PROJECT-BASED LEARNING IN AN INTERNATIONAL VIRTUAL CLASSROOM TO TEACH GLOBAL SOFTWARE ENGINEERING

J.M. Olivares-Ceja¹, B. Guitierrez-Sanchez¹, P. Brockmann², A. Kress², J. Stauffer²

¹ Computing Research Center, National Polytechnic Institute (MEXICO)
² Computer Science Department, Nuremberg Institute of Technology (GERMANY)

Abstract

Software development today is a global process. Multi-national teams, in different time zones, with different native languages, all need to work together on large software projects. Many global software development projects have failed to deliver the expected cost savings or have not been completed on time. Programming skills alone are not enough for today’s IT graduates. In addition to knowledge of agile project management methods, intercultural understanding is vital in order to work together in geographically distributed, international teams. Due to time and budgetary constraints, few computer science students have the opportunity to acquire the international experience necessary to work together with software developers from different cultural backgrounds.

The question arises about what is the best method to teach these intercultural soft skills. In this paper, a virtual, team-teaching experience between two universities is described: The National Polytechnic Institute in Mexico City, Mexico and the Nuremberg Institute of Technology in Germany. At each university, a team of approximately 20 students in each country cooperate on a group project. Communication between the two groups is limited to video conferences, chats and e-mail. The teams exchange documents and code using cloud-based project management and collaboration software.

Two different teaching methods are compared for two different semester classes: a traditional, instructor-based lecture vs. a project-based learning approach. Results are presented comparing the results of these two approaches.

Keywords: Project-Based Learning, Global Software Engineering, Virtual Classroom, International.

1 INTRODUCTION

Information technology enables people world-wide to share and communicate. The software which people use to connect and collaborate with one another is often developed by multi-national teams. Team members can be located in different geographic locations, in different time zones and speak different native languages. Global software development presents a number of new challenges:

- Geographic distance: It can take more than twice as long to complete the same task in a distributed team as it would take for a co-located team [1].
- Time differences: Theoretically, time zone differences should enable distributed teams to work around the clock. In reality, they often hinder synchronous, real-time communication between team members at the different locations [2].
- Language differences: Groups in countries which speak different languages may encounter difficulties communicating verbally in real-time. Each group may need extra time to translate their thoughts into a common foreign language, such as English, which serves as an additional communication filter. Non-native speakers may feel reluctant about participating in telephone or video conferences [3].
- Trust: Teams in different countries who have never actually met often experience difficulty in establishing the types of informal communication which comes naturally to co-located teams [4].
- Cultural differences: Misunderstandings due to cultural differences have repeatedly proved to be one of the most difficult challenges to overcome in successful global software engineering [5], [6].

This means that programming skills alone are no longer sufficient for IT graduates. Today’s IT graduates need to learn both international project management methods as well as the intercultural skills necessary to collaborate with team members in other countries. According to Romero [7], a wide
variety of soft skills are necessary for success in global software engineering, such as communication, teamwork and conflict resolution skills. He emphasizes the importance of following intercultural abilities: self-criticism, dealing with uncertainty and ambiguity, appreciating diversity and multiculturalism, understanding cultures and customs of other countries.

Due to time and budgetary constraints, few students have the opportunity during their study programs to spend an entire semester abroad in order to acquire international experience. Thus, the question arises as to which is the best method to teach students these vital intercultural skills. In this paper, a virtual, team-teaching experience between two universities is described: the National Polytechnic Institute in Mexico City, Mexico and the Nuremberg Institute of Technology in Germany. At each university, a team of approximately 20 students in each country cooperate on a group project to develop a software product. Two teaching methods are compared: instructor-centered lectures and Project-Based Learning (PBL).

