e-Informatica Software Engineering Journal, Volume 6, Issue 1, 2012, pages: 47–59, DOI 10.5277/e-Inf120104

A View on a Successful International Educational Project in Software Engineering

Zoran Budimac*, Zoran Putnik*, Mirjana Ivanović*, Klaus Bothe**
*Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad
**Institute of Informatics, Humboldt University Berlin
zjb@dmi.uns.ac.rs, putnik@dmi.uns.ac.rs, mira@dmi.uns.ac.rs,
bothe@informatik.hu-berlin.de

Abstract

In this paper, a successful and fruitful joint project will be presented. The project joins participants from 9 countries and from 15 universities. Since it started in 2001, this project entitled "Software Engineering: Computer Science Education and Research Cooperation" helped participants to gain excellent, up to date educational material, apply modern teaching methods, exchange experiences with other participants, and work jointly on the further development of lectures, case-studies, assignments, examination questions, and other necessary elements of a course. Project works under auspices of Stability Pact of South-Eastern Europe, and is supported by DAAD. The project started with the creation of a common beginning course in "Software Engineering", but over time it grew and the number of other courses was developed. Finished almost completely are the courses in "Object-oriented programming", "Software Project Management", "Advanced Compiler Construction", and "Data Structures and Algorithms", and some other courses are under development. Aside from the educational collaboration, project members also developed good scientific cooperation, and published several research papers.

1. Introduction

Since its beginning in 2001, an international project entitled "Software Engineering: Computer Science Education and Research Cooperation" assembles participants from nine countries, and from fifteen universities. Project exists under the sponsorship of "Stability Pact of South-Eastern Europe", and is financially supported by DAAD ("Deutscher Akademischer Austausch Diens", or "German Academic Exchange Service"). At the beginning, the project was concerned with the creation of a common course in the field of "Software Engineering", yet as it progressed, project dealt with the development of a number of other courses. Mostly finished so far are the courses in "Object-oriented programming", "Software Project Management", "Advanced Compiler Construction", and "Data Structures and Algorithms", while some other courses are still under the development. The most developed one is still the course in "Software Engineering". Aside from presentations of a theoretical material, a whole set of learning resources has been developed: e-Lessons, case-studies, team and individual assignments, or pool of questions, for example. Naturally, beside the educational purpose, members of a project later developed nice cooperation within the area of scientific research, and published several papers.

At first, cooperation within the project has been started by the group of researchers and educators that still make the core group of a project, and consists of members from: Germany, Serbia, FYR Macedonia, and Bulgaria. Head and the main coordinator of a project is Professor Klaus Bothe from the Humboldt University in Berlin. Over the years, project was enlarged

through the inclusion of participants from other Balkan countries: Croatia, Romania, Bosnia and Herzegovina, Albania, and Montenegro. While originally project was granted for the period of three years, excellent results in cooperation and development of joint teaching resources induced project continuation and new grants year after year, so the project still lasts. DAAD foundation also reported about the successful results of a project in [1] and [2]. The basic information about the project, its participants, and achievements can be found on its home-page [3].

There are several other projects of a similar type and purpose, let us mention [4–8] MuSoft, ISEUC, Swenet, Ariadne, Merlot, where the first three are also dealing with the field of "Software Engineering". Still, we feel that there is a substantial difference between those and our project, mainly in the approach to the creation of teaching material. All of mentioned projects created a set of relatively independent modules that can be combined and used as lecturers decide. In the case of our project, the idea was to create a complete course and the whole teaching material, creating a unity consisting of a sequence of interconnected material, yet allowing the substantial level of parameterization. The scheme behind this concept is to make the whole material usable even to those lecturers for whom "Software Engineering" is not in the key focus of interest.

The main official aim of the project was "academic reconstruction of a South-Eastern Europe". Still, it had a whole list of basic and more down-to-earth aims, of which we list here the most important:

- Inclusion of the course "Software Engineering" into curricula of participating universities;
- Creation of a consensus about the common course in "Software Engineering", selection of topics it will cover, and creation of jointly created pool of presentations from which the participants can choose the most appropriate ones for their university;
- Creation and development of joint teaching and examination materials for selected topics: presentations, case-studies, team and individual assignments, pool of examination questions, adequate literature, lecture notes, etc;
- Forming of bases for the further scientific and educational cooperation in the field, so that the actuality and quality of the teaching material is preserved.

All of the participating countries, more or less, took their part in the development of certain topics or subtopics. Some of the particular activities were:

- Further development of the existing teaching resources;
- Usage of the course as a whole, or some of its parts within the appropriate courses;
- Creation of reports based on the experiences and surveys performed;
- Creation of new topics, case-studies, assignments, and so on;
- Liability analysis, suggestions for the further development paths, creation of re-sources for additional common courses.

