to syllabus of certification that focus on requirements (e.g., the Certified Professional for Requirements Engineering, www.ireb.org). However, as curricula, they (have to) describe the 'what' – the content to be taught – and not much is said on the 'how', that is on the didactic methods more appropriate to teach that content. Finally, when teaching requirements analysis, the choice of the modeling language is almost straightforward if you adopt the de-facto standard, UML (http://www.uml.org). The problem is that students of information systems often have learned and applied other languages to model business procedures and goals (e.g., the BPMN, Business Process Model and Notation, http://www.bpmn.org). A seamless approach would suggest using UML models also for those modeling activities, a solution that depends on an agreement with other lectures. UML diagrams can be realized with automatic tools: there are many of them and the choice of the one to be adopted for a course is not simple. Beside technical and cost factors, there are features that force the students to follow what could be called a depth-first instead of a breadth-first modeling approach. For example, one of the most common of such features is related to the opening of a form suggesting adding characteristics to a class or to a use-case at the moment of their very first creation. The form compels the students to focus on details, such as, for example, values of the attributes for a class, before the overall analysis of the class model has been completed with all the classes. At the same time, focusing on UML, teachers do not always exploit the problem-solving nature of requirements analysis and do start introducing UML models for small projects to explain the notation and the semantic of the

language. However, such models are of little interest to students, as it is not easy to be passionate, e.g. to library lending systems.

Focusing on opportunities, it is important to underline that companies look for business and requirements analysts (http://www.linkedin.com/job/q-business-analyst-jobs). As student projects require limited tools and resources, they can be useful to experience such roles. Also, the need to integrate expertise in different areas could trigger shared lectures and projects with other courses.

III. Designing a Project for Teaching Requirements Analysis

A. The Proposed Approach

To address the issues in the SWOT matrix for a university course in requirements analysis would require the design of the course taking into account a large number of factors and constraints. Among them, the syllabus of the degree, available human resources, time available for the course, the kind of classroom, hardware and software available, technical support, topics to be covered, students' previous knowledge on related topics, etc. In this paper we will concentrate on the problem of the student project: which project can be realized by students to motivate them and to teach a structured, efficient, and systematic way to identify and model business and high level requirements. In this section we propose a framework for a student project that has been defined adapting it year after year according to the results of its implementation in a number of university courses. In particular, in teaching requirements analysis to students of information systems, software engineering and information systems design. These courses were part of different kinds of degree of an Italian university, at the bachelor and master level, in the area of computer science and economics as well; the last one was though in an interdisciplinary degree named 'Net economy'.

These courses had some common characteristics:

- small classes: from 9 to 20 students;
- availability of a computer laboratory, with no more than two students per personal computer;
- support of a tutor for a part of the laboratory activities and for the evaluation of the projects;
- support of an e-learning platform to publish class and projects documents (https://comunitaonline.unitn.it);
- presence of international students: an average of 2 to 4 students per year come from other European and non European countries;
- one term course of 30 to 40 hours.

Many of these characteristics positively contributed to the adoption of an activity based approach in which students have to work in group on a small scale real project. Other factors were in place only for some of the courses; in particular, after the last reform of the Italian university curricula, students of the master courses come not only from a computer sciences bachelors (first degree), but also from degrees in the area of business and economics. The main goal for all these courses was to focus on the problem-solving nature of requirements, and in particular on the need to first analyze a problem from the point of view of all the stakeholders. That in turn, implies a focus on requirements elicitation and analysis in about half of the course, assuming that it is easier to learn, apply and validate other UML models (requirement specification and validation) than to learn a structured way to effectively define the scope of a project and the requirements to be fulfilled. This approach requires soft skills that are very much useful in real projects: working in group, properly interacting with users and owners of the system to be developed, applying different techniques for fact-findings. In this context, the didactic objectives of the student projects were:

- introducing the role of the computer based systems to solve business problems or suggest business strategies;
- integrating organizational issues in the analysis of the problem to answer questions like, e.g. "who has to cooperate to gather the data for the system to be developed?";
- understanding the role of the requirements analysis in the process of the system development, including contractual impacts;
- identifying and managing conflicting requirements;
- using UML since the very first step of requirements modeling, also for business processes and actors; and
- documenting requirements as a project specification and their validation.

