Ten Tips to Succeed in Global Software Engineering Education

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

Ivana Bosnić, Mario Žagar
University of Zagreb
Faculty of Electrical Engineering and Computing,
Zagreb, Croatia
{ivana.bosnic,mario.zagar}@fer.hr

***Abstract*—The most effective setting for training in Global Software Engineering is to provide a distributed environment for students. In such an environment, students will meet challenges in recognizing problems first-hand. Teaching in a distributed environment is, however, very demanding, challenging and unpredictable compared to teaching in a local environment. Based on nine years of experience, in this paper we present the most important issues that should be taken into consideration to increase the probability of success in teaching a Global Software Engineering course.**

Keywords - distrubuted software development, education, global software engineering

I. INTRODUCTION

Education in Global Software Engineering (GSE) is even more challenging than education in Software Engineering. In addition to the simulation of sufficiently complex, but still solvable, "real industrial problems", GSE training also requires the appropriate simulation of a distributed environment. The best way to achieve this is to conduct the training in a real distributed environment, with both students and supervisors geographically separated. Experience in GSE shows that successful performance in a GSE context requires specific skills. It takes a lot of time and considerable effort to achieve these skills. Teaching GSE is limited by the training period, which is usually in the form of a course lasting one semester (five months) or less. There is no room for learning from mistakes. In addition, the possibility of something going wrong is much higher than in a local environment. For this reason, a number of precautions should be considered and applied to avoid problems. The goal of this paper is to discuss necessary measures in the form of tips: what you should think of in advance, and how to mitigate the risks. Our ten tips, we believe, are the most important success factors. The list is based on our experience, data collection and performance analysis of the Distributed Software Development course during the last nine years. In GSE the problems encountered have accumulated through experience (for example in [1],[2],[3]): development tools and environments, communication, design knowledge, infrastructural support, coordination, and availability. Similar factors are important for education in GSE, with some modification. In [4], we have presented the important factors in the Distributed Software Development course (DSD), that can be grouped into "objective" (factors based on facts) and "subjective" (factors based on the personal skills and characteristics of participants). The most important objective factors are communication support, collaboration support, and the infrastructure supporting the global environment. The most important subjective factors are the participants' motivation, awareness, social skills and technical skills. In addition, there are important objective factors that we can call "external" since they are related not directly to the course, but to different rules at local sites.

The main question we pose here is: what are the necessary and most important measures that should be taken to ensure that the DSD course is conducted successfully; what should and what should not be done?

The rest of this paper is organized as follows. Section 2 describes the DSD course. Section 3 gives an overview of the methods used for the collection and analysis of data. In section 4 we give the tips and their rationales. Section 5 discusses the tips and relates them to the successes and failures of the course. Section 6 presents related work and finally, Section 7 concludes the paper.

II. DSD COURSE

The DSD course [4],[5],[6],[7] has been conducted successfully for nine years, starting in 2003. The course is designed as a combination of lectures, guest presentations and distributed projects. Both sites perform all elements of the course jointly. The lectures are common and distributed equally between the sites. They are transmitted using the videoconference system. Experienced collaborators, working on similar—distributed—projects in the industry, give guest lectures. The projects are shared and the students work together as if on a local project. The examination elements are common, as are the students' grades. The students are actually enrolled at both institutions and in addition to the credits that they receive at their own university, they get a certificate from the other university as well. Such a fully integrated course, in contrast to loosely integrated courses (such as, for example, the multi-site project course described in [8]), brings a lot of administrative and technical challenges, but also many advantages. Firstly, the above constellation follows trends observed in GSE [1]: While in previous decades the most common form of GSE was outsourcing, or remote execution of particular activity (such as implementation based on a detailed design, or system test), recent trends show involvement of all sites in the entire project lifecycle. There are several reasons for this. One reason is to achieve a balance of skills and a balance of customers' requirements at different sites. The other reason is the improvement of collaboration and development tools.

A fully integrated course has a better-articulated common goal and in general exposes GSE-related issues to their extremes. By designing a DSD course in this way, we have also noticed indirect effects on global integration; both sites learned about best practices and gained insight through lessons learned from each other into course management, university organizations, student contact, etc.

Group project work is the main part of our course, as our focus is on giving students the experience of GSE. All phases of a software lifecycle are included: requirements specification, design, implementation, integration, delivery and acceptance testing. Projects offered to students are scaled to be suitable for 6–8 team members (3–4 per site), and are done over a period of one semester, 16–18 weeks. Projects are in the form of role plays, where one of the students acts as a project manager who has the overall responsibility for the project, and one student at the other site acts as a local team leader. One of the teachers plays the role of the project supervisor (usually a younger researcher), while the steering group consists of other teachers. Project customers can be either senior researchers, or external customers, such as companies or project proponents in various SE contests [5].

In the initial phase, customers give a general project description to their student teams. It is up to students to proactively discuss the project with the customer – from negotiating the whole project scope to identifying detailed requirements, as well as deciding on the technologies and system architecture. In project work, as students are working over a distance, they make use of many collaborative tools, such as instant messengers, mailing lists, discussion groups, version control systems, bug trackers, polling websites, and content management systems. Besides their project web page, which is the official channel for delivering the project news to their steering group, students are free to use any communication methods and tools they find suitable, all in an effort to remove the obstacle of remote communication as much as possible.