In the beginning, the course of "Software Engineering" was based on teaching mate-rials used at the Humboldt University in Berlin, which are in turn based on the text-book on software engineering [9]. All of the most important suggestions from the significant world computer science associations were taken into account [10,11]. This way, all of the basic and introductory topics were created, but also a lot of advanced topics suggested by the ACM, IEEE, and other world expert bodies. Recommendation was also made, that the course should be conducted on final years of studies, after they cover all of the necessary basic notions indispensable for the field. So far, all of the participating universities, followed the recommendation, and the course in "Software Engineering" has been conducted for the students of the final year of studies everywhere.

The rest of the paper is organized as follows: Section 2 presents the contents of the "Software Engineering" course, with all of its basic components. In Section 3, some problems that project participants encounter in their work for the project are presented. Section 4 presents the other courses developed through joint work within a project, based on the positive experi-

ence with the first one. Section 5 brings some students' reactions and opinions. Finally, in the 6th Section, more general conclusions are given, and further development paths are considered for the project members.

2. The Development of the Project

Based on the first and the most developed course created within the project (course on "Software Engineering") we will present the current practice of course development and refinement. Over the years, this course went through three, very often overlapping phases:

- During the first phase, existing topics based on [9] were translated from German to English language, and then, through the participation of all project members, refined, polished, and further developed. The bases for the refinement were the experiences with the course presentation at the home university;
- In the second phase, new topics were created and developed, basic and advanced. Those new topics are also continuously refined and polished over the years. General rule is that those new topics are at first developed in English, as a universal, common language for all of the participants. After that, through participation of all interested members, presentations and materials are refined and improved. Only in the final phase, after all of the members are satisfied with the quality of the material, resources are translated back to local languages, if needed;
- During the third phase, the final forms of the teaching materials, agreed by all of the project participants, were translated to local languages [12]. For those purposes, a specialized tool has been created [13].

As the basic outcomes for the project, it has been defined that the teaching materials should help students to develop the following abilities:

1. To work in a team;
2. To have analytic and synthetic approach to decision making;
3. To apply gained knowledge on practical assignments in realistic surroundings;
4. To renew, expand, and continually improve their knowledge, and
5. To make the appropriate decisions during the software development cycle [14]

All of the abilities mentioned here are rather generic, not related only to software engineering, yet that doesn't diminish nor weakens their importance. And, as "basic outcomes", we can say that after 10 years experience, those are quite fulfilled, if we believe the reports we get from the industry.

2.1. Teaching Materials for the "Software Engineering" Topics

One of the basic components of the "Software Engineering" course are teaching materials organized in five parts, with altogether 28 topics covered. Each topic is presented primarily as PowerPoint presentation, enriched with the multiply useful "lecture notes" for the lecturer. Those serve as a starting point for the exchange of ideas between the users of teaching resources (lecturers and students); they contain the answers to the questions presented during the lecture, and they enable creation of printouts of the materials presented during the lectures. Topics are divided into parts and organized as presented in Table 1.

From this pool of topics for which teaching materials were developed, each of the lecturers is allowed to select those suitable for his view on the course, or suitable for the curriculum as it is defined at his/her university. A natural consequence of this agreement is a variety of methods for the usage of teaching resources within the "Software Engineering" course. For example:

- Humboldt University of Berlin, Germany and University "Paisi Hilendarski" from Plovdiv, Bulgaria, use all of 28 topics in their course. The course is conducted on the fourth, final year of the bachelor studies;
- University of Novi Sad, Serbia and University "St. Ciril and Methodius", Skopje, FYR Macedonia, are using almost all of the topics, except several from the last part, "Advanced problems". Some of those topics that are not used are covered within some other master courses, while for some of the others there

is simply not enough time. Also, it is worth mentioning that the course is lectured on the last year of studies for couple of different directions, so there are students from the third, and from the fourth year of studies;
– Universities from Belgrade and Kragujevac, Serbia; Zagreb and Rijeka, Croatia; Podgorica, Montenegro; Sarajevo and Banja Luka, Bosnia and Herzegovina; Tirana, Albania, and Timisoara, Romania selected a subset of topics. Depending on the university, the number of topics varies between 5 and 12, and those are incorporated successfully into already existing courses on "Software Engineering", becoming the integral part of those;
– A special case is Polytechnic University of Tirana, where the course is not conducted during the regular school-year, but instead as a one-week crash-course, when about 18 topics are presented. Lecturers are visiting professor from Berlin, Germany, and assistant from Novi Sad, Serbia, and the course is conducted as a part of master studies. After the four crash-courses, part of the topics is taken over by local assistants from Tirana, while the general plan is that the whole course will be once conducted by local lecturers.

