Avoiding Scylla and Charybdis in Distributed Software Development Course

Ivana Bosnić, Igor Čavrak, Marin Orlić, Mario Žagar
University of Zagreb, Croatia
Faculty of Electrical Engineering and Computing
{first-name.last-name}@fer.hr

Ivica Crnković
Mälardalen University, Sweden
School of Innovation, Design and Engineering
ivica.crnkovic@mdh.se

ABSTRACT
Teaching Distributed Software Development (DSD) is a challenging task. A convincing simulation of distributed environment in a local environment is practically impossible. Teaching DSD in distributed environment is more realistic since the students directly experience all its specifics. However, teaching in distributed environment, in which several geographically separated teams participate, is very demanding. Different types of obstacles occur, from administrative and organizational to technical ones. This paper describes some of the challenges, lessons learned, but also success stories of the DSD course performed now eight year in a row.

Categories and Subject Descriptors
D.2.9 Management, D.2.10 Design

General Terms
Design, Human Factors. Teaching, e-learning

Keywords
Distributed development, Teaching, E-learning, Project management, Teamwork.

1. INTRODUCTION
The increased globalization of software development is creating new challenges due to the impact of time zones [7], distance [9], diversity of culture [6], and communication, which requires novel techniques, tools and practices to overcome numerous difficulties. Wide and continuously increasing practice in distributed software development (DSD) was so far modestly followed in education. Although software engineering education emphasizes importance of practice as pointed in [12], courses in DSD are still rather rare. The main reason is difficulty in establishing distributed, and yet common environment, supported by more than one site. An existence of at least two different, geographically separated sites is a basic precondition to establish a realistic distributed development support [11]. This makes everything at least as twice as difficult; first – to prepare a common course and comply with local administrative rules on each side, and second – to provide effective technical support during course execution.

This paper discusses the experience of a course "Distributed Software Development" [5],[4], organized and performed between the School for Innovation, Design and Engineering at Mälardalen University (MDU), Sweden, the University of Zagreb Faculty of Electrical Engineering and Computing (FER), Croatia, and partially joined by the University of Paderborn (UPB), Germany.

The outline of the paper is as follows: in section 2 we briefly present the history of DSD course. Section 3 describes the challenges in creating a joint DSD course (organizational, distance-related and cultural) and provides a description of DSD course solutions to these issues. Section 4 gives lessons learned. Some success stories are described in Section 5. The conclusion is given in Section 6.

2. THE HISTORY OF DSD COURSE
DSD course [3] has been successfully carried out for eight years, starting in 2003. The course is designed as a combination of lectures, guest presentations and distributed projects, executed jointly with the help of Internet-based tools. Most of students' time and effort is directed towards the project work. All elements in the course are performed jointly by both sites: the lectures are common and equally distributed between the sites, transmitted using a video-conference system, students from both sides participate in them; finally, shared projects are performed in at both sites. The lectures cover three areas: (i) technical, which includes modeling and design of software systems, distributed and pervasive systems in particular, (ii) project management, which includes managing distributed software projects, project organizational issues and their relation to software architecture, and (iii) cultural differences of various regions (in everyday situations, in business, in education). Guest lecturers from global companies such as ABB, IBM, SAAB and Ericsson present their experience in distributed development, the challenges they meet, lessons learned and best practices. Typically 2-3 guest lecturers give presentations each year.

The main focus of the course is project development. Each year the course offers several projects that are supposed to be performed by a group of 6-8 students during one semester (16–18 weeks)[1]. The projects include all phases of software development (requirements specification, design, implementation, integration, delivery and acceptance test), and a systematic project management, including planning, reporting and analysis. The structure of a project that follows the concept of a "globally integrated project" [10] is shown in Figure 1. The students (3-4) from each site form a project team. One of the students acts as a project manager who has the overall responsibility for the project, and of a student on the other site who is a local team leader. One of the teachers acts as the project supervisor (usually a younger researcher). In addition, the project has a project steering group that is constituted from the teachers (typically senior researchers) from both sides. Every project also has a customer who provides

CTGDSD'11, May 23, 2011, Waikiki, Honolulu, HI, USA.

[1] In previous years the project work spanned eight weeks – just a half instead of a full semester. Surprisingly, the results (and the amount of effort) in both cases were similar.

the project team with overall requirements, most often in a form of a short description. The project supervisor can also be the customer, but in many cases the projects have external customers, that are not directly involved in the course – either from companies or from university institutions.

