A Collaborative Approach to Software Engineering Education

Neli Maneva † and Krassimir Manev ‡
† Institute of Mathematics and Informatics, BAS, Acad. G. Bonchev str., bl. VIII, Sofia, Bulgaria
neman@math.bas.bg
‡ New Bulgarian University, 21 Montevideo str., Sofia, Bulgaria
kmanev@nbu.bg

ABSTRACT

The paper presents an approach to quality improvement of Software engineering education. The main idea is to state a goal, recognized as significant by the partners from three different areas – education, science and business, and then to try to unify their efforts so as to achieve the goal. The basic principles of the approach are presented, together with a systematic methodology and the formal method, built in it. Some real-life examples of method's application are briefly described and commented. At last, the results obtained are discussed and some ideas for further development of the approach are mentioned.

KEYWORDS

Software engineering (SE), collaborative approach, SE education, quality improvement, Comparative analysis.

1 INTRODUCTION

The increasing role of software-intensive systems in our life and work is the reason for the growing concern of society to the education and qualification of people, involved in software development. That is why the quality of Software Engineering (SE) university education and Life Long Learning (LLL) is a hot topic of many specialized and public discussions. Unfortunately, in such discussions the participants from the three concerned domains – education, science and business, pull in different directions. Although they share the similar goals, there are quite different opinions about the strategy and tactics for goals achievement. The reactions of people, involved in the SE education process, can be explained by the so called "causal attribution". According to this theory from social psychology [3] when people try to explain the behavior of others, usually ascribe to themselves the credit for success, but for failures tend to indicate only external causes, and other culprits.

It is clear that in any discussion on SE education the prevailing complaints, negative findings and making accusations to others will not lead to improvement. So we decided to look for a feasible and a more efficient approach.

Exploring some good European and national practices and integrating the efforts of carefully selected and highly motivated participants, we tried to develop and implement a constructive approach to SE education. Part 2 of this article presents our key ideas and principles to be followed. Part 3 reveals some factors, influencing the implementation of the proposed approach, based on the INSPIRE methodology. The Comparative analysis method is described and how it can be applied for a number of tasks, improving the quality of SE teaching process, is shown. In Conclusions the results of the established mutually beneficial partnership for the educational purposes have been summarized and some open problems that need further study have been shared.

2 OUR APPROACH

Two years ago, when we started to work on a project, entitled "Automated extraction of business rules and business processes from program code", a consortium was created. It involved some lecturers from the Faculty of Mathematics and Informatics, Sofia University, a number of scientists from the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences and a few developers from a software company. Early in the project an idea appeared: to use the potential of the already established tripartite partnership in favor of SE teaching activities, too.

A small team (a specialized working group) of representatives of the above mentioned three institutions has been created. All team members have been enthusiastic and highly motivated to participate (in accordance with their competences) in the SE education and training process so as to try to improve its quality.

The idea to unify the efforts for an important national cause is not new. Our study showed that in the country there were many examples of a successful bilateral cooperation so as to achieve the purposes, recognized as national priority. For example, such cooperation was officially established between some scientific and educational institutions in the 70s of the last century. Unfortunately, after 10 years of existence those United Centers for Science and Education have been disbanded as inefficient, mostly due to bad management and wrong goals setting. Currently, our Institute of Mathematics and Informatics participates in several joint projects with various software companies, and although no special regulations and signed contracts, the results till now are encouraging. Perhaps there are some other successful bilateral partnerships between research institutions, software companies and universities, but we could not find any officially published summary data on organizational, financial and conceptual principles and results of such partnerships.

On the base of the performed field study and experience gained in a number of previous joint projects, we try to develop a collaborative approach to SE education and training. After a broad discussion, a few principles to be followed have been approved:

- To establish a mutually beneficial cooperation with clearly stated regulations for tasks and responsibilities of participants;
- To create awareness and appreciation of some goals, recognized as mutually beneficial and to assure voluntary participation with specific for each partner competencies;
- To develop an environment, in which the collaborative activities will be supported by a systematic, scientifically-based and validated methodology.