In addition to communicating with the project supervisor as needed, throughout the course students present their project status by: (1) sending Weekly Summary Reports every week, and (2) reporting their project status every 3–4 weeks in the form of presentations to customers, supervisors, steering group and other students. These methods give us the ability to monitor their progress, and to see problems early enough to help solve them.

Although the main two course sites are Croatia and Sweden—with a German university acting in a special merger scenario for two years [5]—the number of nationalities involved is far higher. This course is attractive to international exchange and visiting students, as it is held completely in English, and additionally deals with a number of cultures. Table 1 shows the number of students participating per year, their countries of origin, as well as the number of times the project was held.

TABLE I. 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

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

After the initial years when most students were from Croatia and Sweden, recent years have been highly international, especially in Sweden, but also in Croatia, which is now also involved in the Erasmus/Socrates students' exchange. Courses typically include students of six to ten different nationalities, which provides an additional multicultural dimension to the project, while losing the character of two sites, two nations, and two cultures.

III. INFORMATION SOURCES

During the semester we gather several types of course data. We do this for two reasons: a) to adjust the course to the students' needs during the course, and; b) to analyze data after the course in order to improve the performance of the course the following year.

We collect the course data in the following ways.

Initial questionnaire. At the very first lecture, the students are asked to fill-in the initial questionnaire, consisting of two parts: 1) personal information, communication channels (email addresses, instant messenger usernames), professional experience, hobbies, and; 2) grading their experience (with grades 1–5) in various programming languages, development environments, programming platforms, databases, modeling languages and tools, frameworks and collaboration tools.

The purpose of this questionnaire is three-fold: it provides a single place to see students' information at a glance and quickly introduces students to each other at the start of the project; it enables us to have an overview of students' skills, relevant for assigning the students to projects; and it involves students in the course from the very beginning, by giving them a simple task. This questionnaire is done with the help of Google Docs Forms.

Weekly Summary Reports. Team members are supposed to fill in their weekly report and submit it to the project leader who compiles them into a weekly summary report.

This consists of project timeliness, current and forecasted project costs, results achieved in the current week and activities planned for the next week, an action list along with those members responsible, a description of experiences and problems, milestone metrics and work hours for each member.

"Happiness" poll. In order to track some of the students' personal feelings regarding the project, each week the student should fill in a simple two-question poll: rating (1–10) "How happy are you with the current status of your project?" and "How happy are you with YOUR status?" Each student completes the poll without knowing other team members' ratings. This data can help with discovering problems in personal relations inside the group.

Minutes of meeting. As in a professional environment, the students are asked to document each meeting with the minutes document, describing points discussed, actions to be taken and those responsible for those actions.

Final questionnaire. As the course nears its end, the students are required to fill in an exhaustive final questionnaire, which reflects in detail their experiences and thoughts on distributed work, with a separate focus on the differences between work with local and distributed team members. They are asked about topics such as:

- Quality of work coordination
- Level of collaboration of each team member
- Changes in collaboration over time and project phases
- Usefulness of communication tools and the main phases in which they were used
- Team meeting preparations
- Issues in project work and information flow
- Their role in the project's success
- Ability to influence project decisions
- The process of making project decisions
- Project requirements and their changes
- Software development process decisions
- Integration issues
- Motivational and demotivational factors
- Differences in predicted and invested work hours
- Impact of cultural differences
- Regrets

In the last few years, most of these questions have had both quantitative and qualitative answers, so they could be further analyzed.

Course evaluation. Since the first year of the course, we have conducted an internal course evaluation, which is anonymous. Here, the students can help us see possible problems with the course, by discussing topics such as: concept of lectures and projects; cooperation between sites; student workload; project support; and course administration. All the elements are also both graded and described.

The data supporting our tips described in the next section was gathered during a 5-year period (2004–2008.), mainly from the final questionnaires. 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 here show what percentage of those students included in the analysis observed an aspect or issue.

IV. The Tips

Based on nine years of experience, variations in course setup, students' feedback, and observation of students' results, we have identified recurring issues and ways in which to address them. They are given here as ten tips[1].

Tip 1: Start communication by brute force

The most important advice given in the GSE survey [3] is "invest in face-to-face meetings, temporal collocation, and exchange visits." This illustrates well how the initial communication is important. In the first few years that the DSD course was held, we observed that it usually takes quite a long time before the students get up to full speed with the project work. The reason for this is psychological and social, and is deeply rooted in human behavior. Even if, rationally, the students know that they must start to communicate with the other site in order to begin the project, until direct contact is established, the other site remains less important.

We have realized that it is not enough to say to the students that the communication is important and that this is the first thing they should do in the project. They must be forced to communicate by having a concrete assignment with a firm deadline for completion. For example, in the DSD course, as soon as they are assigned to the project, project members must start with intensive meetings in order to complete the first assignments and present them within a week. They must document and present the project plan and vision (the project goal and its outcome). The project plan must contain many concrete details: means of communication, frequency of meetings and how they will be conducted, project members' roles and how their expertise will be used in the project, project development infrastructure, tools and technologies used in the project, breakdown of project activities, expected outcomes, etc. Although many of the specifications and estimates drawn up during these meetings will not be suitable or accurate, the meetings are worthwhile since their most important function is to encourage the students to start communicating from the very beginning of the project.