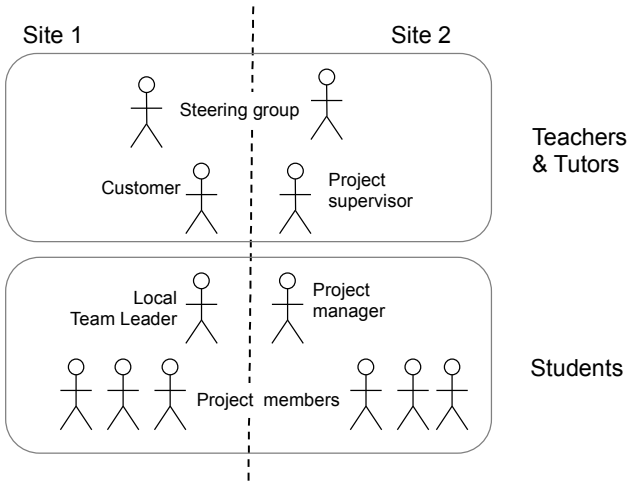

Figure 1. DSD project structure

The projects are performed as follows: the project customer outlines the requirements. The project team analyses the problem, identifies the requirements and negotiates them with the customer. The project team reports the project state to the project steering group periodically (5-6 times during the project) and to all students enrolled in the course. The project members communicate using standard tools such as Skype, chat programs and to some extent the videoconference system. A set of web URLs for each project is maintained on the Content Management System. Team members collaborate using discussion groups and other collaborative tools available on the Web, and use a Subversion repository. In general, the main underlying goal in each project is to remove the distance factor as much as possible.

Different number of students participated in the course each year, ranging from 20 to 65. This required slightly different approaches in project execution, project reporting in particular, since all projects are being presented to all students. The course has shown to be especially attractive for international exchange and visiting students. In the first years of the course, the students were mostly either Swedish or Croatian nationality and we could discuss cultural differences and similarities between these two nationalities. Later many international student both European and non-European participated in the course. A common course setting would typically include students from four to six different nationalities, which provided an additional multicultural dimension in the project performance, while loosing a character of two sites, two nations, and two cultures. Also as a paradox it appeared that the students from Croatia know more about Sweden than the students enrolled in Sweden.

In the last two years the number of participants has significantly grown, due to increased number of international students, but also due to the third site joining the course – University of Paderborn.

Table 1 shows the number of students participating per year, their original countries, as well as the number of project instances.

Table 1. Students enrolled in 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

 # stud. - number of students; # proj. – number of projects

3. CHALLENGES

We have faced a number of problems and challenges in preparation of the common, integrated course with shared activities. Some of them were result of "objective" or permanent differences between the sites, such as the differences in academic calendar, the start date of the academic year or the division into two, three of four semesters or study periods. Certain differences were in admission requirements, and some in examination requirements. While the course organizers (the teachers) were, in general, enthusiastic in building the common course, the administration on both universities approached the problem with more caution, and sometimes skepticism. Our goal was to create a course flexible to the specifics of different universities, and at the same time in compliance with the rules of both universities. This required a great flexibility in building the course and mutual understanding between the sites. Here we list some of the issues we encountered and our solutions.

3.1 Organizational challenges

A number of administrative issues belonged to a category of "conditio sine qua non" to comply to the university rules:

a. *Course approval* –is not an easy task to get a new course approved from the university, especially if it does not fit into the common framework and requires new, and probably not regulated, administrative steps.

b. *Joint student enrolment* – universities have special administrative requests when enrolling extrenal students, such as obligatory personal ID number provided by host country, which was not easy to acquire.

c. *Credits the students obtain in the course* – in the beginning, due to the administrative procedures, the difference in the ECTS credits[2] was huge, 4 ECTS at FER vs. 7.5 ECTS at MDH, with the same workload for students at both sites. This was a big (de)motivational factor during several years.

[2] Standardized by the Bologna System rules [1], 1.5 ECTS credits correspond to one week full-time work

d. *Examination elements* – each institution has its own rules for examinations, such as representation of grade elements in the final grade, grade division based on Gauss curve, even the number of grade choices available (on a scale 1 – 5, where 1 is failing the course, FER can grade students with grades from 2 to 5, while MDH uses only 3, 4 and 5).

e. *Academic calendars* – academic year has a different structure on two sides, from start and end date of the year, to the course length.

f. *Course quality assurance* – in the beginning, one university did not require the official course evaluation, while the other required it, without a defined form. In order to improve our course, the students were evaluating it by filling out the joint anonymous questionnaire. After the Bologna process was introduced, FER started to evaluate the courses based on a local questionnaire, in Croatian, where some of the questions do not relate well to our distributed course.

g. *Administrative course support* – compatibility issues often appeared, due to the differences in the administrative procedures between the universities. One example is different official languages – many forms, web sites, etc. are provided in local languages which required additional efforts for both staff and students

h. *Staff workload* – maintaining a distributed course requires significantly more effort due to various challenges mentioned here. As such a course is an exception to the norm, universities do not regard it as more time-consuming.