With this variety of types of course conduction, it is quite likely that topics are continuously being developed and refined. Large number of lecturers, each one with his/her own teaching style, habits, and pedagogical principles, guarantees the actuality and quality of teaching resources. All of the new ideas, techniques, suggestions, and innovations are exchanged during the regular meetings of the project members, conducted each autumn at some of the participating countries.

One of the methodologies presented, certainly deserves greater attention. Since at the moment, the most of the development methodologies are built around UML, this methodology is represented within the course also. Methodology is not introduced formally, since it has been studied within other, previously taught courses in that manner. Still, because of this fact, the lecturer is in a position to introduce the methodology through examples.

Within the introductory topics, the importance of UML is explained, and so are the notations, being the part of it. The "body of knowledge" for it is not described deeper, because it was the part of several compulsory courses preceding the "Software Engineering" course. Later on, the methodology is used wherever it is needed, as convenient to the lecturer.

To confirm that the students covered this important methodology sufficiently, one of the obligatory assignments students have also requires knowledge and usage of it. We will not discuss it in more details here, since more about this will be given in a subsection dealing with the assignments.

2.2. Case-studies for the course of "Software Engineering"

Second important course component, linked with both theoretical and practical exercises, are relatively complex case-studies. The main reason for usage of those case-studies is the need to illustrate theoretical concepts presented during lectures on some practical and realistic examples. The original course, used as a basic element for the development of the final project resulting course, used throughout the lectures two case-studies:
– "Seminar organization" – a software system, taken and adapted from [9], used to help running the company that deals with the organization of various educational seminars and their presentations to interested clients. The system is also supposed to help with: contact with clients and other companies, communication with the lecturers, students, hotels, travel agencies, and all other necessary users and services. This case-study is used within ten topics to illustrate theory presented during the lectures.
– "XCTL" – a real life software system, used to control the work of measuring instruments at the Institute of Physics, Humboldt University in Berlin, Germany [15]. System was analyzed, measured and enhanced using methods of re-engineering, software metrics, software testing, and some other fields, so the

Table 1. Topics presented within a course

Part I: Introduction to software engineering 1. What software engineering is 2. Quality criteria for software products 3. Software process models 4. Basic concepts for software development documents	*Part III: Software Design* 15. Overview of design activities 16. Structured design 17. Object-oriented design
Part II: Requirements engineering 5. Results of the "analysis and definition" phase 6. Cost estimation 7. Function-oriented view 8. Data-oriented view 9. Rule-oriented view 10. Structured analysis 11. State-oriented view 12. Scenario-oriented view 13. Object-oriented analysis 14. Formal software specification and program verification	*Part IV: Implementation and testing* 18. Implementation 19. Systematic structured testing 20. Functional testing *Part V: Advanced problems* 21. Software metrics 22. Maintenance 23. Reverse engineering 24. Quality of software development process and its standardization 25. Introduction to software ergonomics 26. User manuals 27. Project management 28. Configuration and version management

students are faced with the realistic results of those analysis, within four topics. Being sufficiently big, system and the results of the mentioned measurements, present adequate and satisfactory base for the explanation of all needed methodologies.

During the years of usage, more case-studies have been developed. At Skopje, FYR Macedonia, case-study covering classical functions of a university library was developed. Currently, at the University of Novi Sad, Serbia, two case-studies are arising. One of them presents a system for agent selling of consumer products, while the other is again adapted from [9] and is dealing with the control console for a car. This last case-study is especially important and different from the others by being the only one from a technical domain.

The idea behind the existence of several case-studies is the wish that the lecturers have a possibility to interchange those examples, depending on their needs and wish-es. This type of usage would require some deeper work by the lecturer, who would have to change presentations and examples, but is doable. Another, even more important moment, connected with the case-studies is the fact that they are used within the complex team assignments, used to assess the knowledge students gained during the lessons, and their ability to put it into practice. Within those assignments, it is often necessary to change the case-study used, in order to prevent students from cheating and taking the solution of previous generations. This requires almost no additional effort, since all of the case-studies contain all of the elements needed by all of the assignments.

2.3. Assignments for the course "Software Engineering"

The third essential component of the course is the assignments, created and prepared for team solving. Over the course conduction students are obliged to solve certain number of team assignments. As a common practice for all universities, it has been accepted that the students have to achieve 50% of the points for the assignments, but how are those points used later, is different. At some universities, this is just a condition that

students have to fulfill to be able to approach the exam. On other universities, besides being a condition for the exam, number of points gained for the assignments is used for calculation of the final grade.

With the assignments, the situation is the same as with the topics. Number of assignments created is much bigger than it is necessary for a successful realization of the practical part of the exam. This way, each of the lecturers has enough material to be able to choose those assignments that (s)he finds the most suitable compared to: the topics presented, quality and affinities of students, or compared to other courses available that semester at the university in question.