The students' full engagement will not come spontaneously – it must be instigated by the project supervisors. This implies that the supervisors should have a strong involvement at the beginning of the project, send clear signals to encourage communication, and give warnings in its absence.

Tip 2: Get the students to be familiar with each other as soon as possible

Establishing professional communication is a necessity, but good communication alone is not enough to ensure the success of a distributed project. The students have to build

[1] We use "tips", i.e. less formal advice, since they are valid in general, but their details can differ from case to case.

up loyalty, team spirit and collective responsibility. While individual responsibility is usually the result of one's own professionalism and attitude, characteristics such as loyalty, trust and common responsibility are, to a great extent, the result of group cohesion. The best way to build this up is to make personal contact, where the members become familiar with each other [9],[10]. This is usually not possible in a distributed course. For this reason it is essential that the project members get to know each other as much as possible in an indirect way. In the DSD course, the first thing the students should do is to publish personal information on the project Web page, with their picture, communication channels, professional interests and skills, as well as their interests in general. In addition, we insist that every student participates in the project follow-up presentations during videoconference sessions. In this way the students become visible and familiar to their colleagues.

Tip 3: Keep communication levels consistently high

Starting intensive communication at the beginning of the project is crucial. There is, however, always a risk that communication decreases. For this reason it is important to have continuous measures in place to keep communication at a high level. Examples of these measures are: a) requiring a detailed communication plan to be specified in the project plan; b) identifying communication tools and the means of communication; c) insisting on continuous and detailed reports on the project meetings (frequency, participants, type of communication, decisions taken). In particular, the students should report about potential or existing problems. In some cases, we have experienced misunderstandings and problems with these meetings, due to the language barrier – the students originate from different countries with quite different pronunciation and English skills. This influences the students' choice of communication method, as shown in Figure 1. Text-based tools (instant messengers, email, forums) are preferred over audio and videoconferences. For this reason we insist on having minutes of meetings documented on which every participant agrees. In some cases when problems occurred in oral meetings, the students moved to mostly written communication (instant messengers, email), which, although it limited the communication, decreased the risks of misunderstanding. In other cases, when the supervisors realized that there were communication problems, they would participate in some of the meetings helping in getting the mutual understanding.

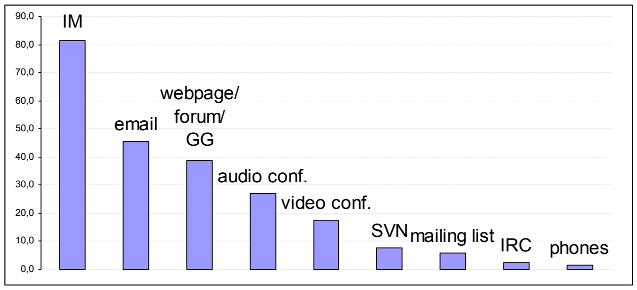

Figure 1. Communications methods usage

Important aspects of the communication are communication patterns. The number of team members (6–8 students) in a local context is conducive to seamless communication on individual and group bases. However, in a distributed environment the distance makes this significantly harder. While local communication usually works well without much effort, distance communication requires explicit and systematic support. Also, achieving direct communication in all directions is not possible; dedicated communication channels should be defined. In the case of the DSD course, project organization is often defined in a way that it explicitly addresses the distance communication, as shown in Figure 2: A continuous communication between the project manager and the team leader, as well as between the project manager and the customer and project supervisor is obligatory for all projects. Dedicated communications are defined within each project and they are established between the project members work together on a common assignment.

Residing at different sites, the project leader and the team leader are compulsory roles in the project. Their main responsibility is the communication between the sites. They are also responsible for communication with the project supervisor and project customer. In addition, some members have communication dedicated to a particular activity. In this way, the communication channels are made explicit, and the individual members get explicit responsibility for a particular communication.