To this end, we defined a framework that integrates a model of the computer based system as a solution of 'business challenges' into a template for a business requirements document.

The first model (Fig. 2) was introduced by Laudon and Laudon [7] and is almost unknown to requirements engineers. However, its efficacy is validated by the high number of editions of their information systems books and of courses adopting them. It was chosen from existing socio-technical information models because it offers some interesting advantages (see section III B)). The L&L model defines and, more importantly, graphically represents the information system as "an organizational and managerial solution based on information technology, to a challenge posed by the environment" [7]. In this way, it can be used for any kind of project (size, domain, sector), forcing the adoption of a problem solving approach (starting from the business challenges, first box in Fig. 2), taking into account the management decisions and goals (actual or needed to address the identified challenges), identifying the main technologies for the project and addressing organizational issues relevant for the project (stakeholders to be involved). The business solutions component compels evaluation of the impact of the proposed system in terms of indexes or any kind of output useful to check if the expected goals or returns have been reached.

The second model has been adapted from a template introduced by Podeswa [8] as part of his approach, BOOM, or business object-oriented modeling. BOOM applies UML to business analysis underlining the role of business use-cases to define the scope of a project and its stakeholders, both internal and external to the company.

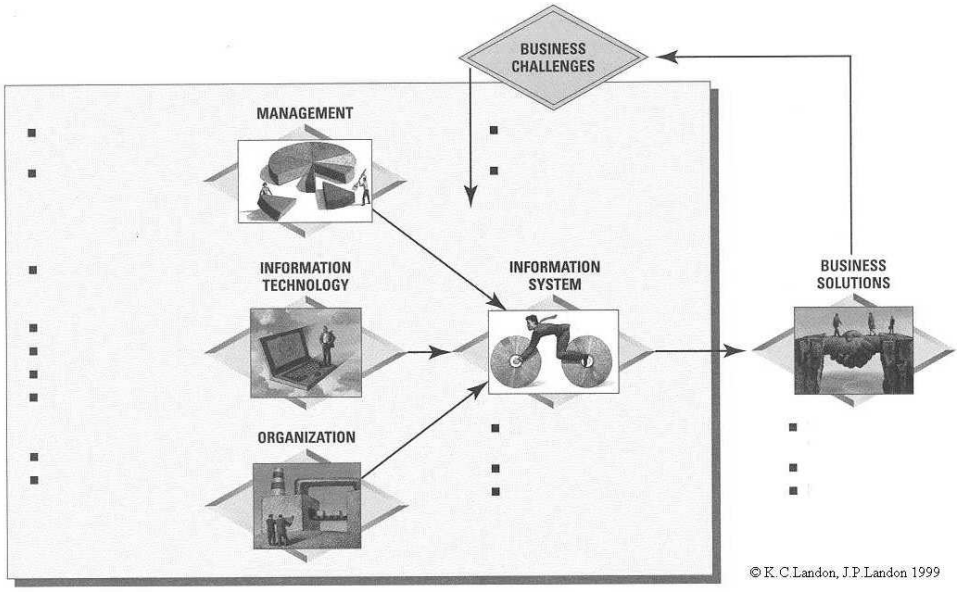

Figure 2. Information System model [6]

The BOOM template of the business requirements document (BRD) implements best practices and guidelines of the author's experience in his company.

For the student projects, an adapted version of the BRD was defined extracting issues relevant for the requirements engineering steps and integrating the L&L model. Table I gives the revised BDR used for the student projects of the 2009-2010 academic year; each group had to complete the requirements analysis for a given module of a museum information system (for other courses, the modules were for the information system of an international sport event, the Universiadi of Trentino; the Unisport initiative of the University of Trento; the e-commerce modules for profit and non-profit cultural organizations, and many more). A short problem statement and instructions for the project are given in the first part of the document (text highlighted in grey was in Italian).

