Technology in Real-Life Teaching of Distributed Software Development

Ivana Bosnić[1], Mario Žagar[2], Ivica Crnković[3]

[1, 2]University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, Zagreb, Croatia, {ivana.bosnic, mario.zagar}@fer.hr
[3]Mälardalen University, Högskoleplan 1, Västerås, Sweden, ivica.crnkovic@mdh.se

Keywords
distributed software development, education, global software engineering

1. ABSTRACT

Distributed Software Development course is a project-based course, currently run together on three universities: in Croatia, Italy and Sweden. The specificity of this course is usage of technology and distance work in order to teach students exactly that – distributed software development. Students experience what they are learning about through course participation. The goal is to offer students real-life experience of building such a distributed project, by working in distributed teams, having teaching staff and customers in remote universities, or participating in software competitions. Different educational activities build on the social constructivism method of learning, supported by various tools and technologies used. Course evaluation, which is being conducted since the beginning, for 11 years, shows constant and high students' satisfaction. Students' comments acknowledge the innovative concept and an opportunity to work in a real-life, distributed environment, with the help of technology.

2. DSD COURSE INTRODUCTION

Distributed Software Development course (DSD) is a joint course of three universities:

- University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia (FER)
- Mälardalen University, School of Innovation, Design and Engineering, Västerås, Sweden (MDH)
- Politecnico di Milano, Information Engineering School, Italy (POLIMI)

It is an elective course for students of Computing and Software Engineering Masters programs, given in a distributed manner for 11 years now, since 2003/04. The specificity of this course is the use of technology and distributed work to teach students just that – distributed software development. Students can experience the issues of working distributed through the way the course is carried out.

The main course goal is to prepare students for distributed work on software engineering (SE) projects, covering all SE project phases: defining the problem, gathering the requirements, creating the project plan, dividing the team roles, developing, testing, and documenting the product. Throughout these phases, students present their project status to convince the customers of the project's final success. All phases, more-or-less usual in project-oriented SE courses, are being augmented by the fact that everything is done in a distributed manner, with students and teachers at three universities. The main course part is project work, where ~40 students are divided into groups of 6-8 members (3-4 from each side). Project proposals are described briefly and technologies to be used are proposed. Each project has a supervisor (one teaching staff member) and customers (usually the other teaching staff members). As MDH is oriented towards international students, ~10 nations participate in the course every year, which poses more intercultural, language and teamwork challenges. Cultural differences can sometimes be observed in a way of communication and understanding among the different nations (Croats, Swedes, Indians, Italians…). Although these differences can reduce the communication effectiveness, this experience of cooperation with other nations and cultures is important for future engineers, so we consider it an additional course value. The official course language is English, a foreign language for most students.

Table 1 gives an overview on countries, number of students and number of projects per each year.

Table 1. Students enrolled in the DSD course

Year	# stud.	# proj.	Originating countries
2003	28	5	Croatia, Sweden, Canada
2004	20	4	Croatia, Czech Republic, Italy, India, Pakistan, Sweden
2005	38	6	Austria, China, Croatia, France, India, Nigeria, Pakistan, Spain, Sweden, Switzerland
2006	31	4	Bosnia and Herzegovina, Croatia, Greece, India, Iran, Pakistan, Spain, Sweden
2007	20	2	Austria, Croatia, Spain, Sweden, Thailand
2008	37	6	Australia, Croatia, India, Iran, Italy, the Netherlands, Pakistan, Spain, Sweden
2009	56	10	Bangladesh, Croatia, France, Germany, India, Iran, Italy, Lithuania, the Netherlands, Pakistan, Sweden, Ukraine
2010	65	9	Bangladesh, China, Croatia, France, Germany, India, Iran, Italy, Kazakhstan, Kenya, the Netherlands, Pakistan, Sweden
2011	35	5	China, Croatia, India, Italy, Kenya, the Netherlands, Pakistan, Poland, Venezuela
2012	49	7	Argentina, Bulgaria, China, Croatia, Germany, Italy, Macedonia, Nepal, Pakistan, Portugal, Sweden, Uzbekistan

Both theoretical part (lectures) and practical part (project work) are conducted remotely, using communication technologies. Such an environment offers various educational goals and innovative approaches to teaching and learning described in the rest of the paper. This gives students an opportunity to face challenges which they would not experience in a local setting, without the use of technology.