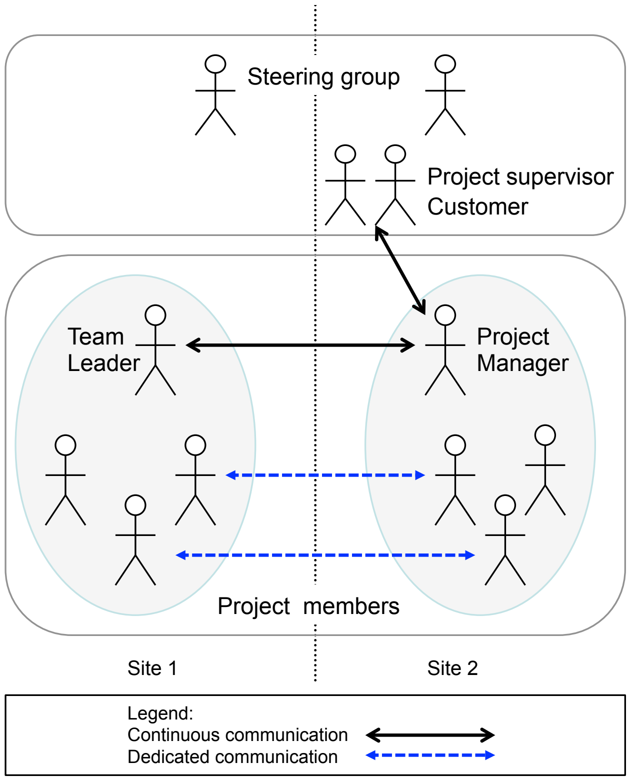

Figure 2. DSD communication channels

Tip 4: Ensure that students keep the other site in mind

Difficult as it is to become aware of the other site, it is very simple to forget them. The challenge of awareness is well known [11] and it has been addressed in several Software Configuration Management (SCM) tools. These tools provide support in managing changes in artifacts (such as source code and documentation), tasks, and in general communication [12]. For this reason using SCM tools is a condition *sine qua non*. For smaller projects, like in the DSD course, simpler SCM tools such as Subversion are quite convenient for source code and documentation management, and with some additional tools, like GoogleDocs, it is possible to manage changes and tasks quite well. Version management and awareness of changes in artifacts has never been a large problem in the DSD course. One of the reasons for this is the strict requirement to use an SCM tool (specifically, Subversion). Another reason is the requirement to have an SCM manager on each project. And the third reason is strict monitoring that the tool is used correctly.

More serious problems have occurred when one site forgot to inform the other site about certain important decision, or made some assumptions without first checking them with the other site. Figure 3 shows that such communication and synchronization issues were the negative aspects most observed by students. To avoid this, we pay special attention that the sites explicitly specify the decisions taken during meetings, and that they jointly share them at the weekly presentations. In some cases we have observed a tendency for students from the local site to have considerably more frequent contact with the supervisor or customer. Since supervisors have several years of supervisory experience, they are aware of this problem and they regularly keep contact with both sites. When a project has an external customer (for example from industry), the distributed meetings can be inconvenient for the customer and the risk that only one site gets their full information increases. In some cases we have observed these problems; the students having less contacts with the customer or with the supervisor lose motivation, especially if an important decision was taken, but other site was not informed about it. In these cases we insist on tight communication between the project manager and the team leader.

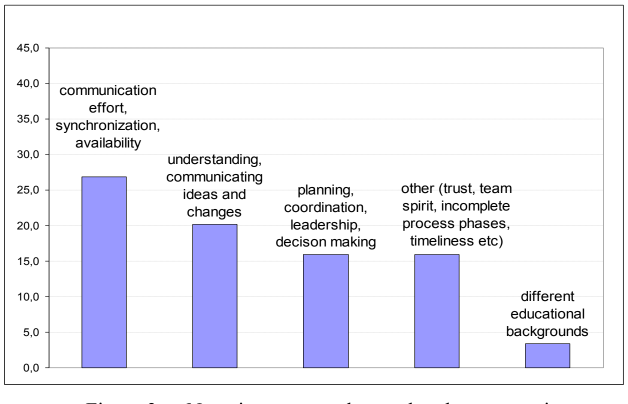

Figure 3. Negative aspects observed at the remote site

Tip 5: Keep the students highly motivated

Motivation is a driving force in education [13], not only in distributed courses. It is a basic prerequisite to achieve any result, and motivation will make the difference between excellent and outstanding performance. Motivation is a complex phenomenon and it depends on many different factors: the individual, collective, cultural and professional environments, age, sex, etc. [13]. What is specific for GSE is that the motivation factors here can differ more among the students than in non-distributed courses. We have observed that students with different backgrounds have different sources of motivation (for example interesting work, nice company, getting the best grades, curiosity, creativity, concrete assignments, direct involvement, encouragement, etc.). Many of these factors depend on the individual, but cultural influences can also be observed by location – and not only by geographical location, but also by organization. At the different universities, students have different habits and expectations. The same is true for the faculty and administration staff. For example, at some universities the timing of studies is important (students are required to pass all courses during the current school year), while at other universities timing requirements are relaxed. At some universities the most important, if not the only, means of assessment is the examination at the end of a course, while at other universities the most important element is performance during project work.

We have observed that there are two aspects to motivation: (i) becoming motivated, and (ii) remaining motivated. Getting students motivated requires similar measures for any type of course, not only distributed – it is important to find out what is attractive to the students. For example, most of the students prefer project-based courses to theoretical ones. Project types are also important: use of modern technologies or devices is often attractive. Projects with important user-interface components are also usually attractive. A smaller, but not negligible, number of students enjoy challenges, and thus like the challenging projects as well. When interviewing students at the beginning of the course, in response to our question, "why did you choose this course?", the most frequent answers (repeated year after year) were: "I like to work in a team"; "I like to work with real projects"; "I have heard from other students that this is a great course"; "I like to learn about students from other countries". While there has been some expression of interest in a distributed course, this was not the dominant factor.

Keeping students' motivated is more challenging in a distributed environment. Since they depend on the students at the other site, whom they do not know well, they might trust them less. If the customer or the supervisor at the other site has limited opportunities for direct communication, this might make the students less motivated, or make them lose motivation more easily. We have realized that for these reasons it is very important to keep the students' motivation high. How to achieve this? In the course questionnaire we asked the question: "What did you like the most in the course"? – a question that can be related to students' motivation. Here are the most frequent answers: "The project work"; "The distributed environment"; "Meeting other cultures"; "New technologies". This shows that, by the end of the course, distribution has become one of the most popular features of the course, in contrast to answers received regarding motivation for enrollment.