This gives the course flexibility, enabling usage of the assignments within crash-courses, one-semester course, and within longer, two-semester courses. Large number of assignments, and the fact that they are parameterized, gives lecturers also the possibility to exchange assignments over the years, so the plagiarism and copying of solutions is decreased to a bearable level. The pool of assignments consists of the following ones:

Assignment 1: Reading and reviewing of the preliminary requirements specification and requirements specification for a case-study "Seminar Organization" (or alternative). Students are supposed to find and correct errors, misunderstandings, and ambiguousness, and suggest the ways of improving the text. This assignment was generally created in order to test students' ability to present their ideas in a precise and concise manner;

Assignment 2: Application of the "Function point" method on the requirements specification, in order to calculate the price and the human resources needed for a chosen case-study. The purpose of this assignment was to create a habit for students to follow the rules and procedures they heard during classes;

Assignment 3: Analysis of a product model, resulting from the application of structure analysis. Again, "Seminar Organization" (or alternative) case-study is used, where students are faced with several data-flow diagrams (including some errors), and are required to notice those, and suggest the ways of improving the diagrams. Data-flow diagrams are taken from an important and distinguished book [9]. As a consequence, we hope to teach the students not to trust blindly to any authority, but to observe and check all of the information they reach;

Assignment 4: Development of a part of a static model by creation of class diagram and use-case diagram. Students are this time faced with a new, small problem, so their creativity is tested here.

This assignment was the one intended to check on students' ability to use UML methodology. While that seemed *not* to be the problem, a need for creativity that this assignment required, was one of the largest problems amongst the assignments, be-cause the usual fact was that the students were over-creative;

Assignment 5: Development of a formal specification for several new operations, based on formal specifications presented during lectures. With most of the students selecting this study direction because of their love for computers, this assignment has a purpose of showing them that they also need some knowledge in other, related fields, such as mathematics and formal logic;

Assignment 6: Analysis and review of another teams' solution of the fourth assignment "Development of the part of a static model". Students are here presented a different view on the same problem, a have to comment on it, and critic it. This gives students a chance not only to see the different view on the same problem, but also to try to assess the value of someone else's solution;

Assignment 7: Application of software metrics methods, through usage of a tool. This assignment faces students with the regular situation in a working life of a soft-ware developer, namely, with the need to find, install, learn, and use tool never seen before;

Assignment 8: Specification of a regression test. Students are required to define a set of test-cases that guarantees branch-coverage condition, using the tool for regression testing. This one, and the next one, introduces students with the most expensive, and probably the

most important part of the software development life-cycle, the testing process, and

Assignment 9: Creation of "classification tree" for software testing. Again using the "Seminar Organization" case-study, this time in combination with the tool for functional testing, students are required to define a set of test cases, and check the correctness of a program.

In practice, for all of the participating universities, the same procedure is applied: teams get their assignment and a term of no less than two weeks, to submit a solution. Team members are required to read and review the assignment and given material, to contemplate about it, and to create their version of a solution before the team meeting. Over (usually) several meetings, a team discusses individual judgments, and creates a common solution.

Occasionally, but compulsory after the first assignment is submitted (but not yet graded) a class is organized where the team of students who submitted the most intriguing solution for the first assignment, present it to other students. Decision, and classification of assignments so that "the most intriguing" solution is found, is up to the assistant. While the experience helps for making the right choice, we think that the presentation of *any* solution would be interesting. Namely, solutions *are* different, so each one will have their opponents, students who would challenge and confront it, so the fruitful discussion would happen in any case.

The rest of the teams, confronted with a different view on the same problem con-template, analyze, discuss, and critic suggested solution. Here we can also mention the cooperation of assistants from Humboldt University Berlin, Germany and University of Novi Sad, Serbia, who jointly created the most logical and most appropriate "correct solutions" for all of the assignments, based on several years of experience with those submitted. Typical, common errors are then presented to other students, while this solution is also used as a model for checking other assignments. Naturally, every year this "correct solution" is tested and further developed, through fruitful discussions with students.

Because of the trend noticed that some of the students participate less, or do not participate at all in some of the assignment solving, while the other students cover for them, at some universities the "experiments" started with the usage of wiki as a tool for the purpose of assignment solving. The idea behind this is to recognize, by care-fully reading through the history log of a learning management system, how much each of the team members participated in creation of final document. Since the application of this technique is still new, it is too easy to comment on it more. Still, the first experiences show that the better students are quite satisfied with this methodology, while those inclining to "cheating" at the exam had a lot of objections. Still, it is worth mentioning that the whole idea aroused from the pleas of students given in the surveys about their satisfaction with the course. Namely, there have been several cases where students asked the lecturers, to find the way of either punishing students not participating in assignment solving, or rewarding those who did most of the hard work.