3. INNOVATION

The field of global/distributed software development is fairly new and a very actual one. There is a great need for knowledge and experience of future engineers in this field, but not many courses have an answer to it. Even less courses give students hand-on experience in distributed software development; some examples of similar courses offered around the globe are (Nordio et al., 2011), (Gloor et al., 2011), (Meyer & Piccioni, 2008) and (Bruegge, Dutoit, Kobylinski, & Teubner, 2000).

From another point of view, the majority of e-courses, implemented with the support of technology, use more or less basic functionalities of learning management systems, such as publishing educational content, providing assignments or tests online, and offering some communication tools. When teaching using videoconferences, often such lectures offer low level of interactivity.

DSD course focuses on project work of distributed project teams, with support of technology in all stages of distributed work, to provide joint lectures, communication, collaboration, and working on projects outside their university.

Using technology and different tools, a virtual space for both students and staff is created. In student teams but also in staff team, all sides need to act as one. Compared to similar courses, DSD course is considered to have *tightly-coupled* connections among sides involved with joint decisions, from organizational agreements to joint grading.

An important role in the course is the experience of real-life environment. Students have the opportunity to work not only with teaching staff, but with external partners as project customers. Companies of different characteristics cooperate in the course, with a representative who is involved in a project. Depending on the project setup, external customer is involved in project definition, requirements gathering and cooperates with team members throughout the course. Such collaboration is mainly done with the help of technology, and the distance involved enhances the real-life aspect of learning.

Student competitions are an added value to the course. In some years, a part of project teams are also involved in a software engineering competition. This again involves external project proponents, with whom students have to cooperate. This involvement is described in more detail in Section 6.

4. PEDAGOGY AND INSTRUCTIONAL DESIGN

Initial design of the course was prepared in 2003/04 based on the instructional design ADDIE model:

- In the **analysis** phase, the current requirements for our students – future software engineers - knowledge and skills needed for the distributed development were analyzed. Besides the specific skills, students were lacking the real project work experience, but also the experience in various non-technical skills, such as collaboration, communication or presenting.
- In the **design** and **development** phase, discussions and compromises between universities needed to take place, due to various institutional and organizational differences, such as the different academic year calendars, number of ECTS points, grading schemes, etc.
- In the **implementation** phase, an approach of **social constructivism** was used, including individual research, peer discussion, self-assessment and student peer reviews, with the support from teaching staff.
- Yearly course changes take place in the **evaluation** phase, based on the following feedback:
    - self-assessment reports from each student, which show us the project work problems
    - anonymous questionnaires, which offer a safe speakers' corner environment
    - unofficial discussions with students during the course

One teaching assistant – supervisor – is assigned to take care of each project team. He is usually from the same faculty as the student project leader, to ensure easier communication in the beginning. Teaching assistants help to solve the problems in the team, which are less of a technical nature, but more problems of organization, communication and cultural differences. According to *Teaching Perspectives Inventory* (Pratt, 1998), the relation between staff and students in that way becomes less of a *transmission teaching type*, and is much closer to *nurturing type*, as students build their knowledge. Student – staff communication is based on e-mail, instant messaging tools and in-person talks. Communication with the remote team site is especially important, to understand the work and difficulties from the distant side. On rare occasions, it is possible for the supervisor to visit team members in another country in person. Such meetings are very valuable and encouraged, but as they cannot be accomplished for every team every year, we consider them an added value, not something strictly needed to be done.