To draw and document object-oriented models, the first projects were supported by Rational Rose, a tool that was available on a CDRom included in some of the widely adopted UML text-books. Later, given that many students owned a personal computer or a laptop, students were taught a variety of factors to take into account in choosing this kind of tools and then they could use the one they identified as the best solution. But the main didactical decision was to introduce the use of the tool only after the L&L model was completed and a preliminary set of requirements were elicited. That decision reduced the risks of premature requirements modeling due to students' tendency to start modeling, even if business analysis and requirements elicitation have just started. It also increased the quality of the requirements modeling and that of the BRDs.

B. Remarks and Guidelines

The L&L model was used for the student projects since the early '90s: it was the first model students had to complete and revise during the project. The model shows a number of characteristics that makes it an efficacious educational tool: it is quite intuitive; it scales well, as it can be applied to small projects and to large ones; it takes into account all the non-technological aspects relevant for any real project; it supports interactions with stakeholders, internal and external to the company and among the students working on the project; it can be applied in an incremental way, starting with knowledge and assumptions available in the very initial steps of the project, and adding to it and revising it whenever useful. To students with knowledge in economics and management, the model is a natural bridge towards requirements analysis; for those in computer sciences, it suggests a wider context for their projects. Another advantage of the L&L model is that it made it possible to work on the project since the first meeting of the course.

The BOOM BRD was first adopted in 2005-2006, but a number of improvements and changes have been made in the following years. In particular, the L&L model was added as a task of the 'Executive summary' (Table I); business use cases introduced by BOOM, helped to deal with stakeholders and with actors relevant for the project. Parts to be completed were reduced, making it mandatory for the students to complete only some of them (in bold in Table I). The results improved, possibly because the students felt the need to realize and document some of the activities as useful for the project and not (only) for the marks. In some cases, students completed also non mandatory tasks, as for example state diagrams to describe critical use-cases. The RACI (responsible accountable consulted informed) chart was useful to identify their roles in the project. Risk analysis, cost-benefit, and timetable supported the multi-disciplinary approach: computer sciences students understood that economics skills are useful for requirements engineering, suggesting to forming groups whenever possible with students of both the economics and IT areas. Also, the difference between the role of the business or IT analyst and that of a requirements analyst was better understood.

TABLE I. THE TEMPLATE FOR THE STUDENT PROJECTS OF THE 2009-2010 ACADEMIC YEAR ADAPTED FROM THE BRD PROPOSED IN [8]

Goal of the project (Problem statement)
Each group has to analyze the requirements for an information system module for the new museum MUSE (http://www.muse2012.eu/), now Museo Tridentino di Scienze Naturali (http://www.mtsn.tn.it/). The module has to be accessible from the website of the museum.
The project has to be completed using the BRD template of BOOM. Some parts of the template are mandatory or have to be adapted (those in bold).
Version Control (add: name of the module, name of the students)
Revision History
RACI Chart
Executive Summary (add the Laudon and Laudon model)
Overview
Background
Objectives
Requirements
Proposed Strategy
Next Steps
Scope (add the scope of the module vs the scope of the information system of the organization)
Included in Scope
Excluded from Scope
Constraints
Impact of Proposed Changes
Risk Analysis
Technological Risks
Skills Risks
Political Risks
Business Risks
Requirements Risks
Other
Business Case
Cost/Benefit Analysis, ROI, etc.
Timetable
Business Use Cases
Business Use-Case Diagrams
Business Use-Case Descriptions (text and/or activity diagram)
Actors
Workers
Business Actors
Other Systems
Role Map
User Requirements
System Use-Case Diagrams
System Use-Case Descriptions
State Machine Diagrams
Nonfunctional Requirements (highlight the most relevant)
Performance Requirements
Stress Requirements
Response-Time Requirements
Throughput Requirements
Usability Requirements
Security Requirements
Volume and Storage Requirements
Configuration Requirements
Compatibility Requirements
Reliability Requirements
Backup/Recovery Requirements
Training Requirements
Business Rules
State Requirements
Testing State
Disabled State
Static Model
Class Diagrams: Entity Classes
Entity Class Documentation (…)