Another characteristic problem with the assignments is a universal one, noticed at each participating university. Students, who are less ambitious, abandon their team as soon as they achieve minimal number of points needed. As mentioned, assessing the assignments is different amongst universities. So at some this means that students achieved 50% of the points are allowed to approach the exam, and they *are* not interested to learn additional methods and techniques. On other universities it additionally means that they are satisfied with the lower grade. In any case, this puts additional burden on those students willing to continue with the assignments. Assignments are created for team solving, so when only one or two students approach the work, they are much more difficult! Adequate and fair solution for this problem has not been found yet, and participants from several universities are working on it.

As mentioned, not all of the assignments are used each year at all universities. That decision depends on some subjective factors - choice and ideas of a lecturer, but also, and mostly objective

factors. Some of the assignments are based on usage of the tools that require significant financial investments and registration of the faculties for their usage, which is not always possible. At other faculties the course is shorter, so there is not enough time for all of the assignments to be conducted.

3. Difficulties And Peculiarities of the Course

Over the years, quantity of teaching material for the course grew to a significant size. There are 28 presentations with lecture notes included, 5 case-studies (more or less used and finished), 9 assignments, collection of around 500 examination questions, and many more. Also, most of those resources exist in several different languages, because interested lecturers were usually obliged by local laws to translate materials to local languages.

Under these circumstances, a natural problem arose. Modifications and improvements of the materials are hard to maintain, evolve, and spread throughout all of it. Even the corrections, improvements of style, grammar, typing errors, or occasional logical or material errors, are repeated over and over again. Between the team of core members of a project, a possibility to employ some kind of configuration management system is considered, but not yet utilized.

Even biggest problem is present at universities who use local, translated versions of the material, we must admit. Those lecturers improve and refine local versions, and hardly are able to find the time to send those refinements back to be used in English versions. So, two versions diverge from each other more and more each school-year, without realistic chance to become one again ever.

Discussing the above problem, the core members were able to recognize additional complication. Even if creators of the local versions find the time, collect all of the versions to send them back, the question would be – send them back to whom? Who is the one (or possibly more) person(s), who would be able to dedicate enough time to incorporate new ideas and findings, combine those coming from several different sources and in several different languages, and create a valid, refined new teaching material. The conclusion was that the solution would be if an employee could be found, permanently connected with the project, in charge of keeping the material up-to-date, of collecting and unifying changes made at different participating universities. Still and unfortunately, this idea is at the moment unsolvable, because there is no possibility for such a thing within a project.

A temporary solution for a problem of material unification occurs every now and then, in a form of an interested student! Several students at different universities, project members were employed to do certain tasks of common interest, as a part of their seminar papers, diploma, or master thesis. Not being a lasting solution, this option helps at least in a part, and decreases number of unsolved issues.

There is one addition to the grading process, as an experiment at some of the participating universities. Besides those generally agreed and used big team assignments, another type of "assignments" is also used. Namely, since "Bologna rules" of course conduction require regular students participation and course attendance, at some universities this was becoming a problem to some extent. In a situation when the course is conducted at master studies, the most of the students are regularly employed, and were unable to be present at all of the lectures. On the other hand, for some of those regular, undergraduate students, topics and lectures, but also the field in general, was not too interesting, while the course was obligatory. Their presence at the lectures was because of that more an annoyance to other students, than help to them, since they were not paying attention at all.

As a result, at some universities, a solution was found through a free interpretation of a notion of "course attendance". Not *all* of the present students were given points for attendance, but only those who actively participated in the lectures, answering (and asking) questions and commenting on presented materials. For questions asked during lecture presentations (possibly

and usually several during one lecture), students were able to earn so-called "bonus" points, and advance their grades. Those points could help a person to improve, but also make up for the points lost at tests, or within the assignments.

Very soon, only those interested in the field were present at the lectures, while the others were just involved in teamwork, and came to the final exam. This had good consequences on the class atmosphere and learning curve of those present. Still, in order to give the equal possibility to all of the students to earn points for participation, and interest them in the field, small assignments, requiring some thinking, searching, and researching were often given to students to be solved at home. The first person, who answers the given question by mail, or using the common "forum" of a learning management system used, would be awarded a bonus point.

4. The Other Courses Created Within the Project

Based on the good experiences and successful cooperation realized for the "Soft-ware Engineering" course, members of the project decided to extend their cooperation to other courses. So far, joint work has been conducted for the development of additional four courses, where some of them are already largely used, while the others are still partly in the development phase. All of those courses are related to the "Software engineering" course: either as a required pre-knowledge, further developed part of it, or simply belonging to the very close expert field.