4.1. Educational goals and connected activities

The main course goal, introducing students to distributed software development, is achieved through a set of educational goals:

- **Obtaining basic theoretical knowledge on DSD** – in the course beginning, students listen to a short series of lectures from both universities. The topics are oriented towards general DSD and social-skills topics, like cultural differences, presenting and team work.
- **Gathering experience from industry professionals** – as a part of introductory lectures, several guest lecturers - industry professionals - present their experiences of distributed collaboration and give advices from the field, for intercultural team work.

- **Simulating the real-world environment** - to move project work closer to the real-world situation, we promote course *role-playing*:
    - the course professors become the project customers, who have a general idea about the future product, but are not concerned with requirement and development details
    - one of the teaching assistants for each project becomes the project supervisor, who closely supervises the project progress
    - one of the students at the site of the customer becomes the project leader, with a big responsibility for the project success, while another student at the remote site becomes the leader of a remote part of the team.
- **Working with external customers** - to make the experience of a real-world environment stronger, the following scenarios of including external customers are involved:
    - **industry cooperation**: several IT companies play the role of a project customer, which enables students to work in a real setting, and to solve a specific real-world problem. The company employees are actively involved in the project work, by giving students the advices and support, as well as discussing the project in the real-world setting.
    - **Software engineering competition:** when available, student teams participate in a software engineering contest organized by the most popular conference on software engineering, *International Conference on Software Engineering (ICSE)*. The conference participation and presentation is a valuable experience for students, not available in the conventional course approach.
- **Improving presentation skills** – during the semester, students describe their project status through various reports, but also through 5-6 distributed presentations. This enables them to practice how to make good presentations, communicate with the audience, give public speeches in the foreign language, try to advertise their own work, but also to use the professional communication equipment and videoconferencing tools. For most students, this is the first such experience during their higher education.
- **Transferring knowledge among the students** – before the project work starts, students present to each other course-related topics, which enables them to practice presenting in a foreign language, and to receive several advices.
- **Improving collaboration skills and responsibility** – students are regularly encouraged to show responsibility and collaborate inside and among the project teams. They maintain the project web pages and collaboration groupware pages, decide on the scope of their projects, divide the videoconference slots among the teams, etc.
- **Developing self-assessment abilities** – in the beginning of the course, students have to assess their own knowledge on several aspects. Throughout the course, students complete the course questionnaire, where they describe their project work experiences, the efficiency of collaboration between local and distributed teams, wrong decisions and estimates being made. The questionnaire quality influences their final course grade, as this course puts more emphasis on the learning process (even with the mistakes) than on the final product delivery.
- **Learning to peer-review** – In the end of the course, the projects are evaluated on ~40 criteria, which include not only the product quality, but also the work process (presentations, deadlines, workflow, etc). Based on the overall project grade, the project leader proposes project grades for the other team members. During the semester, teaching staff regularly talks with team members to discover the possible team problems which could influence the peer grading quality.

As this is an e-learning course, focused on distributed student teams, almost all educational goals need to be supported by various technologies and tools for online education, described in the following section.

5. TECHNOLOGY-ENHANCED LEARNING IN DSD COURSE

By its characteristics, DSD course is oriented towards the whole set of information and communication technologies. Designed as a course which offers students knowledge and initial experiences of distributed software development, DSD grounds the biggest part of its educational activities on project-based tasks, where students of three universities work together. In such environment both students and teaching staff use content management systems, collaborative and social tools and a number of communication tools to achieve the basic course goal – distributed software engineering project work. This creates the joint virtual community space, for students as well as for staff.

Having said that, it is clear that this is not a conventional e-learning course, where students follow the course structure in a learning management system, with higher or lower relation to e-learning methods. This course uses a plethora of tools and technologies, as a natural means to achieve such distributed work. Technology enhanced learning in this course is "a must" and the whole course is rooted in it, from joint lectures, through collaborative software engineering work and communication, to delivering status presentations, final products and feedback from both students and staff.

