Managing Diversity in Distributed Software Development Education—A Longitudinal Case Study

IVANA BOSNIĆ, University of Zagreb, Croatia
FEDERICO CICCOZZI, Mälardalen University, Sweden
IVICA CRNKOVIĆ, Chalmers University of Technology, Sweden
IGOR ČAVRAK, University of Zagreb, Croatia
ELISABETTA DI NITTO and RAFFAELA MIRANDOLA, Politecnico di Milano, Italy
MARIO ŽAGAR, University of Zagreb, Croatia

Teaching Distributed Software Development with real distributed settings is a challenging and rewarding task. Distributed courses are idiosyncratically more challenging than standard local courses. We have experienced this during our distributed course, which has been run for 14 consecutive years. In this article, we present and analyze the emerging diversities specific to distributed project-based courses. We base our arguments on our experience, and we exploit a three-layered distributed course model, which we use to analyze several course elements throughout the 14-years lifetime of our distributed project-based course. In particular, we focus on the changes that the course underwent throughout the years, combining findings obtained from the analyzed data with our own teaching perceptions. Additionally, we propose insights on how to manage the various diversity aspects.

CCS Concepts: • **Social and professional topics** → **Software engineering education**; • **Applied computing** → **Collaborative learning**; • **Software and its engineering** → *Software development methods*;

Additional Key Words and Phrases: Distributed software development, global software engineering, software engineering education, longitudinal case study

1 INTRODUCTION

A well-known and often emphasized fact about teaching Distributed Software Development courses in a distributed environment is that the challenges and related efforts are significantly higher than for standard local courses. Our experience from 14 years of running such a course strongly confirms it and, going even further, shows that these challenges do not cease over time

Authors' addresses: I. Bosnić, I. Čavrak, and M. Žagar, University of Zagreb, Zagreb, Croatia; emails: {ivana.bosnic, igor.cavrak, mario.zagar}@fer.hr; F. Ciccozzi, Mälardalen University, Västaras, Sweden; email: federico.ciccozzi@mdh.se; I. Crnković, Chalmers University of Technology, Gothenburg, Sweden; email: ivica.crnkovic@mdh.se; E. Di Nitto and R. Mirandola, Politecnico di Milano, Milano, Italy; emails: {elisabetta.dinitto, raffaela.mirandola}@polimi.it.

and continuously impact students and teaching staff. From our experience, the key factor to maintaining a long-lasting distributed software development course boils down to the ability to manage diversity encountered at multiple layers and among different involved roles. Various aspects of this diversity stem from the three-layered nature of such a course: *institutional*, *teaching*, and *student project layers*.

At the institutional layer, aspects related to different rules and customs influence the setup and long-term maintainability of the course. At the teaching layer, the influence of teachers and their way of teaching as well as different pedagogical traditions in the local contexts, can result in a significant performance variation among the project teams. The two most prominent sources of diversity at the student project layer are students and project customers. Student diversity is predominantly rooted in their educational and cultural background, but different local organizational aspects tend to introduce additional diversity and obstacles in their distributed project work. Differences between involved customer types are mirrored during the project work phase, where a profound impact on product, process, and communication aspects can be observed. To meet the course objectives and intended learning outcomes, it is necessary to carefully manage the magnitude of problems that the students need to overcome, both individually and project-wise.

The objective of this article is to present and analyze the emerging diversities occurring at the three layers and to discuss how these diversities can be dealt with. While some findings from the course addressing particular research questions (such as customer types [4], development processes [5], student motivation [3], risk factors [6]) have been published before, in this article we explore the course elements as extrinsic and intrinsic sources of diversity. We present the analysis results of qualitative and quantitative diversity-related data collected from the course in a 14-year period. In particular, we analyze the changes that the course underwent throughout the years due to clearer course goals, improved methods, and other factors. We combine the findings obtained from the analysis with our own teaching experience. Finally, we propose insights on how to manage the various diversity aspects.

Our findings are based on a considerable amount of data collected over the years in different organizational settings and with different teaching teams, students and customers, and compared with related literature, such as Reference [9]. For this reason, even if we refer to a specific experience, we are confident that our insights can be generalized to similar contexts and be useful to those embarking in similar endeavors.

The remainder of the article is organized as follows. Section 2 gives a short overview of the course and Section 3 presents the research method with the data sources used for the analysis. Sections 4, 5, and 6 discuss the challenges and lessons learned in relation to each of the three layers (institutional, teaching, and project). Section 7 shortly refers to related work, and Section 8 concludes the article.

2 BACKGROUND AND RESEARCH METHOD

The Distributed Software Development (DSD) course is a joint project-based course held by Mälardalen University (MDH) in Sweden, University of Zagreb (FER) in Croatia, and Politecnico di Milano (POLIMI) in Italy. The course has been continuously running and improving since 2003, taking its current format in 2012. All three universities offer this elective course to Master students in their standard curricula. The number of students enrolled each year varied from 15 to 64. The student population attending the course has been very diverse—so far we have had 482 students coming from 45 countries and 6 continents, bringing in different cultural and knowledge backgrounds (see Table 1). These students have worked on 73 projects, 64 of which have been developed by distributed teams. The teaching staff is also diverse. To date, 17 different teachers have

Table 1. Course Statistics 2003–2016

	Year													
Category	*'03*	*'04*	*'05*	*'06*	*'07*	*'08*	*'09*	*'10*	*'11*	*'12*	*'13*	*'14*	*'15*	*'16*
Students #	26	19	41	30	15	38	64	57	36	49	22	26	21	38
Nations #	3	6	10	8	6	8	13	15	7	14	11	14	12	13
Projects #	5	3	6	4	2	6	10	9	5	7	3	4	3	6

been involved, coming from 7 countries, ranging from young PhD students (often former DSD students) to full professors. The course has been quite successful—it produced the winning team of the Student Contest on Software Engineering (SCORE) at ICSE 2009, in addition to two more teams who participated to the final and six to the semifinal; it provided two teams for the final and five teams for the semifinal of SCORE at ICSE 2011; it produced the winning team again at ICSE 2013, plus two teams for the semifinal, and, finally, two teams for the semifinal at ICSE 2015.

The course started in 2003 with two partners only (FER and MDH) and dealt with several organizational and technical challenges, described in more detail in the next sections. In 2009 and 2010 an additional partner, University of Paderborn (Germany) participated in a "merger scenario" in one of the biggest projects that included students from all three sites. In 2012 POLIMI joined the course as the third partner. Since 2014, the software development process has been changed from a classical iterative approach to agile, using the SCRUM framework, to let students experience a less structured development approach.

The course is project-based but includes a number of common lectures on development processes, as well as on relevant industry expertise and experiences in distributed settings. Students, organized in distributed project teams, carry out software projects applying well-established software engineering practices. Project development follows a set of fixed deadlines, starting from project description and requirements analysis, ending with the release of a final and tested prototype.

To minimize the possible unbalance among project teams due to different levels of knowledge in students, we take a set of countermeasures. First, we carry out an extra thorough course enrollment process, meticulously checking students' transcripts of records. Once the course starts, we run an individual written test to further assess students' knowledge on the basics of software development needed to carry out projects. This, in combination with self-assessment polls filled in by students and transcripts of records with grades documenting successfully passed courses, is used to assess students' knowledge and thereby build balanced project teams. Nevertheless, it may happen that certain unbalance is hard to identify prior to the actual project work start. We continuously look for improvements in this assessment process striving for project teams to be as balanced as possible.

The course aims at offering a "real-life" experience; project proposals are offered and owned by real customers from industry, academia, and the SCORE competition. Projects are run by distributed teams including students from two of the three institutions involved in the course. Students face several challenges, such as dealing with cultural differences, organizing distributed collaboration, presenting in a distributed setting, transferring knowledge to remote colleagues, developing self-assessment abilities, and learning to peer-review.

