Holistic approach in Education – Filling the Gap between Different Disciplines

Ivica Crnković[1], Karmela Aleksić-Maslać[2], Hrvoje Jerković[2]
[1]*Mälardalen University, Västerås, Sweden, ivica.crnkovic@mdh.se*
[2]*Zagrebačka škola ekonomije i managementa, Zagreb, Croatia, {kaleksic, hjerkovi}@zsem.hr*

Abstract. *In the research, education and in everyday practice, a need for and general understanding and a holistic approach is becoming more and more important. Still in concrete cases such approaches meet many challenges, mostly in form of misunderstanding between involved partners experts from different disciplines. Education in general does not provide training for such approach. This paper describes a case – a course which goal was to transfer knowledge from one area to another: A software Engineering and management of software development projects was taught to students of management and economy. In addition to this the course provided new teaching methods that students were not used to. Finally the course has been taught as a distance course, using internet-based technology. All this elements made the course very challenging. The paper gives an overview of the case, identifies the challenges and discusses the lessons learned.*

Keywords. Education, e-learning, management, software engineering, distant learning

1. Introduction

The standard way the modern society works implies distribution of work and specialization. In difference to ideals from renascence, where knowledge and skills from all possible disciplines are supposed to be embodied in the most prominent individuals, the modern society endorses focusing, specialization and excellence in particular small areas. The heroes of today are not people with overall knowledge in natural *and* social sciences *and* with perfect military *and* civil leadership which *also* are poets *and* musicians *and* athletes. The heroes of today are sportsmen specialized in particular branch of particular sport discipline, like runners on 100 meters or tennis experts on the grass. Important people are managers that successfully lead their companies without understanding the technology bases of the company, or technicians that come with innovative ideas and provide new technological solutions but do not need to have organizational skills. The reason of such "separation of concerns" the complexity of subjects that are impossible to grasp by individuals, and a tremendous pressures on the individuals to show the concrete results in a short time (often "just in time") and increasing global competition. Such separation of concerns brings many benefits, but it also exhibits its weak sides. Most of the activities require complicated procedures, large overhead and lack of overall understanding of the problems to be solved. Misunderstanding between different groups striving to the same goals, working in the same projects, companies, living together, is becoming more and more apparent. The society is becoming aware of it, and that is visible in strategic decisions for research and education framework programs [1].

Software is becoming a dominant aspect in many areas, either used as support, or as important part of the final product. In any type of modern development project, software plays an important role, either used for documentation, or planning, or as set of tools used to achieve the project results. Also there are more and more projects which goal is to develop software. Logically, the need of ability to manage software increases not only for software professionals, but also for other professions. Characteristic examples are development and project managers. The many years working experience in industry of one of the authors is presence of a deep misunderstanding between management groups and software developers. Such misunderstanding has many times led to suboptimal or unsuccessful results.

Education should play an important roll in helping to increase ability of educated people to grasp the whole and to manage the diversity. Unfortunately most of the educational systems still have focus on particular, selected areas. The requirements for the very deep, detailed knowledge leaves very little space for studying other, directly or indirectly related topics.

On the other hand experiences from pure holistic approach [2] have shown that there is a big challenge not to remain on a superficial level, and at the same time achieve a common understanding.

In this paper we describe a case study: Performance of a course which goal is to contribute to the holistic approach by bridging the gaps on several levels: (i) Providing a software engineering course with its roots in technical sciences to students of management and economy; (ii) applying a concept from one country (Sweden) to another (Croatia); (iii) introduce a combination of distance and local learning. The objective of the course is to provide students of management and economy with a basic knowledge in software engineering.

We describe the concept of the course, its performance and we analyze the results with emphasis on the challenges and lessons learned.

The rest of the paper is organized as follows. Section two describes the motivation, the overall goal of the course, and the main challenges such type of courses meets. Section three describes the course organization and its elements. Section four gives an overview of the technique used in the course. In section five the students' performance are analyzed, and the findings and lessons learned are discussed. Section six concludes the paper.