In recent years, thanks to the project template, projects become more and more related to real organizations or companies. At first, there was only slight involvement of the company in the projects, by an initial presentation by a company representative and final presentation of results by students. Later-on, there were more and more stakeholder interviews with real stakeholders from the organization. For example, for the problem statement in Table I, student projects were defined for a natural sciences museum for which a new building and new presentations were in construction. An expert of the museum interacted with the teacher to simulate an interview to gather the initial requirements for a number of modules; other stakeholders were contacted by students working in groups of two. The final BRD was then presented and discussed with another manager of the museum in a meeting that also allowed highlighting the logical architecture of a system integrating the projects and the impacts of the proposed modules. Besides, the priority of the modules of the student projects was discussed with the manager, comparing them on the basis of costs and other resource constraints in the context of the museum strategy (for example, the online ticket selling module was given a lower priority than the online event management). The same approach – students completing the activities included in the template, initial interaction and final discussion with representatives of the companies or organizations – has been applied in other projects in which requirements for the same service – an e-commerce module – had to be analysed for different cultural organizations, as for example, a wine road (www.mtvtrentinoaltoadige.it), a Jazz festival (www.fiemmeskijazz.com), a contemporary art museum (MART, www.mart.trento.it), a film festival (www.trentofestival.it). In the last course, student projects were related to a new initiative of the University of Trento, Unisport (www.unisport.tn.it). The proposed approach turned out to be successful: for example, the results of such course projects were then used by the software development company ultimately in charge of projects implementation as input for the design and implementation of the Unisport information system.

C. Preliminary Validation

For a preliminary and external validation of the proposed framework we can use the most important recommendations for teaching requirements described in Ouhbi et al. [4]:

1) Teach how to define the scope of the problem and avoid general and vague specifications.
2) Show how to select and use a requirements engineering tool.
3) Promote activities in requirements analysis and modeling in addition to requirements management and introduce the concept of prototyping in the course.
4) Involve students in industrial projects (…). Instructors could also invite industry practitioners to present real projects and to describe their accumulated industrial experience.
5) Have the ability, skills, and strategies needed to align requirements engineering courses with contemporary global software development (GSD) conditions.

6) Familiarize students with approaches to problem solving, development methodologies, and development tools.
7) Use mobile devices as teaching tools. (…) m-learning (learning with mobile devices) promises a continued extension toward ‘‘anywhere, anytime’’ learning.
8) Deliver the courses in an interactive classroom: The students can share a virtual whiteboard, electronic textbook, and data over a networked environment.

Taking into account that the proposed framework is for the student projects and not for the entire course, all the recommendations were satisfied in our courses, except for number 8. In particular, the document template strongly support recommendation n°.1, 3 and 6.; the choice of the tool by the students in their project forced them to take into account many of the parameters characterizing the alternatives (recommendation n°.2); prototypes were used mainly to identify requirements for human-computer interfaces (n°.3); the quasi-real projects could be further exploited in longer courses and in any case, it must be reminded that it takes a lot of efforts to involve local organizations, experts and managers for the projects (recommendation n°.4); GSD conditions (n°.5) were only partially in place whenever interfaces were relevant for the project and emerged thanks to the international students (sometimes projects were completed by students staying in different countries). The seventh recommendation has not been experimented yet. Finally, the last one (n°.8) was partially fulfilled using the e-learning system (comunitaonline).