Throughout the course, each project team is accompanied by two members of the teaching staff as supervisors, one at each location involved in the project. Their task is to support the team, to meet it regularly, and help out with the challenges it encounters, especially in non-technical issues, such as organizational problems, intra-team communication issues, and so on. They are

Table 2. Data Sources

Data source	Description
Project polls	Students' preferences on which project they would like to work. The students have the possibility to choose projects from a project pool. Each year 5–8 different projects are performed.
Self-evaluation form	A survey on tools and technologies, to be filled out by the students at the beginning of the course.
Summary week reports	The reports that provide an overview of activities and working hours spent in each project week.
Project management tools	SVN/Git statistics, SCRUM sprint activities, and so on
Project evaluation table	Evaluations filled out by the examiners after the final project submission, covering 40+ elements to be evaluated, grouped into project-related, product-related, documentation and presentation elements
Final questionnaires	A document filled out by each student at the end of the course, in which students can critically assess the events of their project, levels of communication, project stages, cultural issues, lessons learned, and so on, in retrospective
Final grades	The grades given by the teaching staff to each student, based on the points given to them by their peers, as well as the quality of their final questionnaire and supervisors' insight into their project work.

also in charge of the project grading, which consists of several phases. In the first phase, the whole project is evaluated based on around 40 different criteria concerning the followed process, the final product, documentation, and presentations. The evaluation table with the obtained points is forwarded to the project team. Team members discuss internally on how to distribute the points among themselves, based on their specific contribution, and deliver the final points distribution to the supervisors. Finally, supervisors supplement the point distribution with their own insights, with the help of a final questionnaire, to which every student needs to answer.

In all stages of the course, we emphasize very "tight coupling" of teaching staff at all locations, unlike most other similar distributed courses. In our course, all lectures and student presentations take place synchronously through video-conferencing and are attended jointly by the whole teaching team and by all students. Project supervision is a joint activity as final grading is, where the grading of each student is decided by the whole teaching team and then mapped to the specific grading system used at the student's home university.

3 RESEARCH METHOD

The character of this research is a combination of action research (where researchers participate in the project but also have a research goal that requires research-related activities in addition to the participation in the project), experience report, and a longitudinal case study. We analyze a case (the course) during its existence of 14 years. During this period, we systematically collected extensive data from the course, including the "pure" facts such as the number of students, their nationalities, type of projects, the success rate of the course, and so on, and data based on questionnaires, surveys, reporting, and interviews.

These data sources, summarized in Table 2, together with the general experiences and views of each partner university represent the basis of this article. The research is based on analysis of collected data,[1] a number of internal workshops that took place before and after the course, analysis of the passed courses, and plans for the improvements.

We started DSD as a teaching endeavor rather than as an empirical research effort. The main objective of collecting data was to gather evidence of the effectiveness of our work as instructors, and on the relationship between the innovations we introduced in the course and students' performance. In particular, we wanted to understand whether running a completely coordinated distributed course was effective for students to achieve the following learning outcomes:

- Get an understanding of the implications of distributed development;
- Be able to cope with the main issues of distributed development;
- Learn how to coordinate within a group of people with different background, needs, goals, and schedule;
- Improve the project planning and estimation skills;
- Improve the design and implementation abilities of each student in the course;
- Improve the ability of students to defend their theses in front of other students and instructors.

Only when the course was well-established and had produced a significant number of well-trained scholars, we realized the importance of distilling and sharing our experience with others. Our main take-away message, which is also the main focus of this article, concerns the difficulties introduced by the many levels of diversity in the course. We claim that issues in a project course like ours should be considered at least at three different levels—institutional, teaching, and project levels. Therefore, the main research questions we pose are the following:

RQ1 What are the challenges that arise from diversity at the institutional level?
RQ2 What are the challenges that arise from diversity at the teaching level?
RQ3 What are the challenges that arise from diversity at the project level?

The quantitative data that we collected concerns mainly the project level and support the analysis to answer RQ3, while RQ1-2 are answered mainly based on the experience of the authors of this article and on the problems we have faced in the 14 years of the course. During data collection, as the main objective of the course was to train students rather than conducting an empirical study, we needed to make sure not to add overhead on the students to collect relevant data from them.

Threats to Validity. To address possible threats to *internal* validity, we were very careful in avoiding any perturbation to our observations due to subjective researcher bias. To this end, in the interaction with students, we always had part of the teaching team playing an active role in driving them and another part observing the progress without interfering with the project and learning activities. Moreover, we were aware that questionnaires and forms could represent a threat to the validity of our data as they may reflect subjective issues and problems of students that are not necessarily related to the course. To minimize researcher bias and misunderstandings, we always cross-evaluated them; all teaching staff members were involved in this activity. In case of unclear answers, the questionnaire was sent back to students for clarifications. Moreover, we focus on the essence of participants' experience, as shown in phenomenology science [8]: when the essence of participants' experience is similar, then we consider it as representative of social reality. To minimize the risk for response bias to questionnaires, when we received diverging reply to a question as compared to the majority of members of the same project team, we sent them back to students

[1]Data can be found at http://bit.ly/DSD_data.

for clarification and more detailed motivation of the divergent replies, while at the same time discussed among the teaching staff whether there might be any evident reason for it.

Regarding *external* validity, in spite of a long period and rich data, the case study does not cover all possible contexts—for example, all institutions belong to the same time-zone and continent, where, in spite of many differences, there are many similarities, both legal and institutional. If we included a partner outside Europe, then some elements of the course, such as shared lectures and presentations, would have needed to take a different form. Similarly, the course was designed for Master students with at least a basic knowledge of programming. Applying the approach in this course at a Bachelor level could require considerable adjustments. Still, since exchange students are often interested in the course, and since their educational background can vary a lot, the enrollment process is actually based on the student's history in terms of a transcript of records rather than on the educational level. This has led to the DSD course enrolling bachelor students in their final year several times. Concerning population validity, the fact that we have enrolled and studied students from all continents and a total of 45 countries makes us fairly confident on the generalizability of our results in terms of population.

4 INSTITUTIONAL LAYER

Most universities provide support to teachers for (1) facilitating their work, for example, by providing resources, advertising, and enrollment procedures, (2) ensuring the quality of the education in general and for the courses in particular, for instance by providing standardized course evaluation, and (3) ensuring fulfillment of all legal aspects, such as student eligibility, grading system, conformance to the rules given by the university and by the government. Besides that, the institutional layer includes a number of unwritten or unspoken rules that "are known" locally and expected to be followed, even if there might not be support for them (e.g., communication between students, student organizations and teachers, coordination between different courses). The most challenging part in starting a joint distributed education initiative is that there are very few "defaults" that can be directly used; many new elements must be defined. This means that distributed courses get less support from the involved universities and must define new ad hoc quality measures themselves. Moreover, some legal aspects are often not clear either. Some of these new elements are locally related to the activities needed to integrate a distributed course into the local education context, while other elements are related to the synchronization between the distributed sites. Introducing these new elements locally can be very challenging, but at the same time it discloses the opportunity to introduce new practices that can be beneficial for the existing local education context.

The challenge of introducing new elements in the institutional layer is threefold. First, institutional support is usually good in maintaining established practices but can be very reluctant to add new or change existing ones. The practices established for a distributed course can easily be overrun by the new or updated institutional rules by mistake, as a consequence of institutional insensitivity to the specific rules concerning the distributed course. Moreover, different local cultures and assumptions related to them can clash.

The second type of challenge is related to incompatibility of practices among the different sites; part of it is solvable in theory but very difficult to solve in practice, and part of it is not solvable at all. Typical examples are different grading systems and the varied number of credits given at the different sites for the same distributed course. Another example is the study-period start and end dates, as well as study-period length (e.g., MDH has 10-weeks long study-sub-periods, so the course spans over two adjacent study-sub-periods), and course enrollment deadlines, which can be different at the different sites. The presence of midterm exams in different periods across the institutions affects the availability of students too, as discussed in the next section.