While many of the issues occurred only in the initial phase of the course deployment, several of them repeated during the years of the course execution. Like software, the course has to be maintained, not only in its technical part, but also administrative, due to continuous changes of the rules on the local sites. For example, one of the sites changed the start and end period of the semester which influenced the start and end of the course. Another example was introduction of the Bologna process [1] on both universities – although officially the same general process was introduced on both sides, the realization was quite different!

3.2 Distance-related challenges

The fact that the course is carried out in more than one place creates another set of challenges. These are not based on differences between organizations, but on different reactions of people cooperating over geographically significant distances [8]. The challenges observed are:

a. *Motivation and responsibility* – in general, when people do not have some kind of a direct relationship, best-established meeting in-person, face-to-face, it is hard to build up a well-integrated team [2]. People feel less responsible for team members who are far away. In addition, it is easy to classify members between "we" (local members, "the guys we drink coffee together") and "them" (remote members, "we never met"). We have experienced this, particularly in the initial phase of the course, when one site in a team would simply forget about the other site. This is a big challenge in the beginning of the project since students need to quickly start working together, but still do not know each other well.

b. *Communication problem* – as students do not have the opportunity to meet in person, they need to pay special attention to communication over distance. Various decisions need to be made, such as the choice of communication methods (synchronous, asynchronous), means of carrying out remote meetings, ensuring mutual understanding of things being said, rules for meeting participation, etc.

c. *Technology* – technology is the backbone of such a distributed course and can be either a great help, or a source of course troubles. University infrastructure prerequisites, as well as tools for a synchronous virtual classroom, asynchronous content management system, and collaboration and communication platforms need to be carefully chosen and activities planned in advance, as improvisation during the distributed course can be hard.

3.3 Cultural challenges

We have observed the following typical challenges due to the multinational structure of students and staff involved:

a. *Language differences* – almost no students or staff are native English speakers. This brings up issues, both in levels of fluency in English and accents, which vary greatly from Europe to Asia. In our case, when we had students from seven or eight different countries, a problem of understanding a presenter appeared often (in particular when combined with occasional lower sound quality of the teleconference system).

b. *Communication characteristics* – there is a difference in communication habits of various nations involved. People from South Europe tend to be more open and direct in their conversation, easily giving comments and critiques, while people from Asia are more reserved in giving their opinions and evade confrontations as much as possible.

c. *Different type and level of technical knowledge* – students' knowledge and university background in our multicultural team is one of the regularly experienced issues. This was in particular visible when many international students participated. For example students from FER are more "programming-oriented", and Swedish students more collaboration-oriented. The international students have shown a diverse level of technological knowledge.

d. *Timing issues* – the attitude towards one's own and other persons' time differs from North European, through South European, to Asian cultures. As students value the time differently, teamwork problems occur, especially with respect to deadlines or meetings participation.

e. *Agreement* – there is a difference between agreement and commitment to work on a particular task. While European cultures would easily discuss the possibility of doing something successfully and on time, sometimes followed by rejecting or redefining the task or milestones, eastern cultures would agree to do the assigned task, although they would not actually commit to do as agreed. This was caused by the different levels of knowledge and ability to understand the nature and the complexity of the problem, but also by the cultural differences. For example, Asian students exhibit a certain reluctance to reject tasks and say "no" when they actually mean it, which confuses European students who do not expect that "yes" can sometimes also mean "no". On the other side the Asian students would find European colleagues rude when they would directly say "it does not work".

f. *Teamwork* – successful teamwork is of primary importance in a distributed course, but it is additionally exposed to different challenges. During eight years of the course we

have seen quite different understanding of teamwork. The (native) Swedish students have a high sense for teamwork in general and tend to prefer group discussions and common agreements. South-European students, including Croatian, are more goal-oriented and easy take "things in their own hands". Asian students in many cases have deep respect for hierarchy and assume that it is of the highest importance to please the supervisor or the examiner.

g. *Different views of teaching staff* – teaching staff should ideally act like just another distributed team, and make joint, single-minded course decisions together. Depending on the situation, this can be tough, for various reasons: having different expectations of students and their results, not having a good insight in the work of the remote students, or simply different backgrounds, which reflects on the teaching and mentoring approach.

4. LESSONS LEARNED

Based on our experiences in dealing with these course challenges, we discuss our solutions to issues encountered. Considering problems globally, there is a feeling that motivation problems are one of the most important to solve. Most of the problems would not have existed in the first place, by carrying out the course only on one location. Therefore, a sufficient motivation should be found, both for the staff and the students.