3 FROM THEORY TO PRACTICE

3.1. Implementation of the approach

After some discussions within the specialized working group, an agreement has been achieved, that SE education, LLL and training should be planned and implemented taking into account a number of factors:

- The widespread use of computers in modern society sets some new, higher requirements to the qualification and skills of people involved in software development. Basic education in universities should be a solid foundation for the unavoidable Long Life Learning [2], accompanying the whole professional life of software people;
- The inherent interdisciplinarity of SE has been further developed and nowadays the software teams involve many experts with different backgrounds, skills and professional experience. The current software projects are implemented with the active participation of both application domain experts and scientists. In many cases the involvement of researchers is crucial, especially for solving some difficult problems with challenging algorithmic, methodological or technical complexity. Therefore, university students need to develop the right attitude and understanding of the power of scientific knowledge and usefulness of its transfer into practice;
- The modern software business is trying to be more intensive and global. For its survival in the current financial crisis it begins to adopt some new managerial and technological approaches. When business people are involved in SE education and training process, they might share some identified best practices, present a few characteristics of a number of contemporary application areas, and to elucidate some requirements to the expected knowledge and skills of the graduated students. So the shared knowledge and experience can be used to define and/or to upgrade the content of specific courses, workshops and diploma theses, motivating additionally the Computer science students to

participate in some internships and real-life collaborative projects.

In accordance with the above mentioned principles, we defined the main tasks and responsibilities of each member of the working group. The lecturer of the basic Software Engineering courses presented some problems and suggested a number of concrete actions to be accomplished so as to improve the teaching process. The participants from the research institute have been involved in the selection of research paper topics, concerning a few innovative software techniques. A number of representatives of our business partners have delivered lectures on "hot" current trends in software industry. Some of them have been coaches of students, working in teams on practical problems. After a tough selection procedure, some software engineers have acquired a specific tutor's qualification so as to define the topics, scope and other project-related parameters of intermediate and diploma works.

As a systematic methodology to be followed, we chose the methodology **INSPIRE,** presented in [5]**.** It integrates a formal method for comparative analysis with some already validated best practices. The detailed description of the methodology is beyond the scope of this paper. Here we will present only the key ideas of the methodology. Its main features are encoded in the name, which stands for **I**ncremental, **N**eat, **S**calable, **P**ermanent, **I**ntegrated, **R**ight and **E**stimable. The methodology has been successfully applied for several different in scope and complexity SE activities – SE objects modeling, improving software quality, ensuring usability, and training in Informatics for beginners. As it is validated both theoretically and experimentally, we believe that this methodology will be appropriate for SE teaching, too.

The detailed description of the methodology characteristics can be found in [5]. We will mention only the specific meaning of its characteristics for the intended use – SE education and training. The essence of the *Incremental* feature is that at the beginning of the experiment we pick just ***a few*** activities that should be object of study, creating and accomplishing an action plan only for them. After evaluation of the obtained results, we might add some new activities to be improved. In this experiment we chose two starting activities: *to update* the content of two basic SE courses (Introduction to SE and Advanced SE) and *to improve* the organization of workshops and diploma work assignment.

The methodology requirement for simplicity (to be *Neat*) here is satisfied by the selection of a feasible objective that can be achieved through some preliminary planned actions, which are with adjustable scope and gradually deteriorated (*Scalable*), constant over time (*Permanent*).

The performed actions have been selected so as to complement each other (*Integrated*). The whole approach is supposed to be implemented by qualified professionals (*Right*) and its progress can be tracked by comparison of appropriately chosen and measurable indicators, so as to meet the requirement for objective assessment (*Estimable*).

The methodology is designed as a flexible and goal-oriented and its use is related to determining the specific tasks, so as to achieve the stated goal. Following the recommended in [5] steps of the procedure for methodology application, our experiment started with setting a goal: ***improving the quality of SE education and training for undergraduate students*** from the Faculty of Mathematics and Informatics, Sofia University. This should be done by involving software professionals from the three institutions. The chosen participants have been with different background, qualification, knowledge and experience in the field of SE. Setting a common goal and taking into account the different points of view and variety of ideas how to achieve the goal, we intend to construct and follow a feasible action plan, comprising the tasks, main actors and deadlines for their implementation. The detailed description of the specific tasks is beyond the scope of this article. We want to mention only the tasks, related to the use of a formal method, supporting the decision making in SE education and training process, namely the Comparative analysis.