Table 3. Differences in Institutional Layer

	Institution		
Category	*FER*	*MDH*	*POLIMI*
Course type	Elective	Elective	Elective
Minimum students #	6	none	none
ECTS credits	8	7.5	5
Grading system	5-point scale (5-1)	3-point scale (5-3)	30-point scale
Study-periods #	2	3	2
Study-period length	16 weeks	20 weeks (+ 11 weeks)	16 weeks
Study-period start	Begin. of October	End of August	Begin. of October[2]
Study-period end	End of January	Mid of January	End of January
Midterm exams	w3-4 Nov	w1 Nov & w1-2 Jan	w4 Nov
Enrollment deadline	Begin. of September	Begin. of April	Begin. of October
Estimation students #	June	April	After course start
Final students #	Before course start	After course start	After course start
Course drop-out	Not allowed	Allowed	Allowed

Specifically to the enrollment process, differences at the various institutions can lead to issues in the common institutional layer. In our case for instance, while FER and MDH can estimate the number of students enrolling for the course several months before the course start thanks to unofficial preliminary interviews (FER) and early enrollment deadline (MDH), POLIMI cannot. The impossibility to make an educated estimation of the number of students leads to the need of a backup plan in terms of extra project proposals in case of a higher number of students than expected. Moreover, specific institutional constraints, such as a minimum amount of students for the course to be given (e.g., 6 at FER in our case) at one or more institutions can potentially jeopardize a course instance for all involved institutions. A summary of the differences at the institutional layer between the three institutions involved in our course is shown in Table 3.

The level of cohesion between the sites has a strong impact on the challenges. On the one hand, a better cohesion between sites can give better results. On the other hand, it induces higher requirements on compatibility. A common characteristic of institutional rules is that they are significantly more *volatile* for distributed courses and more critical with respect to the course existence. Thereby, the problems should be managed in a similar way as managing faults in safety-critical systems: predict a problem, try to avoid it, make the practice problem-tolerant, and adapt the practice to remove or decrease the consequences of the problem.

In summary, as an answer to RQ1 defined in Section 3, Table 4 lists the challenges related to the institutional layer and the practices to tackle them.

Institutional layer: take-away message

The key point in the institutional layer concerns cohesion between sites. Stronger cohesion improves course quality, attractiveness, and its learning outcomes. Typical measures to increase cohesion are: (i) synchronized common activities, (ii) tight interaction in projects, and (iii) well-defined grading correspondence matrix. However, the stronger cohesion is, the larger the effort is required to achieve and maintain it.

[2]Starting from 2017, it will be moved to mid-September.

Table 4. Challenges on the Institutional Level

Challenge	Measure
Initial agreement and MoU[3]	Use existing rules and practices whenever possible. Very important to have MoU(s) to ensure the course existence. Introduction of a double degree supporting the course and/or multiple enrollments of the students at different sites is useful.
Different semester organization	Ensure flexible start of the course on each site. For example, the site that starts first runs local introductory lectures.
Student enrollment process	Be prepared for long-term work on synchronization. If needed, then enable participation of students in the course before formal enrollment.
Different grading systems and granted credits	Expect a long lasting work during several course instances to find a trade-off between grading systems. If it is not possible to have the same or equivalent amount of credits, then try to balance students' workload related to the credits.
Different learning outcomes	Assume flexibility in the interpretation of the learning outcomes, or try to adjust the course to the learning outcomes specified by the different sites.
Different course quality assessment	Plan for increased effort. Often similar but different assessments are required by different sites, which may require a significant increase of total effort.
Different course budgeted effort	Be ready for additional effort from the teaching staff, sometimes on a voluntary basis. If the course budget is related to the number of enrolled students, then a good practice is to enroll the students from the remote sites.
Course volatility	Being more challenging than local courses, distributed ones risk permanently to be closed down. Ensure course lastingness by a MoU. Having the course as compulsory in a study curriculum mitigates this risk.

5 TEACHING LAYER

In our DSD course, the teaching staff composition at the local sites has incrementally changed over time regarding the number of teachers per site, course managers/examiners, and type of teaching staff components. For instance, at MDH there was a complete handover of the course leadership in 2014. Such a handover could have generated unexpected issues in the course planning and execution. This did not happen, since the new course responsible had already been involved in the course both as a student and, later on, as a course assistant. Moreover, the handover was not carried out overnight, but it was planned two years in advance to make it as smooth as possible.

The composition of the teaching staff and the combination of seniors and juniors can be put in perspective. Initially, the staff was composed of one senior and one junior per site. The presence of juniors (Ph.D. students and post-docs) as project supervisors was embraced to detach supervision from examination (when possible, in relation to the number of students). Initially, it was envisioned that seniors and juniors should be, respectively, project customers and supervisors, although juniors often authored project proposals. The presence of juniors showed to be a gap-closer between students and seniors and helped us in solving critical issues in some cases, so we kept this mixed composition of the teaching staff. However, it was also problematic, since it involved

[3] A memorandum of understanding (MoU) describes a bilateral or multilateral agreement between two or more parties.

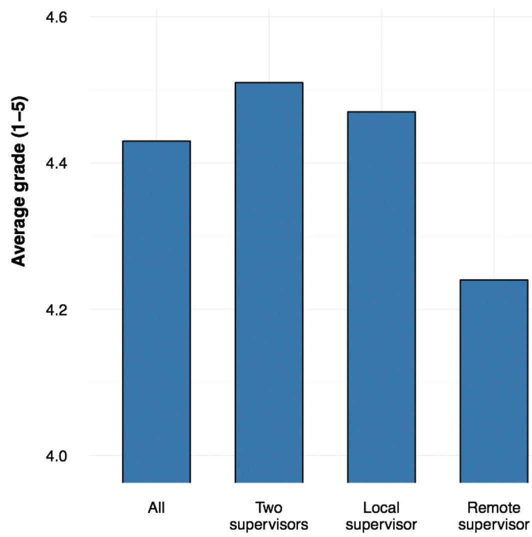

Fig. 1. Overall grades with single and double supervisor settings.

juniors have changed over time since, in most cases, they were temporarily allocated to the course. This resulted in overhead for the senior staff that had to make sure that newly recruited juniors were instructed with all the details needed for them to properly supervise distributed teams and adhere to the course procedures. Fortunately, we experienced also a role evolution of individuals in the course. In five cases former DSD students became junior staff. Of them, two became senior staff, and one became responsible for the course. We also witnessed two former juniors and one former DSD student becoming external project customers once they moved to industry.

The number of involved teachers increased through the years, partly because of POLIMI joining the course, which brought in two new senior staff members, and partly because we decided in 2014 to switch from one to two supervisors per project. This choice was motivated by the observation that a supervisor was more inclined to assign a higher grade to his/her local sub-team against the remote one (see Figure 1, “Local supervisor” and “Remote supervisor” bars). In two cases, we experienced significant differences, up to 1 point on a 3-point scale. Switching to a double supervisor setting has given us a clearer perspective on the performance of the team as a whole, and this has resulted in a more homogeneous and fair distribution of grades. Moreover, having closer supervision from both sides of the team has generally improved the performance of all teams. This has resulted in a slight improvement of the overall grades (see Figure 1, “All” and “Two supervisors” bars).

The effort of the teaching team is organized in two main periods. The first period spans from March to September, when we focus on analyzing the performance of the course in the current academic year and on identifying improvements for the coming year. This retrospective analysis initially consisted mainly of the investigation of the data collected during the course for research and article-writing purposes. Starting from 2014, right after the introduction of SCRUM, this phase has been reinforced and defined in a structured way, and it has influenced the course development process directly since then. Currently, we run it through a number of well-structured workshops specifically dedicated to the following activities:

- *Retrospective analysis*: This is based on the study of students’ and teachers’ feedback on the current year, also compared to the previous editions. We go through the students’ final

questionnaires and the course evaluation questionnaires that each university collects independently. We take into account and discuss any negative comment and suggestion. Each teacher also provides his/her own view on the performance of the class and of the rationale behind issues that occurred during the development of the projects.
- *Improvement definition and actuation*: In case a need for improvement emerges from the previous analysis, we collectively define an improvement plan and actuate it. Examples of recent improvements are: (i) the double supervision approach; (ii) the change in the equipment and communication protocols for our video lectures, which was due to very low video resolution with the previous settings; (iii) the introduction of opponentship, which resulted from the observation that students were not feeling involved during the presentation of the other teams and did not seem to take any advantage well-structured the interaction with them. Regarding (iii), to encourage a stronger participation in the discussions and to exercise the critical capabilities of students, we decided to assign, to each project presentation, an opponent team, which is required to ask pertinent questions and challenge the presenting team. An improvement that we are currently planning is to update the project evaluation sheet to remove some duplicates and unused fields as well as to introduce the possibility to motivate the score for each field. Moreover, we are planning to revise the documentation templates that students have to comply with when documenting the project.
- *Planning for the next edition*: In this phase, we address issues raised in the previous phase. Moreover, we select customers and agree with them on the project proposals for the coming year. Due to the aforementioned barriers in estimating the number of students taking the course, we always select a high number of projects to be prepared for any variation on the final number of students. This, unfortunately, may result in the fact that customers get disappointed when their projects are not assigned to students. To avoid this issue, we take advantage of SCORE projects whenever possible as they do not correspond to a predefined commitment to the corresponding proposers. Other activities that we perform in this phase are the definition of the schedule of lectures and project milestones, and the invitation of guest lecturers.