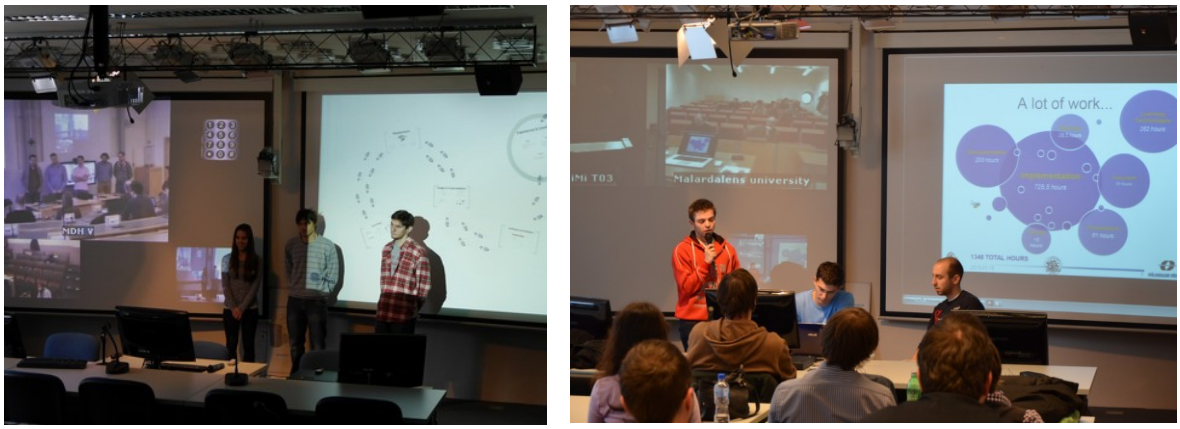

Figure 1. Joint classrooms

The main criteria for our choices of tools are their stability and availability, as the change of tools during the course year would cause a lot of problems. Most often, highly popular freeware and open source tools are used. It should be noted that students play an important role selection of tools; they mostly have the freedom to choose their own ways of collaborating. Their choice of tools influences our future recommendations for the next year. Each course year brings new experiences in the use of technology, which contributes to overall knowledge on distributed project work.

Table 2 presents some of the tools and technologies used in the course.

Table 2. Tools and technologies used in the course

Course requirement	Used for	Tools used
Synchronous in-class communication	Audio and video conferences, desktop sharing	*Polycom, Skype, NetMeeting, Adobe Connect*
Synchronous collaboration	Instant messaging	*Skype, MSN Messenger, ICQ*
Asynchronous collaboration	news, sharing and collaborative document editing, polls and questionnaires, discussion groups	*FER CMS, Google Groups, Google Docs, Google Poll, Doodle*
Software development collaboration	versioning system, bug reporting software, project managment	*SVN, Git, BugZilla, Redmine*

As a project-based course, DSD does not emphasize learning from educational content prepared in advance, but on the personal experience of collaborative work. Therefore the lectures are not specially designed for any learning management system. Instead, they are conducted jointly for all

universities, in their videoconferencing rooms. For **synchronous in-class communication** - video and audio streaming, *Polycom* and *Adobe Connect* systems are used for video streaming of both lectures and student presentations. Teams are strongly encouraged to deliver their status presentations from more than one side, which shows the effort they spent in collaborative work on preparing a presentation. In order to deliver good presentations over a distance, a lecture on preparing and delivering remote presentations is given at the beginning of the course.

The **content management system** developed in-house at FER has proven to be a good solution to connect the students and the staff. All sites have access to this official course web page, hosting team information, project news and deliverables. Each team maintains its own pages – a general project page with photos and information about all team members, *News page* and *Documents page*. Traditional learning management systems (*Moodle*, *WebCT*…) are not needed as the course is completely project-oriented.