2. The case – software engineering course for future managers

2.1. Goals and objectives of the course

The goal of the course is to give insight in basics of software development to students of management and economy. By completing the course the students should be able to understand basic characteristics, processes and some of technologies for software design. Since software design and development are not trivial it would be naïve to expect achievement in deep understanding of all aspects of software development. Rather the goal is to (i) make students aware of these aspects, (ii) make students capable to understand the basic principles, (iii) train students to distinguish different solutions based on different technological assumptions, (iv) prepare the students to successfully participate in software development projects and (v) train students in communication and formation of their view of different aspects in software development projects.

To achieve these goals the following topics of software engineering have been selected as a part of the course:

- Basic characteristics of software
  - What makes software different? Why software development projects are different from other development projects?
- Software development models
  - Software lifecycles, software project models, activities in software projects, software decision models
- Software project management
  - Project organizations, stakeholders, project planning, project performance and follow up, project analysis
- Software lifecycle phases
  - Requirements specification and requirements engineering
  - Software design; Top-down design. Software architecture, Object-oriented design. UML – Unified modeling language (software modeling and specification)
  - Software maintenance
- Software quality
  - Quality assurance, laws and ethical principals

The second objective of the course was to combine traditions in teaching from different cultures. The idea was to apply Swedish style teaching to Croatian students. Although in many aspects similar, approaches in Croatian and Swedish education are somewhat different. Swedish education has both a tradition and strong trends in keeping education pragmatic, related very much to the principles "learning by doing". A second strong characteristic of Swedish education (inhered from Swedish tradition) is teamwork – a strong feeling for a team and sharing responsibility. A third characteristic in Swedish education system is exploration-type of education, focused on searching of knowledge when needed. Similar trends we can see in modern education, and this being introduced in the Croatian education system. However some other traditional elements like emphasis on theories, and acquiring knowledge in form of lectures and reading still are characteristic in Croatian education.

The third objective of the course was to train students to perform in different environments. The course was hold as a distance course – the lectures and seminar have been holding via video conference system.

2.2. The challenges

The concept of the course provides many novelties which in their turn were accompanied by several challenges. The course as designed included many challenges – some of them characteristic for software engineering [6], some of them for distance learning and some of them related to the introduction of new concepts and topics that students are not familiar with. We outline here the main challenges we met in the course.

Challenge 1 *Covering all disciplines vs. concentration on particular disciplines.* Software engineering is an extremely large area covering many disciplines. Teaching even only the most important disciplines requires a complete academic program, not just a course. All disciplines cannot be covered in one course. The problem is to select the most important aspects of software engineering and put them in a consistent set.

Challenge 2 *Striking a balance between theoretical knowledge and practical experience.* The main challenge is to prepare students for the real world, which is inconsistent and unpredictable. The academic world is often an "ideal" world in which students learn about problems and their solutions in a simplified form without all details. A very common solution to this problem is to execute projects in software engineering courses, based on real examples from industry [7], 8]. In our case this possibility was not realistic as for students of management and economy it would be a too complicated step. A simple, but realistic subjects and example were desired. Further, a dilemma is how much to weight the theoretical parts in relation to the practical part. Is it better to give the students a solid theoretical background, which they can utilize later in the "real life", or to "throw them into the water and let them learn how to swim"? Again to achieve a balance between a requirement for general understanding and in the same a feeling of "hands on" was a challenge.

Challenge 3 *To have a proper balance between permitting the students to work independently and under a degree of control.* One of the most important challenges is to establish good relations between students and their teachers. Teachers must be enthusiastic to make their students enthusiastic about their projects. On the other hand it is unsatisfactory if the teachers guide students too closely. In such a case there is a risk that students may stop thinking independently and begin to rely completely on the guidance of the teachers. In the case of the course it was an additional challenge since the lessons were performed without direct contact (except the first lesson).

Challenge 4 *Build new forms of teaching*. Main parts that required students' efforts were not typical for the students – project and teamwork. These forms of teaching are neither typical for students of social sciences, nor much used in Croatian education. The course did not have final exam, but the grade was the result of a student's performance during the course. Further the course was organized as a distance course using modern e-learning technologies.