The course is run in the second period, from October to January. In this period, we give lectures, supervise projects, and coordinate with guest lecturers and customers. Being a project-based course, the most relevant challenges in this period are related to the project layer and described in the next section. Other issues are related to the way lectures are given to the distributed class. The problem that we continuously face is that lectures are given using video-conferencing tools. This implicates a partial perception of the teacher by the distributed class. It is more difficult to see students' eyes and understand if they actively follow or not. We often face technical problems, sometimes related to network issues, other times related to microphones and the habit of some lecturers to walk away from them. It is very difficult for a teacher facing the local class to understand that someone at the remote sites is not following due to bad audio or too much "walking" back and forth. We try to address these issues by constantly keeping a parallel Skype conversation among all members of the teaching staff and by making sure that the teacher giving the lecture is always supported by the others, gets information about possible issues perceived by the others, and gets help with the equipment in case of failure. Another important challenge concerns the selection of lecture topics and their level of detail. Having students with very different backgrounds and skills makes it very hard to identify the right set of topics for all of them. Based on the feedback we gathered through the years, we believe that practical lectures focusing on the process and activities that students have to perform during project development are generally the most appreciated.

Table 5. Challenges on the Teaching Layer

Challenge	Measure
Unforeseeable teaching workload	It is hard to foresee the number of students, and thereby number of projects, taking the course across the three institutions. Teaching staff at each site shall be prepared to last minute re-organization of the teaching assignments across the teaching team and to put extra effort in teaching and supervision activities if more students than expected take the course.
Unexperienced junior teaching staff	Junior staff, mainly responsible for supervision activities, is in most cases temporarily allocated to the course. Well-defined and documented course routines and supervision practices, as well as the in-depth involvement of the new staff in the course preparation, are key factors for a swift and painless integration of inexperienced staff.
Supervision activities format	Every project is unique in terms of positive outcomes and experienced issues. Supervision, regarding the format of its activities, shall be flexible enough to adapt to and mitigate the latter. It is vital that supervision is a cohesive task across the project teams, where supervisors discuss and agree on both common practices and case-specific remedies.

In summary, as an answer to RQ2 defined in Section 3, Table 5 lists the continuous challenges for the teaching layer and the measures to tackle them.

Teaching layer: take-away message

The keypoints related to the teaching layer are to be open minded, to respect the opinions of all, to seek and reach consensus in any decisions, to continuously (i) collaborate in all teaching aspects, (ii) be ready to revise decisions, and (iii) work on course enhancements.

6 PROJECT LAYER

Managing diversity at the project layer proved to be the most complex and dynamic aspect of managing the DSD course. Diversity at this layer does not only represent the sources of risk for both project team and individual student performance. If carefully managed, it provides the necessary ingredients for achieving targeted learning outcomes by making students face the obstacles typical of distributed and multi-cultural project environments.

There are two dominant internal sources of diversity at the project layer: students and customers. However, substantial cross-layer influences can be observed too, mainly from the institutional layer, and they introduce additional externally induced diversities and risks (Table 6). Student-specific diversities are rooted in their educational and cultural backgrounds, but cross-layer influences tend to amplify other sources such as motivation and availability. Differences among properties of involved project customer types (competitions, industry, and academia) include customer availability and responsiveness, requirements definition and volatility, project vs. product focus, and non-balanced access to information within project teams.

In the project layer, we analyze the impact of identified diversities in two temporally distinct subphases: project team definition and project work. The analysis is based on qualitative observations, quantitative data gathered from the aforementioned student final questionnaires, and evaluation of the project results. Additional qualitative classification of project success is introduced by using

Table 6. Cross-layer Influences

Source layer	Diversity	Affected layer	Impact or diversity
Institutional	Semester organization	Project	- uneven workload at different institutions - student availability (exams, holidays, semester start and end dates)
Institutional	Enrollment process	Project	- knowledge diversity - cultural diversity - student number predictability - imbalances in numbers of enrolled students among institutions - student dedication to the course
Institutional	Difference in course credits	Project	- student motivation

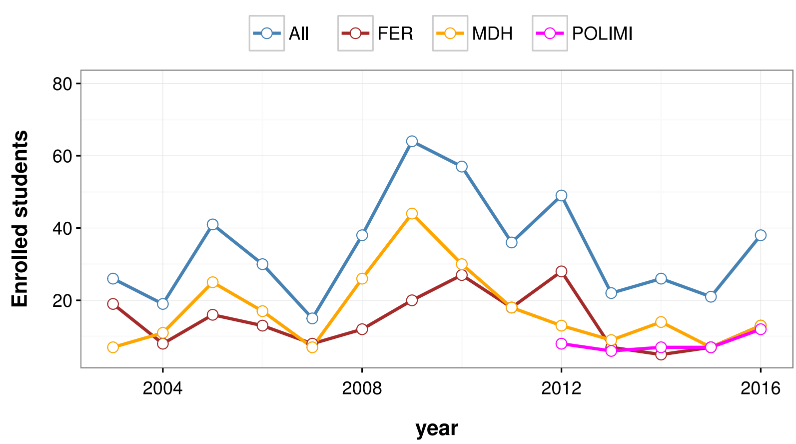

Fig. 2. Student enrollment 2003–2016.

project color grades: *green*—a successful project with only minor process problems and deliverables of good quality, *yellow*—a project with minor to moderate process problems and/or deliverables with lower quality, and *red*—failed project.

6.1 Project Team Definition Phase

The definition of student project teams is a delicate process that must take into consideration course-level, project-level, and personal-level constraints. At *course-level*, it is necessary to include into project teams students from all institutions, balancing the involvement of teaching staff from each institution and the number of location-bound customers. At *project-level*, it is recommended to design suitably sized project teams incorporating project-required knowledge. Also, it is necessary to manage risks related to diversity in students' knowledge and motivation. At *personal-level*, students' preference on team membership and allocation to desired projects should be taken into account, supporting initial students' motivation and self-confidence.

The imbalance in enrolled student numbers at the involved institutions (Figure 2) can result in an imbalanced number of project team members from each location, or in over-/under-sized project teams (Table 7). We have experienced that an unbalanced team composition can have

Table 7. Allocation of Students to Projects 2003–2016

	Year													
Project #	*'03*	*'04*	*'05*	*'06*	*'07*	*'08*	*'09*	*'10*	*'11*	*'12*	*'13*	*'14*	*'15*	*'16*
Local project #	2	-	2	-	-	2	3	-	-	-	-	-	-	-
Over-/Under-sized #	1	-	-	-	-	-	1	1	-	1	-	2	-	1
Imbalanced project #	-	-	-	1	-	-	-	1	-	2	1	-	-	-
Balanced project #	2	3	4	3	2	4	6	7	5	4	2	2	3	5

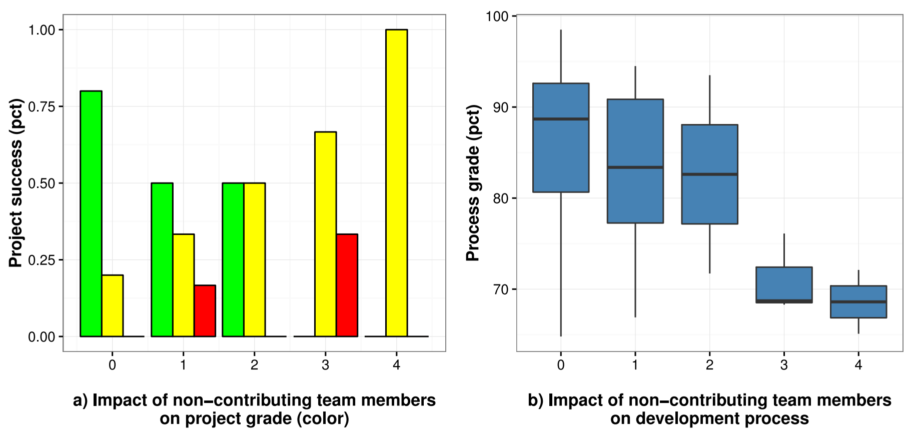

Fig. 3. Final student grade distribution for exchange students and all DSD students.

severe repercussions on the team decision process and, consequently, on the motivation of subdued team members and team cohesion. Extreme situations, when the number of enrolled students is unexpectedly high and/or there is a significant imbalance of enrolled students, may require the creation of local (non-distributed) project teams with customers and/or supervisors at remote locations. This is highly unappreciated by the involved students and cannot adequately support many of the targeted course learning outcomes.