Due to project-based activities, everyday communication is necessary for both local and distributed side. For the successful project progress, team members should be well-acquainted with their colleagues. Therefore personal communication is encouraged, to build the necessary trust among team members. A project leader and a team leader should become "the best friends". Usually technical issues are being solved immediately using the IM tools, which gives the notion of a *virtual team* created.

Project teams have the freedom to choose the means of internal communication. For **synchronous collaboration**, most communication takes place using the Instant Messaging tools. It was interesting to observe the different preferences of students from two universities: while the students in Sweden prefer *MSN Messenger*, the students in Croatia mostly use *Skype*, so the agreement has to be made.

Students prefer using **text-based communication** possibility, rather than using audio or video capabilities. There are several reasons for that. Having in mind that English is not the first language of any students, and that some students have heavy English accents, it is sometimes hard to understand what is being said. The second reason is the possibility to have more time to think about the question or idea proposed while typing, than in audio communication. The third reason is a very practical one – keeping easily searchable logs of all that is said during the meeting is important. Although students write *Minutes of meeting* documents afterwards, sometimes going through the logs to find some particular information is needed.

Asynchronous communication is based on e-mail solutions. In the first years of the course, students mainly used mailing lists. Later on, especially with the advance of *Web2.0* concept, students started using various Web collaboration tools. The first students' choice is *Google Groups*, a Web site which offers file sharing, discussion groups, pools and joint mailing list address. Through the use of *Google Docs* office tools, they often write project documents together, at least in the draft phases. This enables a better project organization and the information flow, together with easier document versioning.

For collaboration on software development in particular, we tested several open solutions, especially in code versioning: *CVS*, *SVN* and *Git*. Currently, *Git* – on *GitHub* - is used as a mandatory. However, in project management and other utility tools, students have complete freedom, to both choose the tools, and also to choose which project tasks will be managed in that way. Some groups use bug reporting tools such as *BugZilla*, the other use project management solutions, such as *Redmine,* to manage their deadlines and subtasks. They are strongly advised to use open source solutions to get more introduced to open software development culture.

In a survey from the 5-year course period (2004–2008.), communication methods used by students were analyzed (Crnkovic, Bosnic, & Zagar, 2012). Altogether, 119 students' questionnaires were included in the analysis. Since in those years the questionnaire items were mainly answered in a qualitative manner, the answers had to be analyzed and grouped by their characteristics. The results presented in Figure 2 show what percentage of those students included in the analysis used a particular communication method. The data shows that instant messengers are most often used method of communicating.

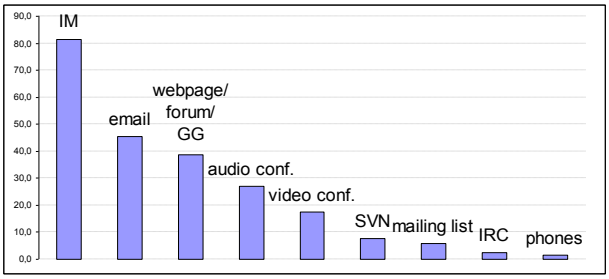

Figure 2. Communications methods usage

6. COURSE EVALUATION

6.1. Course Questionnaire setup

Since the beginning of the course in 2003, an internal, voluntary and anonymous course evaluation is conducted at the end of each course year. Using the questionnaire, students help us to see possible problems with the course, by grading and commenting on the following course aspects:

1. As a whole, the course was *(1-very bad; 5-excellent)*
2. The course has fulfilled my expectations *(1-not at all; 5-completely fulfilled)*
3. The concept of the course with lectures and projects was *(1-bad; 5-excellent)*
4. The course administration (web page, support, information, etc) was *(1-bad; 5-excellent)*
5. I have learned *(1-nothing; 5-a lot)*
6. The lectures were *(1-bad; 5-excellent)*
7. The guest lectures were *(1-bad; 5-excellent)*
8. I liked to work in the project *(1-not at all; 5-I loved it)*
9. The project advising and support was *(1-bad; 5-excellent)*
10. The cooperation between FER/POLIMI/MDH students (in your project) was: *(1-bad;5-excellent)*
11. My workload was *(1-nothing; 5-very heavy)*
12. The equipment for the distance work was *(1-bad; 5-excellent)*