Challenge 5 *(The grand Challenge) How to train students to manage something that is outside scope of their primary study.* To get a holistic and multidisciplinary view of a problem, different aspects of the problem must be studied. The challenge that arises is the following: Is it possible to provide an overall view, but also deep enough which will highlight the essence of the problem which will not require a deep technical knowledge of all these details? In the concrete case, the challenge was to give insight in software development to students which primary interest lies in other, possible very different areas.

3. The Organization of the course

The outline of the course follows a classical approach in software engineering course; it gives an overview of the most important phases and activities of software products lifecycles. The parts that are most important from a management point of view have been emphasized (requirements management, project management and system design), while others (for example software implementation, verification and validation) were only mentioned. Since the course focuses on the management part, the management aspects of the software project development have been explored in more detail. The main goal was to train students in ability to (i) understand the customers' requirements, (ii) understand the basic of software design, and (iii) be able to plan follow up and lead a software project. By such distribution of topics we tried to meet the challenge one.

The second challenge (a balance between theoretical knowledge and practical experience) was particularly difficult due to students' background, with almost no experience in software development, even with a limited experience in software usage. We could not count on any prerequisites in knowledge in computer science, computer or software engineering. Actually we realized that students are not necessary familiar with (or even had a feeling about) many terms usually used in computer science, or engineering. After a while we understood that we cannot used terms like software systems, building (implementation), source code, function, modules, etc. without carefully explaining their meaning. It is clear that pure theoretical knowledge would not help students to understand the topics. For this reason, in addition to lectures we have introduced laboratory exercises in the first phase and project work in the later phase of the course. The exercises included practical examples of the principles and methods presented on the lectures.

In the second phase of the course the students worked in projects. Project groups consisted of four to six students. The assignment was to make a project plan, identify requirements, and provide an overall deign of a system. The aim of the project twofold: a) gain practical experience in project planning and insight in software development process, b) getting experience in teamwork. The students performed the projects quite independent. They had responsibility to organize the project without strong influence or guidelines from teachers. By periodical follow-up they have were obliged to present the project state, and discuss possible problems and proposal for the solutions in the project. The idea of such type of performing was to increase the ability of taking decisions, increase the creativity and the responsibility in the team work. The project work was a means to meet challenges 4 and 5.

Finally the students get individual assignments – to review selected chapters of a book on software management and to write an essay on select topics.

4. The distance learning – the techniques used in the course

Without the help of the technology, especially Internet-based technology, it would be impossible to have this course. Technology had a major role in practically all curriculum activities.

Through the course students were obligated to listen to the 11 lectures, finish three laboratory practices in pairs and do final group project work.

Fast Internet connection between two places (lecture rooms in Croatia and the videoconference hall in Sweden) made possible to establish two videoconference systems at the same time. The first videoconference system was established with the specialized videoconference equipment which enabled us to transfer highest quality picture and sound of professor and students. The teacher from Sweden would give lectures and students in Croatia had opportunity to participate in the lectures through the first videoconference. In that way we achieved audiovisual interaction between the teacher and the students. The second videoconference connection was established for the purpose of presenting materials being lectured during the class. Presentation of the lectures and practical work were given through the separate projector over videoconferencing tool built in Microsoft Windows XP – NetMeeting. This enabled that programs used by the teacher during the lectures, students were able to see that on the second projector. This was a good way for students' practical understanding in resolving laboratory practices, project work and handling project documentation.

Everything that the teacher was doing on his computer; mouse movements, actions in programs and professors voice, was recorded and rendered later on in the form of the Flash movies. These movies students could download or view from the course web page. In this way students which were not able to attend lectures could see the lecture afterwards.

The WebCT [3] system was used as a repository of all data: lectures, exercise, messages. Students, professor and assistant on the course were able to communicate and exchange the files threw WebCT.

For the purpose of exchanging data and mails among the members of the project, special discussion groups were organized for each group. Within a discussion group each member of the group could post and upload their part of the assignment and see documents and posts that other students from the group have uploaded.