4.1 Organizational challenges

DSD course is officially approved through an interuniversity project agreement, to overcome administrative issues.

The solutions to organizational challenges have been different, depending on the type of the problems and abilities to solve them. For some, we simply had to accept incompatibilities and continue working on possible solutions for sever years – examples were different credits at different sites, which we had during the first five years, another was different start dates of the course, with students on the site that started earlier given some internal work, a third example is additional administrative work by the course staff on behalf of the university administration. Over the course's eight years we have come to a state of relative balance and compatibility, not requiring drastic changes, but rather careful evolutionary improvements. Still we have learned to be alert on local changes and analyze potential problematical consequences of these changes.

Quality assurance of the course is currently done on different levels. The most extensive course evaluation is performed in an anonymous questionnaire, filled out by students of both universities after the course completion. The survey consists of ca. 15 questions, addressing various course aspects. Students are also encouraged to leave a comment for each question to helps us evaluate the course in more detail. The evaluation questions are designed with distributed course specifics in mind. Croatian students are also invited to fill in an anonymous questionnaire, provided for all courses by the university. Unfortunately, just this survey is considered relevant for course quality assessment by the university management, although the first specialized one would be more appropriate.

4.2 Distance-related challenges

The distance-related challenges influenced our course process, pedagogical approaches and internal schedules the most. Aware of possible problems, we created a course schedule with regular deadlines (one in ca. 3-4 weeks). At the beginning of the course, the project state is presented more often, as the teams need more guidance for a successful kickoff. It is especially hard for students to start working while they do not know each other at all, so we "force" them to prepare a small joint Project Vision presentation just seven days after they've first met over the Internet. The current state of each team is also monitored through week reports, as well as through the weekly "How do I feel?" pool, which has the students quickly answer (by selecting a value from a scale) two questions: "How happy are you with the current status of your project?" and "How happy are you with your status?". These measures can reveal a heartbeat of a team, and teachers can jump in if problems arise. Teaching staff also gives a lecture about the experiences from the last years' projects, discusses where the problems arose, what kind of technology was most used for communication in each project phase, how to present project status, etc. We also require from the students that they on the course web page upload their picture, name, contact info, and their interest, to help them in getting know each other.

The teaching staff selects the students in the project groups, based on the students' profile. The students namely have to out a lot of information about their technical background. We try to achieve a balance of experience, and nationalities. Students are advised to choose communication tools and methods they feel are best for them. A combination of synchronous and asynchronous tools is mostly used: mailing lists in combination of instant messengers, former in a more sophisticated version like Google Groups, and latter mostly Skype, MSN and Gtalk. Students very soon realize the differences in approaches and suitability of these tools for each occasion. Our questionnaires show that they prefer text-based communication instead of using audio/video calls for meetings. The reasons are mostly related to language: it is hard to understand somebody with a strong accent, and some students need more time to think how to express their thoughts in English, so they rather write than speak.

4.3 Cultural challenges

We have realized that the first problem the students (and to some extend the teaching staff) face is awareness of the cultural differences. We have a lecture about cultural differences and the students have an assignment to compare different cultures. The assignment results are often of low level, typically include known clichés, but at least give the students an opportunity for some reflection, and awareness about possible problems.

The project teams work independently, but have continuous communication with their supervisor, and the supervisor takes measures if potential problems are observed. The measures include individual meetings with "problematic" students, or a common meeting with the group to discuss the problems, with supervisors as mediators and moderators. In most cases the teams would organize the work according to their preferences and knowledge. The groups would usually build a positive spirit and enthusiastically put a lot of effort into their project. This is also achieved do to presentations of the projects in front of the whole class. Recognition of successful presentations and results encourage the teams in further work. One important grading element is the teamwork and the students are aware of it from the start. Over the course's eight years, we had several cases where there have been serious problems in teamwork together with the project results, caused by different factors; low motivation, low expertise of some students, or very egocentric students with difficulties for communication are some examples.

5. SUCCESS STORIES

Around 50 projects were carried out since the beginning of the course in 2003. Most of them were successful, of a good quality, some resulted in research papers published on the conferences, while some continued to live as parts of other applications. Here we briefly describe three types of projects which had an additional component: external customers.

Industrial collaboration. In last two years, some of the projects were done in cooperation with a Croatian enterprise oriented on telecom billing and fraud detection systems. This company mostly employs engineers educated at FER, and is thus familiar with the level of education and experience of our students. The company is also a representative of a business partner in one Tempus project, which is aimed at providing better cooperation between Croatian universities and the industry, from education to research level. The experiences, described in [6], are very positive for all sides: company, teaching staff and students.