- "Joint teaching materials on OOP using Java" [16] Java, is a subproject started very early, after the beginning of the project "Software Engineering: Computer Science Education and Research Cooperation", during the year 2004. The subproject is dealing with the development of joint teaching materials for several project participating institutions. Since this course already existed in curricula of all countries, the entire effort was invested to a pure educational and research cooperation, without triggering a complicated administrative procedure of introducing the new course, as it was the case with the course in "Software Engineering". Participants of this subproject are lecturers from six universities (project members). By joining their existing materials, refined and improved, but also by creating the new material, relatively fast a new course of a very high quality has been created. Such a new course is successfully conducted by six universities who contributed to its' development ;
- "Software Project Management" is a subproject started in 2004, with the aim of development of additional material for this very important subfield of "Software Engineering". For this purpose, mostly participant from the University of Novi Sad, Serbia and Humboldt University of Berlin, Germany were active, with the partial help from the University "St. Ciril and Methodius", Skopje, FYR Macedonia. So far, course is successfully conducted only in Novi Sad, Serbia, since the year 2005;
- "Advanced Compiler Construction" is a subproject started in 2004. The important issue here is that courses with this name already existed at the universities in Novi Sad and Belgrade, Serbia, and Humboldt University in Berlin, Germany. The main purpose of a subproject was to make those courses compatible, and improve them through the exchange of the existing, and creation of new teaching resources. This cooperation was also successful and the new course has been conducted for five years now, at mentioned universities. Possibility to transfer this and other developed courses to other universities, project participants is also considered, and probable in the future;
- "Data Structures and algorithms" is a subproject started in 2006. Within this one, Universities from Novi Sad, Serbia, and Skopje, FYR Macedonia were largely involved. Since course under this name and with the similar contents exists at all other universities, project participants assisted and helped in the development and review of the course;
- For the last three mentioned subprojects, the development of specially dedicated web-pages

is under construction, while the teaching materials that are developed so far, and are used in teaching, can be found at the local learning management systems of the participating universities. For Department of Mathematics and Informatics in Novi Sad, that page is available at [17].

5. Reactions and Opinions of Students

Almost everything we stated in this paper was the views from the position of lecturers. What about the other side? What students think about the course? For autumn workshops of the project participants, we have for years prepared reports with opinions and answers to the anonymous questionnaire we ask our students to fill. We will summarize those results here, and present part of the results.

For start, let us first recognize the character of our students. Even though their average grade is between 7 and 8 (60%) (on the scale from 6-10), only 32% of them between 8-9, and just a symbolical 8% over 9, their expectations stated before the course in "Software Engineering" were much higher. They stated that they will de-serve grade 10 (17%), or 9 (63%)! The rest of 20% said they will deserve the grade 8, and even that grade is above their average.

In reality, the problem with the course was *not* passing it, but grades were at the same level as for the other courses. Also, a general conclusion over the years was that the more students attended the lectures, their grades were higher. The more concrete questions and answers for bachelor students at the University of Novi Sad, Serbia, and master students of the Polytechnic University of Tirana, Albania were:

- Considering the question "Rate the amount of knowledge offered in the lectures" (where grades meant 5=too much, 1=too little): over the last five years, grades for the course were almost ideal, around 3. Grades given to the course by students of master studies in Tirana, were a little bit towards "too much", but we must admit here the existence of language problem, since the course is conducted in non-mother tongue;
- About the question "Rate the contents of the lecture" (5=too easy, 1=too difficult) in last five years, we received the following grades: 2.75, 2.78, 3.00, 3.00, and 3.04. Master student had on the average, almost the same opinion. Again, this gives the course almost ideally balanced difficulty of its' content;
- For the question "Is the course well structured", for the first time there is a signifi-cant difference in opinion. Students of undergraduate studies in Novi Sad rated the course on the scale 5=very well to 1=unstructured, with 3.4 on the average. Still, master students had a much higher opinion of a course, around 4.5;
- Similar difference was repeated with the question "Is the amount of information on slides adequate?" where undergraduate students rated the course with 3.3, while master students estimated it with 4.1. Without wishing to dispraise students of un-dergraduate studies, we estimate that master students have more pre-knowledge, and thus better chance to assess our course accordingly;
- The greatest difference in opinion was shown with the question "Are the slides well-structured and clearly arranged?" Grades from undergraduate students were between 3.4 and 3.7. At the same time, the lowest grade by master students was 4.4, while the other grades went up to 4.63;
- Both groups assessed very well knowledge of the lecturers (undergraduate around 4.3, masters around 4.7), their preparation and readiness for conducting the lectures (4.3 by undergraduate, almost 5 by masters), their engagement during the lectures (same as the previous question), and their willingness to answer questions (by both around 5, this time);
- Finally, when we consider some more general opinions about the course, situation is probably the best graded:
  - Did you learn a lot of new things (5=much, 1=not), grades were around 4.05 by the undergraduates, and around 4.20 by masters;

- Do you think that the content of the lectures was useful: 4.1 by undergraduates and 4.4 by masters;
- Overall rank of the course (5=very well, 1=bad) grades are 4 (with a very slight margin over the years) by undergraduates, and 4.5 by masters.