3.2. Comparative analysis in SE education

Comparative Analysis (CA) is a study of the quality content of a set of homogeneous objects and their mutual comparison so as to select the best, to rank them (establishing a preference order) or to classify each object to one of the predefined quality categories.

The CA shares the main objectives and techniques of the multiple criteria decision making theory, trying to specify and apply them systematically.

For CA application we distinguish two main roles: the ***Analyst***, responsible for all aspects of CA implementation, and a ***CA customer*** – a single person or a group, tasked with making a decision in a given situation. Depending on the identified problem to be solved by a customer at a given moment, a ***case*** should be opened to determine the context of the desired comparative analysis. Each case is specified by the following 6 elements:

case = { View, Goal, Object, Competitors, Task, Level }

The ***View*** describes the CA customer's role and the perspective from which the CA will be performed.

The **Goal** expresses the main customer's intentions in CA accomplishment and can be to describe, analyze, estimate, improve, predict or any other, formulated by the Customer, defining the situation.

The **Object** represents the item under consideration. According to the Fenton's classification, in the field of Software engineering, any studied object belongs to one of the following groups: products, processes or resources. For each object for CA application a quality model should be created – a set of characteristics, selected to represent the quality content in the context, defined by the case, and the relationships among them.

According to the goal, the set C of **Competitors, $C = \{C_1, C_2, \ldots, C_n\}$** – the instances of the objects to be compared – should be chosen.

The element **Task** of a case can be *Selection* (finding the best), *Ranking* (producing an ordered list), *Classification* (splitting the competitors to a few preliminary defined quality groups) or any combination of them.

The element **Level** defines the overall complexity of the CA and depends on the importance of the problem under consideration and on the resources needed for CA implementation.

The most difficult (from cognitive point of view) part of the CA application is the construction of object quality model, relevant to the defined case. Usually this model is presented as a hierarchy. At the top of the hierarchy is the total object quality. The first level comprises some user-oriented attributes, called ***factors***. The next level describes a number of object-dependent attributes, providing quality. These ***criteria*** can be further decomposed to some more simple and measurable characteristics. To each node a weight (a coefficient of importance) is assigned, and for the leaves of the hierarchy some appropriate ***metrics*** are defined. Starting a bottom-up evaluation of characteristics at each level in the hierarchy and applying a modification of the MECCA (Multi-Element Component Comparison and Analysis) method, we can obtain the quantitative measures of all factors and fill the competitors-factors matrix **E(nxm)**, where **n** is the number of the competitors, defined in the set **C**, and **m** is the number of the quality factors. Each element **Ei,j** is the measure of the **i**-th competitor with respect to the **j**-th quality factor. The obtained matrix **E** is further used as input to the software tools, implementing the required selection, ranking or classification methods.

In our experiments the CA has been used for the next tasks:

- Comparing the themes of the basic SE course (Introduction to SE) so as to determine the content, scope and level of details of their presentation, taking into account different opinions about their pedagogical usefulness;
- Selection of some topics that can expand the thematic scope of the Advanced SE course in its part "Contemporary trends in SE";

- Comparing the forms of midterm student's assessment so as to select the two best of them, suited to the stated didactic goals to be achieved;
- Ranking the proposed from business representatives assignments for educational projects to be carried out by teams within the SE workshop;
- Creating a sorted list of team members, working on a given software project. The ranking should be based on three sets of characteristics – professional knowledge, specific skills (soft skills) and practical experience. So for a project manager will be selected the student, received the highest comprehensive assessment. The task assignment can be made taking into account the corresponding estimates of individual characteristics for each team member.

In order to apply the CA for the above mentioned tasks, we have to construct the corresponding quality models for the studied objects: SE course topic, form of assessment, student. For the objects "topic" and "form of assessment" we chose a simple linear model with selected quality characteristics, measured on a scale from 1 to 10 (the highest). For the task of selecting the team leader, we use a hierarchical object quality model, constructed for the object "student".

The context-oriented approach has been accomplished by variations of a standard case, for which the values of the basic elements are as follows:

<*goal*> – evaluation;

<*object*> – depending on the task, it can be lecture topic, form of continuous assessment, pilot software project, student;

<*task* > – ranking;

< *level* > – simple.