Our opinion, based on observation, is that the following factors keep the students motivated:

- Giving the students enough flexibility to develop their creativity (most often we allowed them to choose the technologies, the development processes, and to tailor the requirements);
- Giving the students the opportunity to express themselves through the presentations (although frequent presentations were demanding, being able to show their results to all students from both sites was encouraging);
- The distributed environment and communication using a videoconference system (the ability for students to see each other and to see the classroom at the other site increases the feeling of communication with "real" colleagues and the importance of the presentation);
- Awards and positive competition (students who participated in the competitions doubled their engagement and project work efforts).

Tip 6: Remember: we are different

On our DSD course we have been faced with all sorts of differences. Firstly, differences can be seen in local settings, for example in Sweden, where the DSD course forms part of an international Master's program, there are many international students (mostly from Europe and Asia), and many exchange students (organized through the Erasmus/Socrates exchange office, mostly from EU countries). These students have obtained their Bachelor degrees in their home countries. In Croatia, the local student body is more homogeneous. Most of the students are from Croatia and most of them follow their studies at the same faculty (FER). We have observed that this has implications for the students' performance and behavior, mainly as regards communication between the two sites [14]. Secondly, differences can be seen between the sites. The most indicative of the differences we have observed are described below.

Language differences. The students in Sweden use several styles of English, which are also used locally in conversation between students. Among the Croatian students, the local language is their native Croatian language (with a few exceptions). This diversity sometimes causes problems during conversation; for some students it is difficult to understand students from the other site. There have been some cases of unofficial local comments, written in Croatian, which were not understandable to the other site.

Technical background. The other visible difference has been the different technical background of Swedish students. While Croatian students have had backgrounds as engineers and programmers, Swedish students have a wider spectrum of backgrounds, from project-management orientation, to real-time experts. The level of knowledge of Swedish students is also more varied – from a very basic, borderline acceptable level (even in some cases too low) to a very high level of expertise.

*Openness in communications***.** Croatian students 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 avoid confrontation.

Time. Compared to North-European students, students from Croatia have a more flexible understanding of time. However, in comparison to Asian students, the Croatian students are significantly more punctual.

Commitment. While European cultures are comfortable discussing the possibility of rejecting or redefining a task, but take decisions firmly, Asian students show a certain reluctance to plainly reject tasks and say "no", even that is what they actually mean. That confuses European students, who do not expect "yes" to sometimes mean "no".

Teamwork. The (native) Swedish students have a developed sense of teamwork in general and tend to prefer group discussions and common agreements. South-European students, including Croatians, are more goal-oriented and easily make their own decisions (good or bad). In many cases, Asian students have a deep respect for hierarchy and will wait for decisions from their supervisors. These differences played a negative role in several cases and posed a great challenge. Figure 4 shows which cultural differences the students reported as the most important to them. According to [15] there are two levels of challenges related to differences. The first level, called *tolerance of diversity,* deals with acceptance of differences – it is about understanding that differences exist. In our case, it means that the students should be aware of these differences, and be positive about diversity in general. The second level is *tolerance of difference*, whose characteristic is a tolerance of differences in action, during the interaction of participants with different backgrounds, opinions, etc. This means that our students should be tolerant if one of their colleagues speaks differently, maybe has less knowledge of programming but is good in documenting, or has a different opinion about the group work. Our duty as supervisors is to increase and maintain a high level of tolerance of diversity. At the beginning of the course, we have a lecture about cultural differences and students are given an assignment to compare different cultures. The assignment results are often of low quality, and typically include known clichés, but they do at least give students an opportunity for some reflection, and awareness about possible problems.

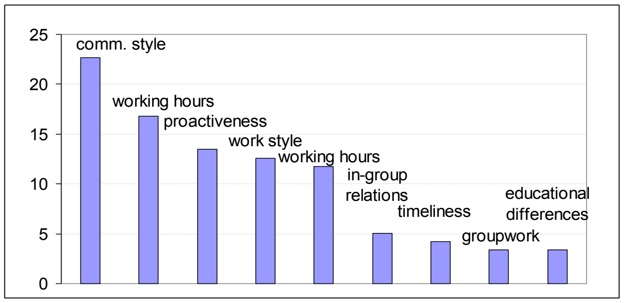

Figure 4. Cultural differences observed

The second level, *tolerance of difference*, is more difficult to teach *ex cathedra*; it is a matter of everyday practice during the project work. For this reason, the supervisors are cautious. As soon as a supervisor observes a sign of a possible problem (not only technical, but related to teamwork, project management, communication, a student's engagement, or similar) she/he reacts by taking some measures. These include individual meetings with "problematic" students, a common meeting with the group to discuss the problem, or proposing some possible solutions such as improving documentation, emphasizing the importance of commitment, and information sharing. In most cases, the teams organize the work according to their preferences and knowledge. The groups usually build a positive spirit and put a lot of effort and enthusiasm into their project. This is also seen in project presentations in front of the whole class. Recognition of successful presentations and results encourage teams in their subsequent work. One important grading criterion is teamwork and students are made aware of this from the beginning of the course.