Merger scenario. The third university, Universität Paderborn (UPB), Germany, has participated in our DSD course for two years, creating a company merger scenario. Students from UPB are involved in their local software engineering, project-oriented course, which has some similarities to DSD, but differences like duration of the course (full academic year) currently prevent full integration with DSD. Therefore, a scenario of involvement is created, where some of the Croatian and Swedish students would join the ongoing project in Germany, building a product which will be integrated in the full solution. This puts our students in the real-world position of a new-coming team, where they would be faced with the new challenges of a company merger.

SCORE competition. SCORE competition is a software engineering students' competition, where student teams work on a software engineering problem, defined by an external customer from a foreign university. Teams need to communicate with a person whom they have never met, find out what the project is about, and keep them informed and involved. Most of the times, teams also need to find potential users in their own environment, to discuss functional requirements with, and to get help testing the product. Several DSD projects in years 2008 and 2010 were SCORE projects, and our project stakeholders were extended with external SCORE customers. Such work was more complicated, as students had to communicate with the third side, and write additional SCORE reports. However, they were highly motivated by the involvement in the competition, and were very successful[3].

6. CONCLUSION

Distributed software development is "here to stay". Preparing students for such work thus is very important, as there are not many chances to experience it before they start working. Having the students obtain a first-hand distributed experience brings along lots of issues, and requires flexibility and greater effort from students, teaching staff and university administration. Our finding is that there are two the most important factors of the successes: a strong motivation, both from the students and the teaching staff, and flexibility in accepting different rules and habits.

[3] In 2009 all (four teams) our DSD SCORE projects entered the semi-finals, while three teams were invited as the finalists to ICSE 2009, Vancouver, Canada. The overall winner of the competition was a DSD team with the project "BTW: if you go, my advice to you". (http://score.elet.polimi.it/)

7. ACKNOWLEDGMENTS

We would like to thank Prof. Wilhelm Schäfer and his group from Universität Paderborn, Germany, for collaborating with us on this course, and Mr. Branko Beslać from Kapsch TIS company for participating in the course as an external customer. This work is supported in part by the Croatian Ministry of Science, Education and Sport, under the research project "Software Engineering in Ubiquitous Computing" and the project Hi5 funded by Swedish Vinnova funding agency.

8. REFERENCES

[1] The Bologna Process - Towards the European Higher Education Area. http://ec.europa.eu/education/higher-education/doc1290_en.htm.

[2] Bosnić, I., Čavrak, I., Žagar, M., Land, R., and Crnković, I. Customers' Role in Teaching Distributed Software Development. *CSEET '10 Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training*, IEEE CS Conference Publishing Services (2010), 73-80.

[3] Crnković, I., Čavrak, I., Fredriksson, J., Land, R., Žagar, M., and Åkerholm, M. On the teaching of distributed software development. *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, University of Zagreb, University Computing Centre (2003), 237-242.

[4] DSD course, the local site at MDH. http://www.idt.mdh.se/kurser/cd5610/.

[5] DSD course, the official site. http://www.fer.hr/rasip/dsd/.

[6] Durnell Cramton, C. and Hinds, P. Subgroup Dynamics in Internationally Distributed Teams: Ethnocentrism or Cross-National Learning. In B.M. Staw and R.M. Kramer, eds., *Research In Organizational Behavior*. Elsevier Ltd, Oxford, UK, 2005, 231-263.

[7] Gorton, I. and Motwani, S. Issues in Co-Operative Software Engineering Using Globally Distributed Teams. *Information and Software Technology 38*, 10 (1996), 647-655.

[8] Hinds, P.J. and Mortensen, M. Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science 16*, 3 (2005),290-307.

[9] Kraut, R.E., Fussell, S.R., Brennan, S.E., and Siegel, J. Understanding effects of proximity on collaboration: Implications for technologies to support remote collaborative work. In P.J. Hinds and S. Kiesler, eds., *Distributed work*. MIT Press, 2002, 137-162.

[10] Lago, P., Muccini, H., Beus-Dukic, L., Crnkovic, I., and Punnekkat, S. GSEEM: a European master program on global software engineering. *International Journal of Engineering Education 24*, 4 (2008), 747-760.

[11] Meyer Bertrand. Another DOSE of distributed software development. 2010. http://bertrandmeyer.com/category/distributed-software-development/.

[12] Shaw, M. Software Engineering Education: a Roadmap. *ICSE '00 Proceedings of the Conference on The Future of Software Engineering*, ACM Press (2000), 371-380.