Since students which were taking the course were from different years, they had a lot of problems adjusting their schedules and trying to organize some time to meet for project work. Therefore we gave them opportunity to use real-time chat and whiteboard tool provided by WebCT. Chat could be used for communication and whiteboard for creating demonstrative project drawings, such as different UML diagrams. We have also encouraged students to use any of the VoIP (Voice over Internet Protocol) programs with whiteboard, in case they have broadband Internet connection at home.

Students were given tutorials in Flash which explained them how to use more complicated tools like the one for submitting assignments.

The professor and the assistant were also communicating through videoconferencing and VoIP programs (NetMeeting and Skype) [3] which enabled professor to communicate with assistant and to gain control over assistants computer to discuss the results of the students' work.

5. The course results and lessons learned

In total 20 students participated in the course and 11 passed the course. Four students more have possibility to pass it when completing their work. This is acceptable pass-through percentage although it is not the best one. Before we discuss the reasons why this relative low number of students passed, and what were other problems and their causes, we present first the students' evaluation.

The students' satisfaction with the course was in general quite high (see Figure 1), and the different elements of the course had approximately the same evaluation grade (Figure 2). The grade scale in the figures are from 0 to five, meaning five the best possible or the highest and 0 worthless or the lowest possible.

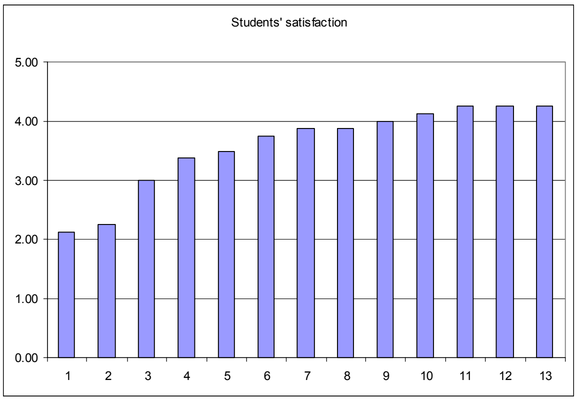

Figure 1. Students' satisfaction

In addition to these data the students provided a list or recommendation for the improvement, as well as the comment about which part they appreciated mostly.

The following comment dominated for the best parts of the course:

- The course concept that includes lectures, exercises, projects and reporting.
- Flexibility of the course.

The parts that according to students' opinion find as a problem:

- Distance learning decreases possibility of bidirectional communication.
- Difficulties in gaining motivation of some students that jeopardized the final project results.

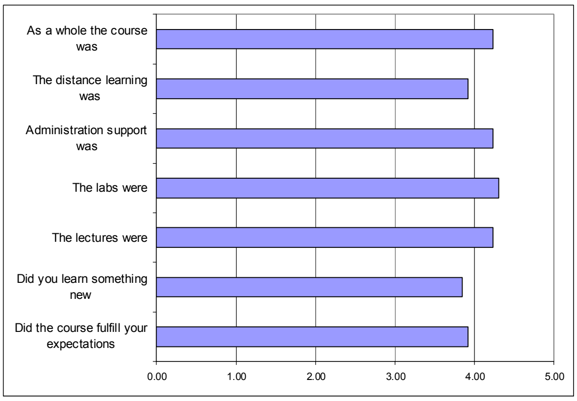

Figure 2. Students' evaluation of the course

Our main concern was the difference between ZSEM students and computer science students. ZSEM students were not acquainted with computer science terminology nor do they have proper foundation in fields of programming concepts. Besides that, do not use engineer's approaches and thinking while resolving problems.

Considering that, students' results have been in certain elements surprisingly good. The students have provided very good results in requirements analysis which were comparable within the computer science students. In addition, the students showed unexpected creativity in finding new requirements. Some of the results have been expected – the system specification was not in the rank as students of computer science would do. Some students had problems in understanding of concepts of object-oriented approaches, and some did not manage to provide solutions that followed UML formalism. On the other hand the most reviews and essays have been of higher quality that computer science students would produce. The best reviews were so good that the publisher company decided to give each student a book as a present.