In some occasions in which we had an unexpectedly high number of students, we had to run more than one instance of the same project proposal too. This was beneficial regarding competition between teams but needed additional effort from teaching staff to avoid plagiarism issues.

At project-level, it is necessary to consider knowledge- and technology-related requirements of the proposed projects and to try to form student teams based on explicit information provided in the initial self-evaluation survey and implicit information from the project preference poll. It is also necessary to estimate the impact of educational diversity and minimize the risk by allocating at most one presumably less-knowledgeable or less-motivated student per local team. In our experience, optimally-sized project teams of six to eight members are able to handle up to two less-contributing team members (classified as students with final course grade less than 4). Figure 3(a) depicts the percentage of successful, unsuccessful, and challenged projects at the growth of non-contributing team members within teams. Moreover, Figure 3(b) details the negative impact on the development process. This impact is measured in terms of the score (up to 100 points) we assign to the process part of each project in the evaluation phase. Projects with up to two less-contributing students manage to compensate negative effects with graceful degradation in product/process quality utilizing internal team resources. However, available internal resources

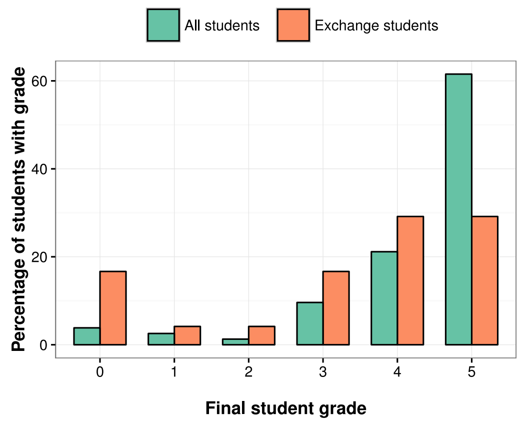

Fig. 4. Final grade distribution (scale 0–5) for exchange students and all DSD students.

are not sufficient to compensate for more than two less-contributing members, resulting in a low-quality development process, and failing or challenged projects.

Surprisingly, as one of the highest sources of risk in forming distributed student teams, we identified exchange students: in the period 2012–2016, exchange students formed 15% of all DSD students population, but with a dropout rate of 16.7% (3.85% in the whole population) and significantly lower individual grades (Figure 4). This risk is a consequence of a combination of differences in personal motivation and lack of both official and unofficial institutional-level mechanisms to prevent inadequately motivated exchange students from enrolling.

On the one hand, the impact of cultural differences has been widely recognized as one of the key sources of inefficiency in distributed development [12, 14]. On the other hand, exposing DSD students to cultural differences and creating awareness is one of the key ingredients of any distributed course. Therefore, we analyze students' awareness/impact on project work by collecting data on observed cultural differences and on particular cultural factors that students deemed to affect their project work. Perception of overall cultural differences within teams is assessed by students selecting among levels of differences: *high*, *medium*, *low*, and *no*. Our educational goal is for students to become aware of cultural differences and to be able to handle them avoiding serious teamwork problems; the ideal student assessment level for such differences is *low*, while a predominance of level *no* would imply lack of awareness in the students, and of level *medium* an indicator of culturally-induced teamwork problems. In our experience, level *high* occurs only as a consequence of personal issues between individual students.

According to our analysis results, cultural diversity of composed project teams does not predetermine the effectiveness of their project work, therefore encouraging the formation of culturally diverse teams. Figure 5 reveals that occurrence of *medium* level perception is not correlated with the number of subregions involved in the project (region and subregion definitions have been adopted from the United Nations geoscheme[4]), but levels of *low* and *no* show slight positive and negative correlations.

At individual-level, student motivation and dedication is taken into consideration by collecting student preferences on offered projects and trying to allocate students to higher ranked ones. Since the project preference poll was introduced, we managed to allocate 90% of students to one of their first three most-preferred projects. However, an analysis of work hours invested and final student

[4]https://en.wikipedia.org/wiki/United_Nations_geoscheme.

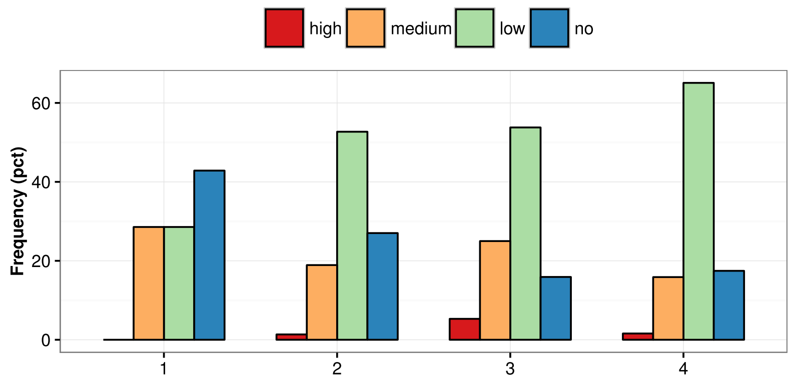

Fig. 5. Cultural differences per number of subregions on project.

Table 8. Customer-type Properties

Customer type	Availability and responsiveness	Definition of requirements	Requirements volatility	Project vs. product focus
university	high	high-medium	medium	process and product
industry	medium-low	high-medium	medium-low	product only
competition	low	medium-low	low	process and product

grades, compared to the preference ranking of the project they were allocated to, did not reveal any correlation between them.

6.2 Project Work Phase

In the project work phase, the dominant source of diversity is the project customer type, but other sources such as educational background, cultural background, and different factors affecting student motivation are revealed too.

Table 8 lists the key differences among customer types involved in our DSD course. These differences create diverse priorities for project teams and influence their internal communication. Therefore, it is necessary for project supervisors to be aware of customer-induced specifics as well as to take both preventive and corrective measures to avoid negative aspects to hamper the project work. Figure 6 presents the analysis results of the evaluation of student projects in the period 2012–2016, grouped by customer types. The upper left chart does not reveal any significant difference in project work hours invested per customer type, but with a higher variation of work hours for *competition* projects (these are the projects offered by the SCORE competition). In addition to higher motivation on *competition* projects, we also attribute this variation to less easily accessible customers, effectively putting students in control of detailed requirements. Similarly, *industrial* and *university* projects exhibit a lower variation in work hours thanks to the higher accessibility of their customers, with *university* projects investing, on average, more hours due to the closeness of both customer and supervisor.