What we feel that must be noted here is that for the master students, attendance of the lectures was obligatory. For undergraduate students in Novi Sad, it wasn't. Yet, even though undergraduate students estimated that they attended only about 40% of the lectures, on the average, and that this fact forced them to spend more time both on studying and assignment solving, they felt qualified to assess presentations, lecturers and the course in general. What can give us optimistic bust is the fact that even those lower grades were very good, and that they prove that joint creation of common courses worth the effort.

6. Conclusions

Experience we gained so far during the ten years of creation and usage of common course and teaching resources developed by the project participants, can be in short enumerated with several basic results [18]:

- Courses are developed as a whole set of resources, containing presentations, but also lecture notes, assignments, case-studies, and all other necessary materials. Still, our experience shows that such a course can be adjusted to local curriculums, and also to style and needs of a lecturer, and be taught in different ways, at different universities, and different countries;
- Courses have been taught at different universities in a different manners and using a diverse subset of teaching materials, yet in each case those resources proved to be extremely useful;
- Examination and practice assignments were also used in different ways, but they also proved to be developed in a satisfactory quality and quantity to fulfill all of the needs arising;
- Exchange of the teaching materials is worthwhile. The development time is greatly shortened, guarantee for actuality and quality of the material is largely increased, exchange of experiences in enabled, and so is the exchange of technical and educational findings;
- Development of "lecture notes" enables usage of the teaching material and lecture conduction even to lecturers with less experience in the field that is taught. On the other hand, for those closer to the field, preparation time for the lectures was largely shortened;
- The validity of the previous two claims we can illustrate and prove by actual situations at Skopje University in FYR Macedonia, and Rijeka University in Croatia. Since the course was already conducted at Humboldt University in Berlin, Germany, and University of Novi Sad, Serbia, all of the needed materials were developed and practically tested. As a consequence, the whole course was rapidly introduced at Skopje University, where the professor and her assistants needed only two months to introduce the course. Still, this was the extreme case, caused by the fact that both professor and the assistant were the long time members of the project, that they have heard lectures during the workshops, heard the experiences with the assignments and case-studies, and so on. The other mentioned University of Rijeka is a more natural case that even better proves that this joint preparation of a course was worthwhile. Rijeka University is a member of our project, but a professor that introduced the course there, never was. She just took over our joint course, including all of the presentations, case-studies, assignments, and everything else, and within six months, she started conducting it successfully. We are still waiting for the written, numerical results of a survey conducted on students, but verbally given opinions and experiences are highly positive;
- Existence and usage of the common material enabled also exchange of experiences between lecturers, conductions of surveys and application of students' wishes and suggestions,

as much as the continual improvement of courses;
– The various experiences collected over the years have been described in several papers published over the years, at several conferences and journals: [12–14, 18–24].

Since the starting idea of the project was the exchange of experiences, rising of the teaching quality, and decreasing the effort needed for a creation of new courses, the above conclusions clearly prove that these aims are not only fulfilled, but surpassed by far. Based on the experiences gained with the first course in "Software Engineering", collaboration was extended to the development of new courses, already used in practice, but still refining and developing.

Currently, most of the efforts in a process of further refinement of the course are aimed at the development of appropriate e-Learning support for the course. Depend-ing on the University, these activities are in different phases. Universities in Novi Sad, Serbia, Skopje, FYR Macedonia, and Rijeka, Croatia incorporated joint materials into their learning managements systems, and students are freely using them. Even more, in Novi Sad, e-Lessons, glossaries, and quizzes for knowledge self-testing based on original presentations were developed, so that the students can choose the type of study resources they prefer.

Lecturers gathered around this project didn't stopped just to deal with the educational elements – great cooperating experiences with the development of new teaching materials have been deepened with the research cooperation. This cooperation extended over the limits of the courses that started it. Autumn each year is the time when participants of the project gather to exchange ideas and experiences, and to communicate and consult about the further educational and research efforts. Each year, these workshops include young assistants, but also the best students from the participating universities. Over the ten years of project existence, among the students that participated in the workshops, more than 10 have been selected as new assistants at various participating universities.