The other two elements, necessary to determine the CA context by a case, have been with varying values. For the element **<*view*>** we performed the CA with four different values, reflecting the point of view of the lecturer, a representative of software business, a researcher in the SE field and a student, currently/previously attending the SE course.

For the element of the case **<*competitors*>** – the set of compared objects, some instances of the studied object have been chosen. They have been identified as appropriate for comparison after some discussions within the working group.

The assessment procedures for each case use different coefficients of importance (weights) for the selected quality characteristics. In this way we can adjust the performed analysis to the context, determined by the defined situation.

3.3 Results from the experiment

We shall briefly present the results obtained after the CA use for the following four tasks, described below.

A. Comparing topics from the basic part of the SE course and from the section Contemporary trends in SE

In this case for quality assessment of each topic we chose three characteristics with equal weights: ***utility***, ***applicability*** and ***validity***. They have been assessed by questionnaires from four different perspectives – that of a teacher, a researcher in the SE field and a software engineer – a former student, trained in this program. Based on the estimates, for each factor and for the themes as a whole, the CA method has been applied. The obtained two sorted lists have been further analyzed in order to update the course content.

For example, within the first set of themes (in the basic part of the course), after the CA use, the ranked second to last topic "Software Metrics" will be a subject for further improvement. The topic, ranked as last – "Boehm's COCOMO Model" – will be entirely re-designed or even canceled.

Taking into account the obtained results, after some discussions within the working group and looking for compliance with the SE content, given in [7], a list of some additional themes, covering the topics of contemporary SE, have been created. The CA method has been used for selection of three themes, enriching the thematic scope of the SE course in the part of SE modern trends. As a result of the performed CA, the three themes with

the highest rating have been found, namely: "Innovations in software business", "Legal aspects of the software development and distribution" and "Agile methodologies in SE". Now the development of lectures on these topics and their inclusion in the SE course is in progress.

B. Comparing some forms of midterm student's assessment in order to choose the most appropriate ones for achieving the stated didactic goals.

For the object "form of midterm student's assessment" a quality model with two characteristics with equal weights have been constructed. The first characteristic is ***effectiveness***, reflecting whether the requested results are obtained in accordance with the objectives. The second characteristic is ***efficiency***, reflecting whether the assessment form fulfills the stated goals with minimal resources such as lecturer's time and efforts for exam materials preparation and evaluation.

In this case the comparison has been carried out from two perspectives – the lecturer's and that of the former student with a Bachelor's degree, who has passed such assessments. Two experiments have been conducted to compare different sets of assessment forms, according to the number of students (25 or 230), enrolled for the SE course and connected with its workshop.

For the stream of 25 students, the recommended assessment forms for the lectured course have been *a written exam* on two topics from the syllabus and a short research (referral) paper on a topic close to the covered by the lectures. For the workshop the selected form, recognized as the best, was a small software project assigned to a team.

For the stream of 230 students, the recommended assessment form for the lectured course has been a test (with 2 multiple-choice questions and another 8 questions, required open answers). Such a test will be made twice - in the middle and at the end of the semester. In the syllabus has been stated that in case the average test score of a student is at least 4.5 (in the 2-6 scale), the result can be considered as final grade, i.e. the students, achieving such result, no need to sit for the final exam. The experiments with use of this assessment form have shown, that this regulation additionally motivates students to go through the material and learn it in order to decrease their burden during the final exam period.

C. Comparing the assignments for projects within a SE workshop

The assignments have been defined by some experienced project leaders from the software company, participating in our experiment. The proposals have been described in a standard form and in the most cases they have been parts of company's ongoing projects, properly redefined and shaped in accordance with some typical project resource constraints. The team work on a project involves design, prototype development and writing a standard set of documents – both project-oriented and user-oriented.

The quality model of the object "project assignment" comprises three characteristics, considered with equal significance (weights): ***relevance, technological complexity*** and ***usefulness***. They have been evaluated on the base of a questionnaire from four different perspectives – lecturer's, SE scientist's and that of a software engineer – a former student, participating in the same workshop during her study within the same Bachelor's program.

Among the proposed four project assignments, after the CA ranking, the following two assignments have been selected for further implementation: "Internet-based Organizer " and "Mobile Application for Social Networking".