However the most unexpected results from students work came from the project work. The results themselves have been of different quality. The quality and distributions of the results have however been expected. What was surprising (at least to the teacher from Sweden) was the attitude of students toward the team they belong.

Individual interests were strongly being preferred to the interest of the entire group. In most of the cases the teams work as groups of individuals rather then as coherent teams. The students have been more accurate to provide the individual tasks (when for example pointed from the teachers) than participate in the common work. Even when a common work was explicitly required, the students have divided the tasks and made them individually. The project work has clearly showed that additional training is required in teamwork. Similar experience the teacher had with international students vesting Sweden, and a similar experience has been reported in [2]. The lesson learned here is that introduction of teamwork requires additional efforts in teaching students about the teamwork patters, demands on the individuals, and abut the benefits. This finding is in line with students' evaluation and comments (both positive and negative). Similarly in many cases there was a clear discrepancy between the planned and the realized project activities, a well as planed and realized deadlines of the activities. While the teacher's intention was that the project plan shows a realistic plan that can be carried through, the students' intention was to present the plan as they thought that the teachers would like to see it. The students did not realize that they have to commit to the plan. What was the reason for that? The reason is definitely a lack of experience of working in project and in teamwork. The expectation from the teacher has been different as they have been based on the experience in working with Swedish students. The teamwork spirit is however built deeply in the Swedish culture and maintained carefully during the entire school and working life of people. Still, according to the teacher experience, better results in teamwork have been achieved in other distributed courses with one side placed in Croatia, but for students of electrical and software engineering [9]. In that case the students are used in performing project-type courses. Although the difference may be caused by many other factors, our finding is that a careful emphasis on teamwork can increase the team awareness and lead to better results.

We also have realized that the distance learning requires a good technical support, but also that the local support which increase the individual's evolvement in the group.

6. Conclusion

Our lessons learned from the course are that the holistic approach is not simple and it goes beyond bridging gaps between different communities by exchanging basic facts. In many cases the basic facts are not sufficient to completely describe particular. In addition to this we have learned that many informal, "cultural" habits can be barriers for achieving overall understanding. We are however convinced that the holistic approach is unavoidable in the modern education and that it can be steadily built up by learning about the differences of different approaches and areas.

Our intention is to continue to develop the course giving more attention to the teamwork and project follow-up.

7. References

[1] Specific Programmes implementing the Seventh Framework Programme, CORDIS, What is FP7? , http://www.cordis.lu/fp7/faq.htm

[2] Andreas Berglund, A framework to study learning in a complex learning environment, ATL-J, Research in Learning Technology, 12(1), pp 65-79, 2004

[3] WebCT Inc., www.webct.com, 2006.

[4] Skype Limited, www.skype.com, 2006.

[5] Ivica Crnkovic, Rikard Land, Andreas Sjögren, *Is Software Engineering Training Enough for Software Engineers?*, *(former)*, 16th International Conference on Software Engineering Education and Training, IEEE, Madrid, March, 2003

[6] Ivica Crnkovic, Magnus Larsson, Frank Lüders, *Implementation of a Software Engineering Course for Computer Science Students*, , Proceedings, APSEC, Asia-Pacific Software Engineering Conference, Singapore, December, 2000

[7] Dawson R.J., Newshman R.W., Kerridge R.S., "Bringing the 'Real Word' of Software Engineering to University Undergraduate Courses", *IEEE Proc. In Software Engineering*, 144, 5-6(1997), 44-48.

[8] Leventhal L.M., Mynatt B.T., "Components of typical undergraduate software enineering courses: Results from survey", *IEE Trans.Softw.Eng.*, vol SE-131,11(1987) 1193-1198

[9] Johan Fredriksson, Rikard Land, Mario Zagar Mikael Åkerholm, Ivica Crnkovic, Igor Cavrak, *On the Teaching of Distributed Software Development*, 25th International Conference Information Technology Interfaces, IEEE, Dubrovnik, Croatia, June, 2003