References

[1] "Bringing curriculums and equipment up to date," DAAD, Sep 2002, issue 3.
[2] "DAAD newsletter," http://www.daad.de/imperia/md/content/hochschulen/stabilitaetspakt/newsletter/2009_1-en.pdf, 2009.
[3] "Project home-page," http://www2.informatik.hu-berlin.de/swt/intkoop/jcse/, 2011.
[4] E.-E. Doberkat, C. Kopka, and G. Engels, "MuSofT – multimedia in der softwaretechnik," *Softwaretechnik-Trends*, Vol. Volume 24, No. 1, 2004.
[5] K. Modesitt, "International software engineering university consortium (iseuc), a glimpse into the future of university and industry collaboration," in *Proceedings of 15th CSEET*, Covington, Kentucky, USA, 2002, pp. 32–41.
[6] T. Hilburn, G. Hislop, M. Lutz, S. Mengel, and M. Sebern, "Software engineering course materials workshop," in *Proceedings of 16th CSEET*, Madrid, Spain, 2003.
[7] "Ariadne project," http://www.ariadne-eu.org, 2011.
[8] "Merlot project," http://www.merlot.org, 2011.
[9] H. Balzert, *Lehrbuch der Software-Technik*. Spektrum Akademischer Verlag, 1998, Vol. 1 and 2.
[10] "Computing curricula 2001, ACM and the computer society of the IEEE," http://www.acm.org, 2011.
[11] "Guide to the software engineering body of knowledge SWEBOK," IEEE Computer Science Press, 2001, eds. Bourque, P. and Dupuis, R.
[12] K. Bothe, K. Schuetzler, Z. Budimac, and K. Zdravkova, "Collaborative development of a multi-lingual software engineering course across countries," in *Proceedings of 35th ASEE/IEEE Frontiers in Education Conference*, Indianapolis, USA, 2005, pp. T1A–1 – T1A–5.
[13] K. Bothe and S. Joachim, "Tool support for developing multi-lingual course materials," in *Proceedings of ONLINE EDUCA Berlin, 10th Intl. Conference on Technology Supported Learning & Training*, Berlin, Germany, 2004.
[14] Z. Budimac, Z. Putnik, M. Ivanovic, K. Bothe, and K. Schuetzler, "On the assessment and self-assessment in a students teamwork based course on software engineering," *Computer Applications in Engineering Education*, Vol. Volume 19, No. 1, pp. 1–9, 2011.
[15] "Behavioral specification (requirements) of XCTL-control program," http://www2.

informatik.hu-berlin.de/swt/intkoop/jcse/case_studies/xctl/XCTL-Man-Adj.html, 2011.

[16] "Java course home page," http://perun.pmf.uns.ac.rs/java, 2011.

[17] "Course home page," http://perun.pmf.uns.ac.rs/moodle, 2011.

[18] K. Bothe, K. Schützler, Z. Budimac, Z. Putnik, M. Ivanovic, S. Stoyanov, A. Stoyanova-Doyceva, K. Zdravkova, B. Jakimovski, D. Bojic, I. Jurca, D. Kalpic, and B. ico, "Experience with shared teaching materials for software engineering across countries," in *Proceedings of Informatics Education Europe IV*, Freiburg, Germany, 2009, pp. 57–62.

[19] K. Bothe, K. Schuetzler, Z. Budimac, K. Zdravkova, D. Bojic, and S. Stoyanov, "Technical and managerial principles of a distributed cooperative development of a multi-lingual educational course," in *Proceedings of The 1st Balkan Conference in Informatics*, Thessaloniki, Greece, 2003, pp. 112–120.

[20] K. Bothe and S. Joachim, "Interactive tool-based production of multilingual teaching and learning materials," in *Proceedings of The 5th IEEE International Conference on Advanced Learning Techniques*, Kaohsiung, Taiwan, 2005, pp. 516–518.

[21] Z. Budimac, Z. Putnik, M. Ivanovic, and K. Bothe, "Common software engineering course: Experiences from different countries," in *Proceedings of The 1st International Conference on Computer Supported Education*, Lisboa, Portugal, 2009, pp. 375–378.

[22] M. Ivanovic, Z. Budimac, Z. Putnik, and K. Bothe, "Short comparison of tasks and achievements of different groups of students with the common software engineering course," in *Proceedings of The International Conference on Software Engineering Theory and Practice*, Orlando, USA, 2009, pp. 84–91.

[23] K. Zdravkova, K. Bothe, and Z. Budimac, "SETT-net: A network for software engineering training and teaching," in *Proceedings of The Information Technology Interfaces*, Cavtat, Croatia, 2003, pp. 281–286.

[24] ——, "The structure of SETT-net," in *Proceedings of The Eurocon*, Ljubljana, Slovenia, 2003, pp. 126–129.