Tip 7: Be flexible – overcome the differences

This tip is related to the previous one (remember and accept the differences), with a focus on how to overcome differences – to neutralize the possible negative effects and to utilize the positive ones. There are several ways to do this. The first, and most important, is to place students into project groups with care and insight. In the early years of the project, we allowed students to organize themselves into groups, and the results were varied – some groups worked very well, but in some groups conflicts and problems arose. Later we formed the groups based on students' starting questionnaire responses and their preferences. In the most recent project, group formation was also based on a technical skill test and some other factors. For example, we have been careful not to run into a situation where a student may feel isolated (e.g. by putting three students together from one country and a fourth from another country). We have also tried to build groups based on diversity – with students from different countries and with varying levels of estimated knowledge. Another method of utilizing the differences is through the selection of the project technology (depending on the students' experience), and even through the definition of the project requirements and goals. In some cases, the group of students may be less experienced in design and programming, so we lower the project goals. Also, in some projects, where the students achieved excellent results in reaching the project goals, we raised the goals by requiring improvements in the project. Our end goal is not only to achieve perfect results, but also to maximize the amount of knowledge the students can gain.

Tip 8: Be flexible – beat the administration

Even in a local context, there is often a conflict between the creativity of faculty staff and administrative rules. In a distributed environment we face at least two sets of rules – those valid at the two project sites – and these rules are often rigid. The more the sites are integrated, the more problems arise. Despite this, for many other reasons, we have aimed for a tightly integrated course. The combination of two inflexible sets of rules brings unsolvable situations and a lot of headaches due to inefficiency in many procedures. The examples are numerous: different start date and length of teaching periods; different types of course enrolment; different grading systems; different credits awarded for a course; different course evaluation; different course schedule principles; administration support existing only in local languages; different rules for running the course; different holidays; different specifications for prerequisite knowledge – these are just a few examples of the problems we have met. Even worse, such constraints come from administrative staff who often lack common course, and consequently have no understanding of course needs. Teaching staff, meanwhile, do not usually have influence on the decisions that cause these problems. What, then, should be done? The answer is the absolute flexibility and creativity of the teaching staff in finding solutions, and a full understanding of the constraints faced by the other site. Continuous adjustment to administrative rules is a key factor for the success of the course.

Tip 9: Be alert

One of the most important lessons we have learned is that "nothing is default" and that you cannot take anything for granted. With two different systems, students from different backgrounds, limited communication, possible technical problems, and many other events that can jeopardize the course work, we have learned that new problems can arise at any time, and that we need to be ready for unexpected new requirements. This implies that each step in the course must be re-examined before it is taken, and that a continuous risk analysis and mitigation should be performed. For example, well before the course starts we must check that the same rules for enrolment are still valid at both sites; whether we can start the course as we did the previous year, or if there have been changes in communication equipment since the last lecture. This also implies that we must be ready to react very fast to any indication of an unexpected change. This requires the attitude of a researcher with a systematic approach to problem solving[2], which includes continuous risk analysis and risk mitigation.

Tip 10: Be enthusiastic

Due to the various challenges mentioned above, maintaining a distributed course requires a significant amount of effort. This, however, does not mean that the local authority will recognize the additional effort involved. For this reason teaching staff must be enthusiastic above and beyond the standard level. Without enthusiasm, due to the continuous new challenges, the course would eventually fade away. What is the payoff, then? The payoff is in giving students a (unique) opportunity to gain knowledge they will very likely

[2] Indiana Jones: "Nothing shocks me. I'm a scientist."

need in their future professional life. The payoff is also in the enthusiasm of students during their involvement in the project, in their increasing interest in communication with other, unknown, people, and, finally, in their success.

V. The DSD Course Specifics

The tips proposed above are based on our experience and on the analysis of students' answers in questionnaires and reflective reports. Table 2 shows the main issues students have referred to in their answers, and which tips each issue is related to. It also shows what percentage of students referred to a particular issue.

The table above shows that most of the issues are directly related to one or more tips. However, there are some issues (leadership, planning and coordination, goal awareness) that are not directly related to the tips, but specifically to project management. Since project management is as important in a local setting as it is in distributed projects, these issues are not specific to distributed courses, unless they are related to communication and awareness issues. Low percentage of students did refer to issues related to tips 8 and 9 ("beat the administration" and "be alert"). Those tips are mostly the teaching staff's and the students are not aware of them.

Since the beginning of the course in 2003, 51 projects have been carried out. Most of them were successful and of a good quality. Several results were used and shown in research papers published at conferences, while several others continued to live as part of other applications.

Several DSD projects participated in the SCORE (Student Contest on Software Engineering) competition in two occasions, International Conference on Software Engineering, ICSE 2009 and 2011 [16],[17]. For ICSE 2009, DSD submitted four projects and in 2011, five projects. In 2009 three of our projects were among the six finalists, and one won the first prize. In 2011 we had two projects in the final five. This is a very strong indicator of not only the success of a particular project, but of the success of the course as a whole.