The upper right chart of Figure 6 presents the students' perception of the initial requirements definition and the requirements volatility during the project. All three project types have a comparable mean when it comes to requirements definition perception (*industry* has slightly lower),

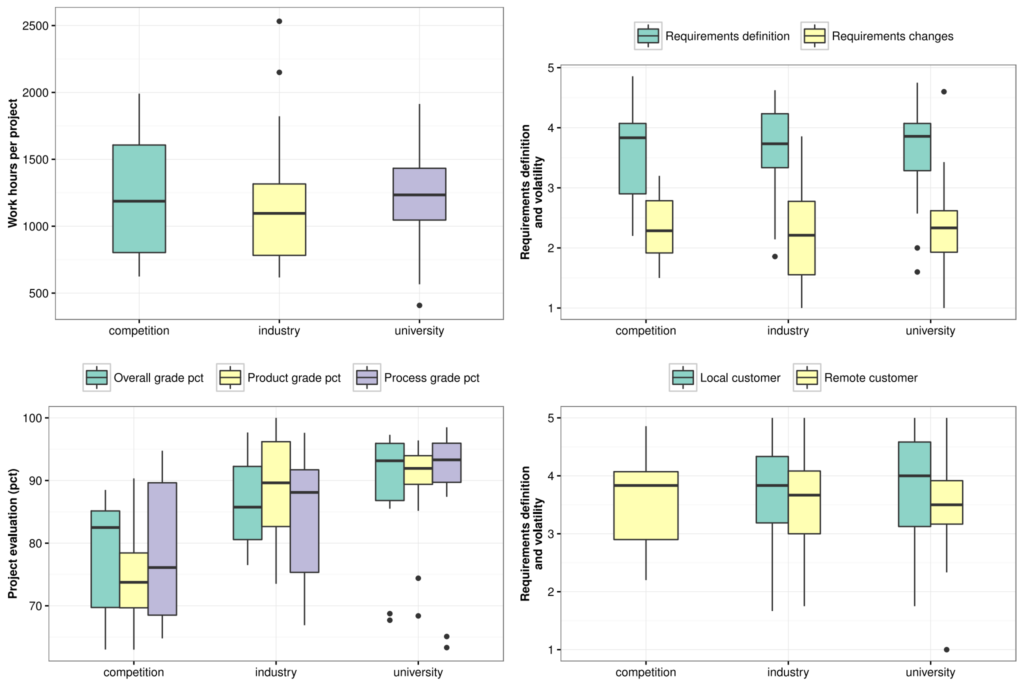

Fig. 6. Customer types and impact on student projects.

again with significant variation for *competition* projects. However, *industry* projects exhibit the highest variation in requirements volatility, implying that the mechanisms adopted by the teaching staff for supervising the initial specification of project requirements are efficient; supervision of requirements change is less efficient. *University* projects display the highest mean and the lowest variability rates thanks to the experience of the teaching staff in defining and guiding this kind of projects.

The lower left chart of Figure 6 presents the evaluation results (overall, project, and process) of projects employing different customer types. *Competition* projects score the lowest (again, due to the self-sufficiency of project teams); particularly interesting is the huge variation in process grade (SCORE competition values the process as well as the product). As expected, *industry* projects tend to put more focus on product than process, while *university* projects, being the most tightly supervised, have a balanced product-process ratio.

The last chart (Figure 6 lower right) reveals the problems in the internal team communication when conveying initial requirements information from customer-located sub-team to the remote sub-team. *Competition* project teams are treated as one remote team, since the customer is not local to any sub-team. Self-sufficiency and high team control of the requirements turn this into a measure of cohesion between the remote sub-teams. *Industry* and *university* projects expose differences in the perception of the initial requirements between customer-located and remote sub-teams, although requiring initial assistance from supervisors to kick-start the project work without immediately inducing imbalance and compromising team cohesion.

Albeit an evident diversity of educational background in the student population, individual differences in knowledge cannot be reliably assessed in the project composition phase. However, educated guesses based on self-evaluation surveys and education profiles of home institutions can provide useful information. Differences in institution education profiles can be deduced from

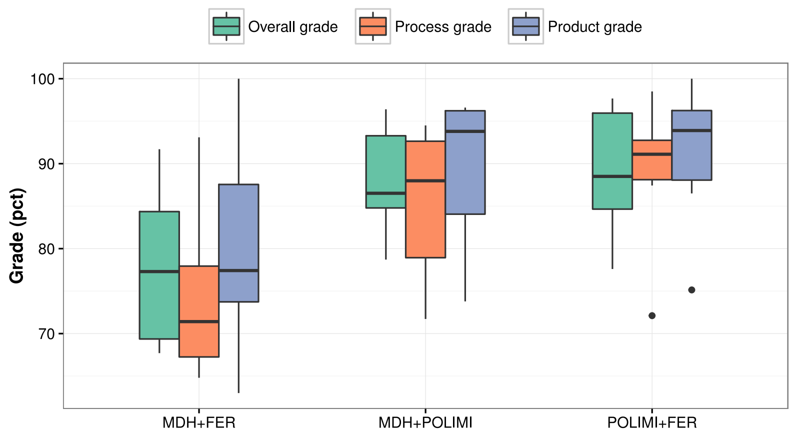

Fig. 7. Diversity in student knowledge.

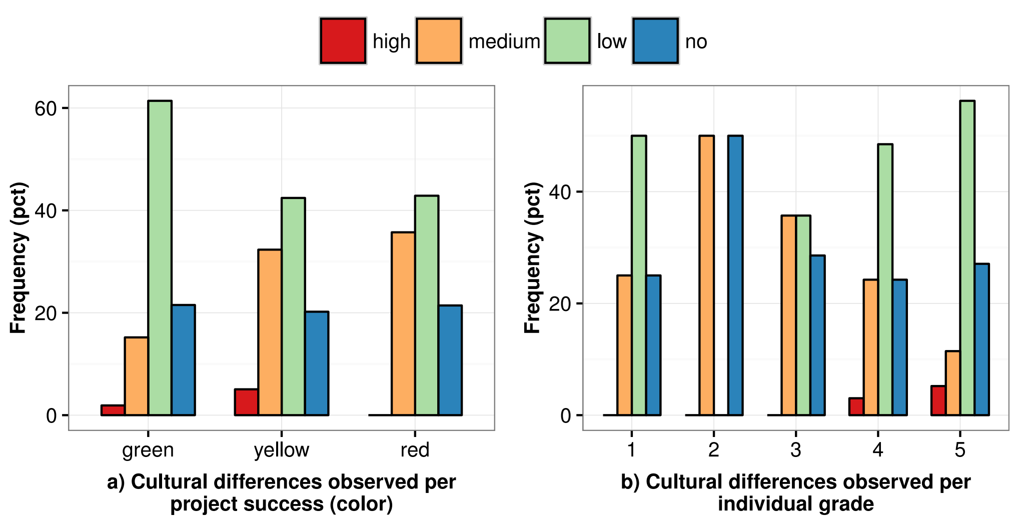

Fig. 8. Observed cultural differences.

empirically collected data on project evaluation over a prolonged period (Figure 7). Analyzing project performance based on pairs of involved institutions reveals differences in overall student knowledge/ability to perform in project settings—integrating both product- and process-wise knowledge.

The observed levels of cultural diversity can be correlated to both individual and project success (Figure 8). The frequency of observed *no differences* tends to be constant for all *project types*, while the lower ratio of *low* and *medium* observation frequencies characterizes lower levels of project success. If correlated with individual student grades, then observation frequencies display a similar pattern: the higher the grade, the higher the *low*/*medium* ratio of observation frequency. Without a statistical proof, our conclusion is that there is no clear cause-consequence relation between cultural diversity and project/individual success; rather a combination of cultural and knowledge differences, together with other issues, leads to the ascription of root cause to cultural diversity by students.

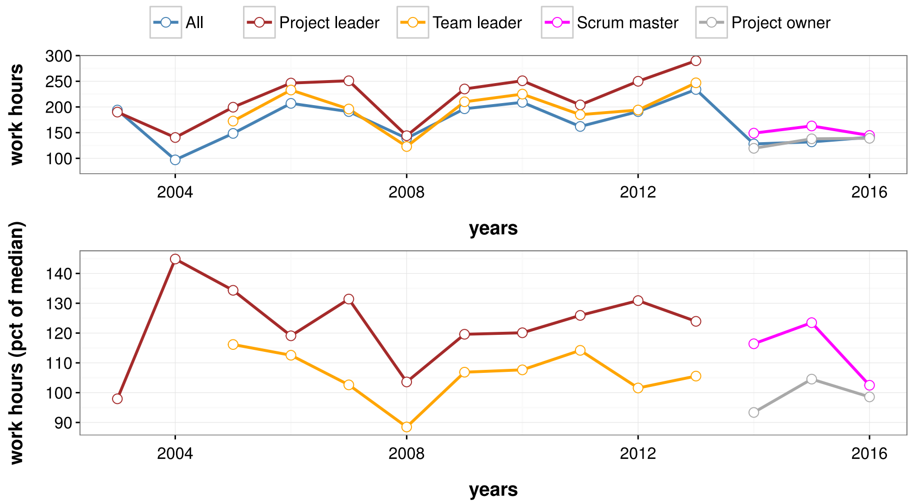

Fig. 9. Average student workload per year.