In addition, another two opportunities for "speaker's corner" were provided by asking for:

- The things I most liked in the course
- The following can be improved in the course

6.2. Course Questionnaire results

Out of 379 course participants throughout 10 years, 264 completed the questionnaire, as it was not obligatory. A part of previous years' results is elaborated upon in (Feljan, Crnkovic, Bosnic, Orlic, & Zagar, 2012). Full evaluation results and comments per year are available at DSD course web page ("DSD course, the official site," 2013). In this paper, results of two statements are presented:

Statement 1 - **As a whole, the course was** (1-very bad; 5-excellent)

Statement 2 - **The course has fulfilled my expectations** (1-not at all; 5-completely fulfilled)

The first statement is taken as a general students' view after taking the course. The second one relates students' satisfaction with the course. The following table presents the grades given by students per each year.

Table 3. Average of grades given by students for two questionnaire statements

Year	03/04	04/05	05/06	06/07	07/08	08/09	09/10	10/11	11/12	12/13	Avg
Students#	**21**	**52**	**44**	**26**	**11**	**28**	**36**	**15**	**22**	**9**	**28.33**
Statement1	4.71	4.58	4.32	4.08	4.36	4.14	4.67	4.53	4.23	4.11	**4.38**
Statement2	4.48	4.29	4.02	3.88	4.27	4.11	4.19	4.18	4.09	4.11	**4.16**

The evaluation grades are constantly high, especially having in mind that students come from very diverse environments, different countries, etc. Going into details with particular questions reflects various circumstances in each academic year, such as involvement of external partners, a higher number of students than usual, etc. In general, although students find the course hard-working and demanding, they consider it a good and useful experience, different from conventional courses.

Some student statements gathered from the questionnaire about the course in general are:

- *"This course was one of the best that I had on faculty."*
- *"I was scared at the beginning, but now I feel lucky I had been a part of this course."*
- *"Another week and I would have died."*
- *"...I found out that it takes a lot of work to make something work as you want it to work."*
- *"I learned much more than I expected. I learn lot of team working, good organization of work, time and documentation importance."*
- *"I honestly expected much less of this course than i got."*
- *"I expected hard work, and that is what i got. Pain and suffering was present during the whole project, but in the end all paid off."*
- *"When getting at the FER I thought that every course will be like this. Soon I changed my mind and thought that something like this exists only in america. I'm very glad I found something like this at FER."*

An excerpt of comments regarding technology enhanced learning:

- *"All the free technologies are enough this days to work on such a project. Also, all the technology on universities was great also, so we had no major problems."*
- *"the equipment in the lecture room was good, and we did not have any communication problem other than one lecture. Then the professors have to deliver lectures seperately to both the destinations."*
- *"Definitely not sufficient on the PoliMi side, with only 1 monitor to show screens and cameras. Also Adobe Connect is not the right solution IMHO, and using a personal laptop instead of that of the teacher becomes very hard."*
- *"Skype is more than enough. :D"*

These comments show that a level of technology support should be ensured for a good course delivery. Also, an experience from both students and staff is that various backup plans are needed when dealing with such amount of technology, especially with video conferencing, doing live demos project demos, etc. If all else fails, there must be a way to even deliver staff lectures or student presentations locally. All this requires more time and preparation to ensure the high quality of a distributed course.