D. Election of the team leader for a project

For the CA in this task, a quality model for the object "member of a team for project" should be created. On the base of our previous works on Informatics student modeling and some identified software personnel soft skills ([6]), a hierarchical quality model for the object under consideration has been created. It comprises three factors (user-oriented quality characteristics): psychological traits, basic knowledge and skills, and practical

experience. These factors have been further decomposed into a number of criteria (object-dependent characteristics).

The factor "psychological traits" has been decomposed to:

K11 – cognitive adaptability;
K12 – constructive thinking;
K13 – discipline and ability for self-organization;
K14 – stable psychological profile;
K15 – ability to focus quickly on a problem.

The factor "basic knowledge and skills" has been decomposed to:

K21 – level of basic IT-skills;
K22 – achievements in mathematics (based on the academic record with grades in mathematical subjects;
K23 – English language knowledge and skills (understanding, reading and speaking).

The factor "IT soft skills and experience" has been decomposed to:

K31 – communication skills;
K32 – be able to work in a team;
K33 – conflict resolving and negotiating skills;
K34 – practical experience in a software company.

The quality characteristics, which are leaves at the constructed tree structure, have been assessed through questionnaires from four different perspectives: instructor's, SE researcher's and that of a working as project manager former student, involved in such workshop during her study in the same Bachelor program. According to the evaluation procedure, described in [5], the person, performing the evaluation, should define some weights for the factors and criteria, which are not leaves.

For this task, the CA has been applied four times so as to produce an ordered list of all team members. After that the CA has been used again in a case with the elements:

<***view***> – that of the SE teacher, responsible for the workshop;
<***goal***> – evaluation;
<***object***> – a student;
<***competitors***> – four students, who have been at the top of the ordered list, produced during the previous ranking;
<***task***> – selection, i.e. finding the best;
<***level***> – simple.

So, the project leader has been elected – the student, identified as the most appropriate among all team members through an objective procedure, taking into account the opinions of all four evaluators.

4 THE COLLABORATION – SOME CHALLENGES AND RESULTS

The tripartite partnership, established for the purposes of improving SE education quality and training, has been active during two academic years. Now we can analyze the obtained results so as to decide whether and how to continue.

In order to accomplish a systematic approach, at the beginning of the experiment we tried to clarify:

- What are the intentions and the potential benefits for each partner and how they can be achieved?
- What are the possible risks and how they can be avoided in a systematic way?

Considering the motivation of participants as a crucial success factor, we started with creation of positive attitude to the experiment. Through discussions, exchange of opinions and professional know-how, we reached the strong belief in the power of such mutually beneficial cooperation.

Very soon after starting the experiment, three organizational problems have been anticipated:

- Creating a convenient (for all participants) schedule for the planned face-to-face meetings;
- Establishing and maintaining a few effective ways for communications;
- Dealing with some psychological problems (mainly in communication), because in groups of people with bright leadership skills and high self-esteem for intellectual and professional level, usually it is difficult to

work efficiently and to reach consensus in discussions.

To tackle the first two problems, we decided to minimize the number of meetings that require personal presence with prevailing communication via Internet (through emails and Skype). We agree to stick to the rule that the acceptable delay of answers to emails can be up to 2 days.

For the third problem, we follow some recommendations for efficient teamwork: to select on rotation a main contractor for each task and a moderator for each discussion – the participant possessing the required competences in the highest degree.

At the end of the experiment we can summarize the benefits for the involved groups of participants as follows:

For students:

- The content of the SE courses has been updated and (we deeply hope!) improved;
- Diversification of lecturing style has been achieved due to possibility some successful managers from well-known software companies to act as invited speakers and discussion moderators;
- The forms of continuous student assessment have been selected, taking into account the opinion of lecturers, students and their future employers;
- The organization and practical benefits from the SE workshops have been improved by involving some SE practitioners in assigning and coaching projects in workshops, defining topics for diploma works, etc.

The benefits for SE lecturers are:

- Improvement of the SE courses based on the content evaluation from two other points of view – that of researchers and software business people;
- Increasing student's interest and motivation to attend classes by inviting some guest speakers and consultants;
- Reducing the workload and enhancing the quality of course-works and diploma theses awarded and/or managed by selected representatives of the scientific or business partners.