Differences in course credits among institutions create differences in the student expectations on the required project workload. To accommodate these differences (MDH ~135h, POLIMI ~120h, FER ~160h) in a unified project framework like ours, the option of finding an acceptable common workload is a better solution than insisting on the nominal local effort. The other potential source of uneven student workload is related to the strong commitment of key project roles (Project/Team leader, SCRUM master and Project owner), motivating them to invest more, and in some case of dysfunctional teams, significantly more hours than regular team members. Figure 9—upper chart—presents the median student workload since the beginning of the course, as well as median workloads for significant team roles, while the lower chart lays out the percentage of work hours for significant roles compared to the median of work hours for all team members. Since the introduction of SCRUM in 2014 (in place of an iterative process), we can observe that (i) the median of work hours have decreased to a commonly acceptable level and that (ii) the disproportional workload of key project roles is converging towards the median project workload.

In summary, as an answer to RQ3 defined in Section 3, Table 9 lists the continuous challenges for the project layer and the measures to tackle them.

Project layer: take-away message

To reach the desired learning outcomes, as the main purpose of the project layer, requires to (i) compose project teams taking in mind the overall constraints and different contextual, as well as personal factors such as motivation, knowledge and cultural background, (ii) manage different customer types and projects being aware of their specificities, and (iii) actively supervise and help project teams.

7 RELATED WORK

One of the pioneering works describing the need for training future software engineers to face the new challenges related to distribution was reported by Favela et al. [10]. After that, the research

Table 9. Challenges on the Project Level

Challenge	Measure
Disproportion in enrolled student numbers	All enrolled students at all locations must be accommodated on (distributed) projects; if significant disproportion of students occurs, the following compromises in forming project teams and running projects should be employed: (i) duplicated projects, (ii) over-sized project teams, (iii) undersized project teams, (iv) imbalanced project teams, and (v) local (non-distributed) projects.
Composition of project teams	Differences in educational background, cultural background, motivation, and other factors determine the cohesion and effectiveness of the project team. It is necessary to compose teams with both (i) presence of key distributed project ingredients such as educational and cultural diversity and (ii) minimized risk of team failure due to insufficient/imbalanced knowledge and inadequate motivation and dedication to increased workload.
Different workload expectations	Differences in course credits, stemming from the institutional layer, imply differences in student workload expectations. A compromise must be reached to determine a common acceptable workload expectation from students as one of the key components of the joint project framework.
Customer types	Different customer types bring subtle but significant differences to student projects. For the teaching staff, it is necessary to be aware of those differences and to take preventive and corrective actions, in cooperation with the project team, thus mitigating negative aspects that customer types bring along.

interest on distributed software engineering education has exponentially grown and the related literature is currently particularly rich.

The Runestone project [21] is an initiative very similar to our course, run jointly by universities in Sweden, Finland, and China. Similarly to our course, students are organized in teams across two universities and they are in charge of software development from A to Z. The main difference is that, while in Runestone the teaching staff represents the customers of the projects, in our course, we focus on industrial "real-world" customers, to make the distributed experience as vivid as possible.

Many (more or less formal) empirical studies have been documented throughout the years. Clear et al. [9] report on a systematic literature review with the aim of delineating challenges in teaching distributed courses and providing recommendations for helping teaching staff in overcoming them. Our work is not a secondary research study but rather a primary research study in terms of a longitudinal case-study with the aim of sharing our experiences on a long-lasting distributed course. Petkovic et al. [22] presents methods and experiences from several years of teaching a distributed course across two universities. The focus of the article is on the analysis and evaluation of class effectiveness based on student questionnaires. Similarly, Gotel et al. [11] describe a software engineering education initiative focusing on the trajectory that the initiative took over its first five years. Hoda et al. [13] report on the findings of an empirical study on the socio-cultural aspects of distributed education through a case-study conducted on fourteen participants from ten different universities in eight countries. The authors define seven dimensions of socio-cultural distance that caused the most significant challenges in the course. Another secondary study has been published by Monasor et al. [17] in the form of a systematic literature review of papers related to global software engineering teaching with the aim of discovering existing practices and identify open challenges. Beecham et al. [1] applied a taxonomy for computer science education publications to judge the quality of several papers in the literature related to global software engineering, as well as discuss challenges and recommendations for educators. Beecham et al. [2] also propose an interesting alternative way to address distributed development education by letting students use a kind of simulator or game for their project development. The idea is to exploit simulation to compress time, simulating an entire project work in very short time (e.g., an hour), thus enabling students to try several times with different strategies and thereby learn by experience.

Communication seems to represent one of the most studied issues in distributed education. The work in Reference [16] focuses on the results of an education project that aimed at (i) conveying to students the needed skills for tackling distribution-related issues, and (ii) investigating how different communication patterns impact development projects results. Nordio et al. [19] analyze the effects of distribution and time zones on communication in distributed projects, showing that shorter distance between teams increases communication rate. Also the authors of Reference [24] studied the collaborative behavior of online learning teams and how this behavior is related to communication mode and task type. The paper includes recommendations for building effective online collaborative teams too. Kuhrmann et al. [15] present a course where students learn and experience the role of different communication patterns in distributed agile software development to gain awareness on the importance of communication and the effects on collaboration and team performance. Collaboration is another crucial aspect in distributed courses, and the authors of Reference [7] (partly among the authors of this article) provide a set of collaboration patterns, identified through collecting and analyzing data from 14 distributed student projects, and their causes as well as implications on projects results.

Agile development methodologies introduce additional challenges when enforced in distributed courses. Some recent studies report on the process and outcomes of introducing agile process in established distributed courses as well as the difficulties encountered by the students new to agile in adhering to it [5, 23]. Other works focus on the importance of risk assessment from a teaching perspective when evolving a distributed course over the years [6] and on preparatory exercising for students to prior to distributed project development [18]. Paasivaara et al. [20] describe their experience in augmenting distributed Scrum with industry best practices for a course that is jointly run by three universities and with both co-located and distributed student teams. Students get to experience both co-location and distribution in the same course; the course's results have not shown any significant difference between co-located and distributed work, suggesting that Scrum can help alleviate some issues deriving from distribution.

While the aforementioned works focus on an in-depth assessment of one specific aspect (i.e., communication, collaboration, development process, risk assessment, socio-cultural aspects), our work embraces all three layers (i.e., institutional, teaching, and project) that characterize a distributed course, as well as inter- and intra-layer challenges.

8 CONCLUSION

In this article, we have reasoned on the impact of diversity on three important aspects of a project-based distributed course organization: institutional, teaching, and project.

As concerned the institutional aspects, the institutions involved in the course may face important differences in the number of credits, grading systems, enrollment processes, allocated resources, specific expectations regarding learning outcomes, course volatility. We have provided possible measures to mitigate them, but issues at this level are very hard to solve due to many specific constraints and rules to which each institution must obey. The establishment of MoUs may help in defining a common baseline that, however, will have to coexist with the inevitable differences.

At the teaching level, the difficulty to foresee the teaching load is intrinsic in the nature of the course and is related to the specific characteristics and rules adopted by the involved institutions. As such, there is no specific countermeasure for it except being prepared to handle a different situation every year. The presence of young and inexperienced teachers is certainly mitigated in the case by the presence of other members of the teaching staff (even distributed) that can help newcomers in acquiring the right abilities and attitude toward students. Similarly, a numerous teaching staff composed of peers from various institutions can mitigate differences in the supervision style and, on in the long run, allow the teaching team to identify the optimal one given the types of students being trained. Clearly, this requires a continuous interaction within the teaching staff and willingness of all members to be open to others' opinions and ideas.

At the project level, the disproportion in the number of students from the involved institutions is a fact that cannot be changed. Countermeasures can concern the organization of project teams that can take this disproportion into account. The composition of the team is certainly a crucial factor not only because of the disproportion but also because of educational and cultural differences among students and because of differences in the credits they acquire for the course from the home university. The right balance has to be identified taking into account as many aspects as possible. As such, a careful analysis of the self-evaluation forms compiled by students, together with the judgment of local instructors is of paramount importance. In addition, even when project assignments are carefully checked to give students an as fair set of challenges as possible, different types of customers can lead to different workloads for different project groups. In this case, the only solution is for supervisors to constantly monitor the situation and intervene when needed.