Based on students' comments, and our own observances, the course is being adapted every year, at least in some details. These changes are described in (Feljan, Crnkovic, Bosnic, Orlic, & Zagar, 2012), grouped in several categories: *Technical resources*, *Knowledge level*, *Project selection and assignment*, *Lectures*, *Course organization*, *Workload*, *Course advising*, *Grading* and *Other issues*. Besides examples of adaptations given there, it is important for us to explain students why something is done in that way in the course, if we conclude that this is currently the best approach to that particular problem.

6.3. Course awards and team competitions

During implementation of this course, this course has been recognized as a successful implementation of technology enhanced learning with a clear goal. It has been given two awards:

1. At the "**International e-Learning Awards 2010**" contest, organized by IELA - International E-Learning Association, the course **won the second award,** in the category of **academic e-learning** ("IELA - International E-Learning Awards - Past Winners," 2013). The awards were announced during the International Conference on Computer Aided Learning - ICL 2010, in

Hasselt, Belgium. Submissions were received from 26 countries, and awards were given in academic and business categories, for e-learning, mobile learning and blended learning.

2. At **The best e-course contest at University of Zagreb**, in the academic year 2008/2009, DSD course was awarded the second prize. The purpose of this competition was to acknowledge the best examples of using ICT in university education, which enhances not only technological, but the pedagogical parts of the course education process.

In addition to this, DSD project teams won several awards in **software engineering contests**. We regularly participate in competitions named SCORE ("SCORE Contests home page," 2014), organized by the world's largest software engineering conference (ICSE – *International Conference in Software Engineering*). Teams compete on project work, which fulfills both course and contest requirements.

Here is the summary of student results in all three contest instances held by now:

1. **2012/2013:** in the third SCORE competition, students were involved in projects regarding Open data, creating various services for citizens. Out of more than 50 registered teams, three teams were invited to ICSE 2013 conference, held in San Francisco, USA. DSD project team *Travel'n'study* became an **overall SCORE winner**.
2. **2010/2011**: in the second SCORE competition, students were working on projects proposed by professors from universities and research centers all over the world, communicating with them on a distance. 94 teams from 22 countries participated in the contest. Four DSD teams were accepted among 18 teams for the second round. Two DSD teams, *Mass Observation* and *Public Transportation Product Line*, are selected among the best five teams - **finalists**. Their representatives were invited, with paid expenses, to ICSE 2011 conference, which took place in Hawaii, USA, in the end of May, to present their projects. Their specific - distributed - way of working was received with great interest.
3. **2008/2009**: in the first year of SCORE competition, six finalist teams were selected out of 50 project teams. Three out of six finalists were from DSD course. Project *BTW: if you go my advice to you* was declared the **overall winner of the SCORE competition.**

7. CONCLUSION

Having in mind all the aspects of the course, from a high level of technology use, to a number of team activities where students build their knowledge, this approach to education requires more efforts from the conventional, or even resource-based e-learning approach, for both students and the staff. In case of endorsing such an approach for the other courses, the benefit - which would vary depending on the course topic and goals - should be evaluated. However, for educational goals of this course, the proposed way of learning proves to be more suitable to students' future needs, more motivating and useful, therefore we find that advantages justify the invested effort.

8. REFERENCES

Bruegge, B., Dutoit, A. H., Kobylinski, R., & Teubner, G. (2000). Transatlantic project courses in a university environment. *Proceedings Seventh AsiaPacific Software Engeering Conference APSEC 2000*, 30–37. doi:10.1109/APSEC.2000.896680

Crnkovic, I., Bosnic, I., & Zagar, M. (2012). Ten tips to succeed in Global Software Engineering education. In *Software Engineering (ICSE), 2012 34th International Conference on* (pp. 1225–1234). doi:10.1109/ICSE.2012.6227020

DSD course, the official site. (2013). Retrieved January 25, 2011, from http://www.fer.hr/rasip/dsd/