TABLE II. IMPORTANT ISSUES IN GROUP WORK – STUDENTS' VIEW

Issue	%	Related tips
Communication	55	Tip 1, 2, 3
Task assignment/work distribution	37	Tip 4
Responsibility/trust	27	Tip 4, 6
Punctuality – respecting schedules and meetings	22	Tip 4, 6
Team spirit	21	Tip 5, 10
Cooperativeness	20	Tip 5, 7, 10
Leadership	20	(Tip 1, 3)
Planning and coordination	17	(Tip 1, 3)
Honesty and openness	15	Tip 6, 7
Tolerance	14	Tip 6, 7
Goal awareness	14	(Tip 1, 3, 4)
Awareness of individual tasks/roles	10	Tip 4
Proactiveness/attitude	10	Tip 5, 6

Another good indicator is the cooperation with industry. In the last three years, three projects were done in cooperation with a Croatian enterprise oriented in telecom billing and fraud detection systems. The experience [5] was very positive for all sides: company, teaching staff and students.

Although all projects have been approved, some have also experienced problems. In a very few cases individual students have not passed the course, as their performance was not satisfactory. The problems that less successful projects have faced have been the following:

Technical knowledge gap. In about 10 projects there were problems due to the technical knowledge of some students being too low. This was a consequence of enrolling students with lower skills, without the opportunity to reject them at the beginning of the course, due to administrative rules and the fact that they were new students coming from other universities. In a few projects this was a serious problem. Projects that managed to deliver a good result solved this problem by better work distribution, thus giving the weak students more time to learn. The projects that were close to failing did not manage this, and we had two students who left the project before its completion.

Personal attitude. In two cases we had students who were not sufficiently motivated or had no intention of contributing to the project, and even obstructed the project work. The intention was to remove them from the project, but after having personal discussions with them and introducing additional controlling measures, they usually improved and were able to participate in the project. A problem specific to a distributed project is the difficulty in identifying such students since a lot of time can be spent during discussions between the sites.

Unrealistic goals. There was a case where the initial project plans were very ambitious, and the students were confident that they would implement them. Almost at the halfway point of the project, the supervisor realized, and the students agreed that, with the given competence and the amount of work that could realistically be achieved, the planned goals would not be attained. They managed to finish the project, but with considerably weaker results than expected. In this case the resulting problems were caused by poor communication between the sites and low awareness of what the other site was doing, as well as misunderstanding between the sites.

Combinations of different factors. In a few cases we have had students with a mixture of ignorance and bad attitude, combined with weak project management. One project in particular was problematic: several students had low technical knowledge, one student obstructed cooperation and the manager did not understand the importance of teamwork. After a while the sites did not communicate properly and one site tried to do all the tasks, ignoring what the other site was doing. The problems continued until the most problematic student left the project (when faced with all the problems, and the requirement to change his attitude and produce concrete results).

In one case, one site had direct contact with an external customer, and it unfortunately forgot to include the other site in communications with the customer and in the requirements management process. The project went well, but the students from one site were bitter and felt like second-class citizens. We recognized this problem only at the end of the project.

We often succeeded in improving problematic projects on time, due to tip 9 ("be alert"), and by applying the principles found in other tips.

Fortunately, there have been few problematic projects; the majority of projects did substantial work, and a few, to our great satisfaction (good for Tip 10), achieved outstanding results.

VI. Related Work

There are several courses focusing on global development which are facing similar issues.

Meyer and Piccioni [18] describe their experiences with the DOSE course, which is now held in collaboration with eleven universities [19]. The students work on projects following the specific method and design by contract approach. The projects are initially divided into sub-components, so 2–3 teams do each sub-component. Based on their experience, the authors describe the challenges that have been faced in interface specification, project management and communication, as well as fluctuations in number of students participating in the project from different sites.

Bruegge et al. discuss three years of their transatlantic course (two project sites and a customer at the third site), where they experimented with two different overlapping schedules – with and without overlapping of project phases at different project sites [20]. Lessons learned included problems with collaboration over time zones and phase overlapping; the usefulness of the temporary collocation of both client and student representatives; and how videoconferencing as a form of communication fell short of expectations.

Gotel et al. [21] describe a global supply chain scenario with distributed sub-contract students, as well as students playing the role of clients. Their recommendations include strong social bonding; taking care not to alienate students in the sub-contractor role; ensuring equal participation of all sites; ensuring that staff are not taking over the project manager role; keeping transparency and regular communication.

Lessons learned from six years of teaching a distributed course are presented by Gloor et al. in [22]. They included students from a non-technical background, such as art and design, or business management. They experimented with the concept of a "virtual mirror", to show each participant his communication behavior. Their experiences emphasize balancing the number of students in cross-site teams, making a commitment to the team, having a good information flow, meeting strict agendas and timelines, building global trust, as well as understanding different cultures.

To our knowledge, these courses are more loosely coupled; each country has some freedom in organization, separate grading, even the project goals can be different. Our approach poses additional organizational issues, because it is a fully collaborative course, with the same schedule, grading, ECTS points, etc. An interesting analysis would be whether integration level has an impact on a course's success.

VII. Conclusion

The ten tips we proposed address issues that can be categorized into the following groups:

- *Communication and awareness*
  - Tip 1: Start communication by brute force;
  - Tip 2: Get the students to be familiar with each other as soon as possible;
  - Tip 3: Keep communication levels consistently high;
  - Tip 4: Ensure that students keep the other site in mind;
- *Issues of diversity and difference*
  - Tip 6: Remember: we are different;
  - Tip 7: Be flexible – overcome the differences;
- *Motivation and socio-psychological issues*
  - Tip 5: Keep the students highly motivated;
  - Tip 10: Be enthusiastic;
- *Practical issues*
  - Tip 8: Be flexible – beat the administration:
  - Tip 9: Be alert.