Many of the diversity factors that we mentioned are also visible for "standard" non-distributed courses, such as large diversity of students' experience and skills. Nevertheless, they have a heavier impact on distributed courses and a specific diversity factor can display significant and different characteristics across the sites. This usually requires additional effort from all sites, and by this also additional pressure on a local site responsible for managing this diversity factor.

As we already mentioned, additional diversity factors originate from the peculiar nature of distributed courses—different types of and more challenging communication, distribution of lectures, more complex project team coordination, to mention a few. However, diversity can generate added value—for example, cultural diversity has a clear (positive) impact on increased tolerance, communication skills, and cooperation abilities both among students and within the teaching staff.

As far as we can foresee, the course will continue in the future and might be extended to include other institutions with characteristics similar to ours. We plan to take into account the findings we have identified in this article by applying the lessons we have learned whenever this is possible. In all cases, the strength of the teaching staff and its cohesion and ability to address the situations that will arise is an important factor in success. In fact, it will be up to this staff to mitigate all those diversity factors that cannot be eliminated completely with the main goal of ensuring that the students will achieve the learning objectives in a pleasant context.

All in all, the primary goal of this type of course, i.e., preparing young individuals for their career and life in a globalized world, is worth the extra effort needed for successfully running them. Awareness about continuous challenges in distributed courses helps its implementation and boosts its outcome.

REFERENCES

[1] Sarah Beecham, Tony Clear, John Barr, Mats Daniels, Michael Oudshoorn, and John Noll. 2017. Preparing tomorrow's software engineers for work in a global environment. *IEEE Softw.* 34, 1 (2017), 9–12.

[2] Sarah Beecham, Tony Clear, Daniela Damian, John Barr, John Noll, and Walt Scacchi. 2017. How best to teach global software engineering? Educators are divided. *IEEE Softw.* 34, 1 (2017), 16–19.

[3] Ivana Bosnic, Igor Cavrak, Marin Orlic, Mario Zagar, and Ivica Crnkovic. 2011. Student motivation in distributed software development projects. In *Proceedings of the International Conference on Software Engineering (ICSE'11), Companion Volume to Proceedings of the 3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD'11).* ACM Digital Library. Retrieved from http://www.es.mdh.se/publications/2061.

[4] Ivana Bosnic, Igor Cavrak, Mario Zagar, Rikard Land, and Ivica Crnkovic. 2010. Customers role in teaching distributed software development. In *Proceedings of the 23rd Annual IEEE-CS Conference on Software Engineering Education and Training*. IEEE. Retrieved from http://www.es.mdh.se/publications/1653.

[5] Ivana Bosnić, Federico Ciccozzi, Igor Čavrak, Elisabetta Di Nitto, Juraj Feljan, and Raffaela Mirandola. 2015. Introducing SCRUM into a distributed software development course. In *Proceedings of the 2015 European Conference on Software Architecture Workshops.* ACM, 34.

[6] Ivana Bosnić, Federico Ciccozzit, Igor Čavrak, Raffaela Mirandola, and Marin Orlić. 2013. Multi-dimensional assessment of risks in a distributed software development course. In *Proceedings of the 3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD'13).* IEEE, 6–10.

[7] Igor Čavrak, Marin Orlić, and Ivica Crnković. 2012. Collaboration patterns in distributed software development projects. In *Proceedings of the 34th International Conference on Software Engineering (ICSE'12).* 1235–1244. DOI: https://doi.org/10.1109/ICSE.2012.6227019

[8] Samantha J. Charlick, Jan Pincombe, Lois McKellar, and Andrea Fielder. 2016. Making sense of participant experiences: Interpretative phenomenological analysis in midwifery research. *Int. J. Doctor. Studies* 11 (2016), 205–216.

[9] Tony Clear, Sarah Beecham, John Barr, Mats Daniels, Roger McDermott, Michael Oudshoorn, Airina Savickaite, and John Noll. 2015. Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. In *Proceedings of the ITiCSE on Working Group Reports.* ACM, 1–39.

[10] Jesús Favela and Feniosky Peña-Mora. 2001. An experience in collaborative software engineering education. *IEEE Softw.* 18, 2 (2001), 47–53.

[11] Olly Gotel, Christelle Scharff, and Vidya Kulkarni. 2012. Mixing continents, competencies, and roles: Five years of lessons for software engineering education. *IET Softw.* 6, 3 (2012), 199–213.

[12] James D. Herbsleb and Deependra Moitra. 2001. Global software development. *IEEE Softw.* 18, 2 (2001), 16–20.

[13] Rashina Hoda, Muhammad Ali Babar, Yogeshwar Shastri, and Humaa Yaqoob. 2017. Socio-cultural challenges in global software engineering education. *IEEE Trans. Educ.* 60, 3 (2017), 173–182.

[14] Helena Holmstrom, Eoin Ó. Conchúir, J. Agerfalk, and Brian Fitzgerald. 2006. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *Proceedings of the International Conference on Global Software Engineering (ICGSE'06).* IEEE, 3–11.

[15] Marco Kuhrmann and Jürgen Münch. 2016. Distributed software development with one hand tied behind the back: A course unit to experience the role of communication in GSD. In *Proceedings of the IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW'16).* IEEE, 25–30.

[16] Juho Makio and Stefanie Betz. 2009. On educating globally distributed software development—A case study. In *Proceedings of the 24th International Symposium on Computer and Information Sciences (ISCIS'09).* IEEE, 480–485.

[17] Miguel J. Monasor, Aurora Vizcaino, Mario Piattini, and Ismael Caballero. 2010. Preparing students and engineers for global software development: A systematic review. In *Proceedings of the 5th IEEE International Conference on Global Software Engineering (ICGSE'10).* IEEE, 177–186.

[18] Martin Nordio, H. Christian Estler, Bertrand Meyer, Nazareno Aguirre, Rafael Prikladnicki, Elisabetta Di Nitto, and Anthony Savidis. 2014. An experiment on teaching coordination in a globally distributed software engineering class. In *Proceedings of the IEEE 27th Conference on Software Engineering Education and Training (CSEE&T'14).* IEEE, 109–118.

[19] Martin Nordio, H. Christian Estler, Bertrand Meyer, Julian Tschannen, Carlo Ghezzi, and Elisabetta Di Nitto. 2011. How do distribution and time zones affect software development? A case study on communication. In *Proceedings of the 6th IEEE International Conference on Global Software Engineering (ICGSE'11).* IEEE, 176–184.

[20] Maria Paasivaara, Kelly Blincoe, Casper Lassenius, Daniela Damian, Jyoti Sheoran, Francis Harrison, Prashant Chhabra, Aminah Yussuf, and Veikko Isotalo. 2015. Learning global agile software engineering using same-site and cross-site teams. In *Proceedings of the 37th International Conference on Software Engineering—Volume 2.* IEEE Press, 285–294.

[21] Arnold Pears and Mats Daniels. 2010. Developing global teamwork skills: The Runestone project. In *Proceedings of the Education Engineering Conference (EDUCON'10).* IEEE, 1051–1056.

[22] Dragutin Petkovic, Gary Thompson, and Rainer Todtenhoefer. 2006. Teaching practical software engineering and global software engineering: Evaluation and comparison. *ACM SIGCSE Bull.* 38, 3 (2006), 294–298.

[23] Christelle Scharff, Samedi Heng, and Vidya Kulkarni. 2012. On the difficulties for students to adhere to scrum on global software development projects: Preliminary results. In *Proceedings of the Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD'12).* IEEE, 25–29.

[24] Fatma Cemile Serçe, Kathleen Swigger, Ferda Nur Alpaslan, Robert Brazile, George Dafoulas, and Victor Lopez. 2011. Online collaboration: Collaborative behavior patterns and factors affecting globally distributed team performance. *Comput. Human Behav.* 27, 1 (2011), 490–503.

Received April 2017; revised November 2017; accepted March 2018