The benefits for researchers are:

- The possibility to verify the compliance of research directions with the real needs of the practice in software industry;
- Evaluation of organizational and technological feasibility of some research and/or development methods. For example, the identified by investigation possible approaches for automated extraction of business rules from programs, have been discussed and experimentally examined by developers in the software company, so as to choose the most appropriate approach to be further prototyped and validated;
- Learning the effect of the application of some good managerial and business-oriented practices. For example, the best practice "Invest a little, learn a lot", mentioned in [1], has been studied and successfully adopted in a scientific project. It was recognized as useful, because it requires caution in financing projects in developing knowledge and effective use, when we make decisions about keeping project into a profitable direction.

The benefits for business-partners are:

- Learning and deeper understanding of some real-life problems in the other two areas - science and education. A side effect of this was the mitigation the aggressive tone in public debate;
- Increase of empathy and search of solutions through dialogues and discussions;
- Acquisition of theoretical knowledge about some new methods in the field of co-developed SE projects;
- Increasing the willingness to cooperate and work together so as to achieve a clearly defined goal of common interests, e.g. preparation of better software engineers;
- As a result of personal contacts established during the joint work (course-works, diploma works, joint projects within the workshops), the most successful students have been invited

to apply for free job positions in the respective software companies.

5 CONCLUSIONS

The experimental study began with some reasonable doubts about its feasibility. The main question was: Could we apply a systematic approach to SE education, demonstrating the benefits of the combined efforts to achieve a significant goal, or would confirm the incapacity of the created working group of people with quite different current needs and interests.

Now, analyzing the results, we can say that the experiment has been successful and shows that such trilateral partnership can be fruitful. Among the success factors it is worth to mention the following:

- Setting a common goal, recognized as significant for each of the partners;
- Creation of a jelled team, comprising appropriately selected, highly motivated and open-minded people, doing their best in this collaboration.

As a difficult and not yet solved problem, we can point the existing differences in defining the partners' tactical and strategic goals and the period of time, considered for their achievements. We can illustrate this with the so called Buxton index. The Buxton Index of an entity (person or organization), is defined as the length of the period, measured in years, over which the entity makes its plans. For some more conservative areas as science and education, this index is large (5-10 years), while for the business it is only 1-2 years. A well-known fact is that an effective collaboration can occur only between partners with almost the same values of this index. So, in such endeavor we should look for a combination of long-term goals of science and education with the short-term business-oriented goals, through appropriate formulation of some specific tasks of the joint work.

The experiments with the CA use for improving SE education can be extended with the support of SE lecturers in other (both economics and technical) universities. Some new cases with a higher level of complexity of implementation can be defined and implemented. For example, we can use the built in the CA classification method for re-formulating the groups of SE best practices, mentioned in [4].

In accordance with the principles of the INSPIRE methodology, our approach can be applied for trilateral cooperation in other activities, recognized as significant for SE, e.g. transfer of some theoretically validated scientific methods to practice, innovative pilot projects, specialized training of software developers in a company, etc.

6 ACKNOWLEDGEMENTS

This work is supported by the National Scientific Research Fund under the Contract ДТК 02-69/2009.

7 REFERENCES

[1] Anthony, Sc., M. Eyring, L. Gibson, Mapping Your Innovation Strategy, In Harvard Business Review on Business Model Innovation, Harvard Business School Publishing Corporation, 2010.

[2] EQF - The European Qualifications Framework for Life Long Learning, Brussels, 2009, http://ec.europa.eu/education/pub/pdf/general/eqf/broch_en.pdf, visited March 2014

[3] Harmon-Jones, E. & Mills, J. (Eds.), Cognitive Dissonance: Progress on a Pivotal Theory in Social Psychology, American Psychological Association, Washington, DC, 1999.

[4] Jones C, Software Engineering Best Practices, Mc Grow Hill, 2010.

[5] Maneva N., INSPIRE: A Software Engineering Methodology, Journal of Information Technologies and Control, № 1, 2005, pp.31-37.

[6] Maneva N., N. Nikolova, Soft Skills Training for Software People, Proc. of the 7-th Int. conference on Computer Science and Education, Dobrinishte, July 6-10, 2011, pp. 117-129.

[7] Pressman R., Software engineering - A Practitioner's Approach, 7-th edition, McGraw Hill, 2009.