Many of the tips are specific to GSE. Others do exist in local settings too, but in a global setting require special treatment and considerably more effort. There are many other important considerations to successfully run such a course or project. Some examples are: project management, project organization, and development processes. However, these factors are valid in general for any project and project-oriented course. In this article we tried to emphasize the most important issues necessary for running a successful project-oriented course in a global environment.

Acknowledgment

This work is supported by the Croatian Ministry of Science, Education and Sport, under the research project ZP0361965 "Software Engineering in Ubiquitous Computing" and the project Hi5 funded by Swedish Vinnova funding agency.

References

[1] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, vol. 15, no. 1, pp. 91-118, 2009.

[2] S. Komi-Sirviö and M. Tihinen, "Lessons learned by participants of distributed software development," *Knowledge and Process Management*, vol. 12, no. 2, pp. 108-122, 2005.

[3] C. B. Šmite D A Wohlin, "A Whisper of Evidence in Global Software Engineering," *IEEE Software*, vol. 28, no. 4, pp. 15-18, 2011.

[4] I. Bosnić, I. Čavrak, M. Orlić, M. Žagar, and I. Crnković, "Avoiding scylla and charybdis in distributed software development course," in *Proceeding of the 2011 community building workshop on*

Collaborative teaching of globally distributed software development, 2011, pp. 26–30.

[5] I. Bosnić, I. Čavrak, M. Žagar, R. Land, and I. Crnković, “Customers' Role in Teaching Distributed Software Development,” in *CSEET '10 Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training*, 2010, pp. 73-80.

[6] I. Crnković, I. Čavrak, J. Fredriksson, R. Land, M. Žagar, and M. Åkerholm, “On the teaching of distributed software development,” in *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, 2003, Informatio., pp. 237-242.

[7] “DSD course, the official site.” [Online]. Available: http://www.fer.hr/rasip/dsd/. [Accessed: 25-Jan-2011].

[8] P. Lago, H. Muccini, L. Beus-Dukic, I. Crnkovic, and S. Punnekkat, “GSEEM: a European master program on global software engineering,” *International Journal of Engineering Education*, vol. 24, no. 4, pp. 747-760, 2008.

[9] C. Ebert and P. De Neve, “Surviving global software development,” *IEEE Software*, vol. 18, no. 2, pp. 62-69, 2001.

[10] N. B. Moe and D. Smite, “Understanding Lacking Trust in Global Software Teams: A Multi-case Study,” *Software Process: Improvement and Practice*, vol. 4589, no. 3, pp. 20-34, 2007.

[11] A. Sarma, D. Redmiles, and A. Van Der Hoek, “Empirical evidence of the benefits of workspace awareness in software configuration management,” *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering SIGSOFT 08FSE16*, p. 113, 2008.

[12] A. Sarma, D. Redmiles, and A. V. D. Hoek, “Categorizing the Spectrum of Coordination Technology,” *Computer*, vol. 43, no. 6, pp. 61-67, 2010.

[13] D. H. Schunk and P. R. Pintrich, *Motivation in education: Theory, research, and applications*, 3rd ed. Prentice Hall, 2007, p. 448.

[14] I. Bosnić, I. Čavrak, M. Orlić, M. Žagar, and I. Crnković, “Student Motivation in Distributed Software Development Projects,” in *Proceedings of Collaborative Teaching of Globally Distributed Software Development: Community Building Workshop (CTGDSD 2011)*, 2011, pp. 31-35.

[15] E. Langman, “Rethinking the place of tolerance in -education - Encountering otherness between acceptance and rejection,” *Nordic Studies in Education*, no. 2, pp. 96-103, 2011.

[16] “SCORE 2009 contest.” [Online]. Available: http://score.elet.polimi.it/. [Accessed: 26-Oct-2011].

[17] “SCORE 2011 contest.” [Online]. Available: http://score-contest.org/2011/. [Accessed: 26-Oct-2011].

[18] B. Meyer and M. Piccioni, “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*, pp. 3-16, 2008.

[19] M. Nordio et al., “Teaching software engineering using globally distributed projects: the DOSE course,” in *Proceeding of the 2011 community building workshop on Collaborative teaching of globally distributed software development*, 2011, pp. 36–40.

[20] B. Bruegge, A. H. Dutoit, R. Kobylinski, and G. Teubner, “Transatlantic project courses in a university environment,” *Proceedings Seventh Asia-Pacific Software Engeering Conference. APSEC 2000*, pp. 30-37, 2000.

[21] O. Gotel, V. Kulkarni, L. C. Neak, C. Scharff, and S. Seng, “Introducing Global Supply Chains into Software Engineering Education,” *Software Engineering Approaches for Offshore and Outsourced Development*, no. 2006, pp. 44-58, 2007.

[22] P. Gloor, M. Paasivaara, C. Lassenius, D. Schoder, K. Fischbach, and C. Miller, “Teaching a global project course: experiences and lessons learned,” in *Proceedings of Collaborative Teaching of Globally Distributed Software Development Community Building Workshop CTGDSD 2011*, 2011, pp. 1-5.