IV. Conclusion

The ability to manage requirements from a software engineering point of view as well as to grasp the business problems and strategies is critical for a requirements engineer. To address challenges in teaching requirements analysis we proposed a framework – a template and guidelines – for student projects that offers a number of interesting insights. Threats for the results are related to two main points. Firstly, reporting the author experience, mainly in an Italian university, all the following remarks embodies open questions and should be further investigates in a future work. Then, the small class sizes and the other characteristics described in Section III (apart from the last one) correspond to ideal conditions that are not always in place. The design (as lecturer) and realisation of non-trivial term-sizes student projects for real companies or organizations based on the proposed framework allowed exploiting and addressing, albeit to a different extent, the issues summarised in (Fig. 1). Having students with different skills working in group helped them to recognize their respective roles in the analysis of the problem and in turn, the relevance of other disciplines. Students were highly motivated to realize a good project thanks to the possibility to communicate with real business stakeholders and to validate their BRDs with them. The presentation and discussion of the requirements documents to company representatives is also a good introduction to business areas and gives an understanding of how to work with real customers. In some cases, the effect was prolonged in other courses of the graduation or master program (e.g., using requirements analysis for programming courses); some of the BDRs were adopted by the companies and used for in-company thesis projects. Other positive outcomes were: a higher number of students completing the project and the exam in the first exam session and better quality of the models [13]. When the course was not mandatory, the number of students decreased, but those in the course were more capable (higher marks and more skilled in IT) and motivated. From the students questionnaires this phenomenon (less, but more capable students) seems correlate with a higher difficulty to work on a project during the term.

Acknowledgment

The author would like to thank all the tutors who cooperated in many ways to the student projects. Many thanks also to Nadzeya Kiyavitskaya, Barbara Froner and Daniel Berry for their precious suggestions. Thanks to the anonymous reviewers for their thoughtful reflections and suggestions.

References

[1] N.R. Mead, and E.D. Hough, “Security requirements engineering for software systems: case studies in support of software engineering education,” in Proc. 19th Conf. on CSEE&T, IEEE CS, Washington, DC, USA, 2006, pp. 149-158.

[2] D N. Port, B.W. Boehm, and D. Klappholz, “Making requirements prioritization a priority,” in Proc. 21st CSEE&T IEEE CS, Washington, DC, USA, 2008, pp. 250-261.

[3] T. Nakatani, T. Tsumaki, and T. Tamai “Requirements engineering education for senior engineers: Course design and its evaluation,” 5th Int. Work. REET, 2010, pp. 26-35.

[4] S. Ouhbi, A. Idri, Fernández-Alemán, L. José, and A. Toval, “Requirements engineering education: a systematic mapping study,” Req. Eng., Springer, London, 2013, pp. 1-20.

[5] O. Hazzan, T. Lapidot, and N. Ragonis, Guide to teaching Computer Science, Springer, 2011.

[6] M. Fernandes, R. J. Machado, and S.B. Seidman. “A requirements engineering and management training course for software development professionals,” in Proc. 22nd CSEE&T, IEEE CS, Washington, DC, USA, 2009, pp. 20-25.

[7] K.C. Laudon, and J.P. Laudon, Management Information Systems. Organizational and technology in the networked enterprise, Prentice-Hall, 1999.

[8] H. Podeswa, UML for the IT business analyst. A practical guide to object-oriented requirements gathering Thomson, 2005.

[9] PM Solution, Strategies for project recovery, 2011, www.pmsolutions.com/resources/view/strategies-for-project-recovery

[10] J.M. Wing, “Computational thinking,” Comm. of the ACM, 2006, 49 (3): 33-35.

[11] L. Mamykina, L. Candy, and E. Edmonds, “Collaborative creativity”, Comm. of the ACM, 2002, 45(10): 96–99.

[12] P.J. Denning, and A. McGettrick, “Recentering computer science”, Comm. of the ACM, 2005, 48 (11): 15-19.

[13] N. Kiyavitskaya, N. Zeni, L. Mich, and J. Mylopoulos, “Experimenting with linguistic tools for conceptual modelling: quality of the models and critical features”, in Proc. 9th Int. Conf. NLDB 2004, Springer, 2004, pp. 135-146.