Feljan, J., Crnkovic, I., Bosnic, I., Orlic, M., & Zagar, M. (2012). Distributed Software Development course: Students' and teachers' perspectives. In *2012 Second International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD)* (pp. 16–20). IEEE. doi:10.1109/CTGDSD.2012.6226943

Gloor, P., Paasivaara, M., Lassenius, C., Schoder, D., Fischbach, K., & Miller, C. (2011). Teaching a global project course: experiences and lessons learned. In *Proceedings of Collaborative Teaching of*

Globally Distributed Software Development Community Building Workshop CTGDSD 2011 (pp. 1–5). Retrieved from http://www.ickn.org/documents/GSETeaching_final.pdf

IELA - International E-Learning Awards - Past Winners. (2013). Retrieved January 31, 2014, from http://www.ielassoc.org/awards_program/past_winners_AD_2010.html

Meyer, B., & Piccioni, M. (2008). The Allure and Risks of a Deployable Software Engineering Project: Experiences with Both Local and Distributed Development. *2008 21st Conference on Software Engineering Education and Training*, 3–16. doi:10.1109/CSEET.2008.41

Nordio, M., Ghezzi, C., Milano, P., Nitto, E. Di, Tamburrelli, G., & Aguirre, N. (2011). Teaching software engineering using globally distributed projects: the DOSE course. *Human Factors*, 36–40. Retrieved from http://se.ethz.ch/~meyer/publications/teaching/global.pdf

Pratt, D. D. (1998). *Five Perspectives on Teaching in Adult and Higher Education* (p. 304). Krieger Publishing Company. Retrieved from http://www.amazon.com/Perspectives-Teaching-Adult-Higher-Education/dp/089464937X

SCORE Contests home page. (2014). Retrieved January 30, 2014, from http://score-contest.org/

9. ACKNOWLEDGMENTS

The authors would like to thank all partners from universities which have been involved with the course in particular years: University of Paderborn (Germany), Charles University (Prague, Czech Republic) and Politecnico di Milano (Italy) who joined our course cooperation. We would also like to thank our external project partners, ABB Corporate Research (Sweden), Ericsson Nikola Tesla (Croatia) and Kapsch TIS (Croatia).

10. AUTHORS' BIOGRAPHIES

Ivana Bosnić, PhD CS is a postdoctoral researcher at the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia. Her primary interests include technology enhanced learning, content reuse and recommendation, open educational technologies, learning management systems, as well as applying e-learning in software engineering education. She is the chair of "Open educational technologies and content" group, at "Croatian Society for Open Systems and Internet". She is a member of IEEE. More information is available on http://www.linkedin.com/in/ibosnic

Ivica Crnković is a professor of industrial software engineering at Mälardalen University, Sweden where he is the scientific leader of the software engineering research. His research interests include component-based software engineering, software architecture, software configuration management, software development environments and tools, as well as software engineering in general. Professor Crnković is the author of more than 180 refereed articles and papers on software engineering topics and a co-author of two books. His teaching activities cover several courses in the area of Software Engineering undergraduate and graduate courses. Professor Crnković received an M.Sc. in electrical engineering in 1979, an M.Sc. in theoretical physics in 1984, and a Ph.D. in computer science in 1991, all from the University of Zagreb, Croatia. More information is available on http://www.idt.mdh.se/~icc/

Mario Žagar, professor of computing at the University of Zagreb, received Dipl.ing., M.Sc.CS and Ph.D.CS degrees, all from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER) in 1975, 1978, 1985 respectively. In 1977 M. Žagar joined FER and since then has been involved in different scientific projects and educational activities. He received British Council fellowship (UMIST - Manchester, 1983) and Fulbright fellowship (UCSB - Santa Barbara, 1983/84). His current professional interests include: computer architectures, design automation, real-time microcomputers, distributed measurements/control, ubiquitous/pervasive computing, open computing and e-learning. M. Žagar is author/co-author of 5 books and about 100 scientific/professional journal and conference papers. He is Senior member of IEEE/CS.