2 BACKGROUND

2.1 Intercultural Challenges in Global Software Engineering

One of the first researchers to use empirical methods to investigate the cultural differences in the software industry was Hofstede [8], who applied multivariate statistical methods to analyze data collected from thousands of IBM employees world-wide. Hofstede classified differences in cultural perspectives according to six dimensions:

- Power distance: The attitude of a society to inequalities among individuals in a society (PDI)
- Collectivism vs. individualism: The degree of interdependence among members in a society (IDV)
- Masculine vs. feminine: Success and achievement vs. caring for others (MAS)
- Uncertainty avoidance, ambivalence: Feeling threatened by unknown situations (UAI)
- Long-term vs. short-term orientation: Planning for the future vs. living in the present (LTO)
- Indulgence vs. restraint: The extent to which life is to be enjoyed vs. showing restraint (IND)

These dimensions can be used as a framework to make international project teams more aware of country-specific differences. With respect to our situation, Hofstede’s analysis [8] shows that Mexico and Germany differ significantly on these five cultural dimensions:

![Figure 1: Graph of Mexican Hofstede dimensions.](image1)

![Figure 2: Graph of German Hofstede dimensions.](image2)

In addition to these five cultural dimensions, Laroche [9] differentiates between two ways how different cultures perceive time: M-time (monochromatic) and P-time (polychromatic). Monochromatic cultures, such as Germany, tend to start and end a meeting at a precisely scheduled time. Polychromatic cultures, such as Mexico, may feel that such a meeting is being rushed through and then cut off abruptly, before they have a chance to adequately express their views.
As evident from the comparisons above, Mexico and Germany differ considerably on the cultural dimensions of power distance, individualism, long-term orientation and time perception. Masculinity and uncertainty avoidance score high for both cultures. The two teams of students need to consciously adapt to different views on the power distance between students and professors. German students need to recognize that their individual grades for the course may not be as important as the success of the entire group. The dynamic, short-term flexibility of the Mexican team can be unsettling for the German team, who are used to long-term planning. The polychromatic time perception of the Mexican team can conflict with the monochromatic German view of time.

2.2 Teaching Global Software Engineering

Beecham, et.al. [10] and Clear, et.al. [11] conducted systematic reviews of papers to define the challenges facing global software engineering education, including: global distance, teamwork, soft skills, stakeholders, infrastructure, development process, and curriculum. Hoda, et.al [12] discuss socio-cultural challenges in global software engineering education. They conducted a case study of 14 participants from 10 different universities in 8 countries. They identified six dimensions which caused significant challenges: 1) language 2) concept of time 3) attitude towards grades 4) assumptions about national culture 5) autonomy 6) influence of lecturer. Fortaleza, et.al. [13] conducted a literature review of 19 courses in global software engineering in 25 countries. They propose creating a collective repository to store and report research experiences to build a teaching network community.

A number of other European universities have conducted learning experiments designed specifically for global software engineering. Deiters [14] developed a common environment for teaching a course in software engineering, which was geographically distributed over four universities within one country. In this case, the emphasis was primarily on overcoming geographic distance. Team members shared a common language and culture, but had to cooperate remotely from different sites. A different joint European cooperation [15] developed a course on software design for teams distributed in different European countries. The project had to overcome the additional challenge of team members who spoke different languages, but who shared a common European cultural heritage. Petkovic [16] described an international cooperation to teach global software engineering between an American and a German university. In addition to the language barrier, significant time differences between Germany and the U.S. also had to be addressed. A virtual, team-teaching cooperation between two universities in Germany and Mongolia is described by Ende, et.al. [17]. In addition to the expected geographical and temporal differences, enormous barriers in language and culture presented huge obstacles for the students. Solutions to anticipated problems, such as distance and time differences, could be alleviated by scheduling video conferences in advance. Although the language barrier could be somewhat alleviated by translation software, the cultural barrier proved to be almost insurmountable.

3 METHODOLOGY

3.1 Project-Based Learning (PBL)

This paper investigates the effectiveness of “Project-Based Learning” (PBL). Although PBL is often used interchangeably with the term “Problem-Based Learning”, the focus is slightly different. Problem-Based Learning was first introduced to teach medicine at the McMaster University in Canada [18]. The idea is to replace instructor-centered, frontal lectures which emphasize the passive consumption of material and the rote memorization of facts. As an alternative, students actively self-organize to investigate and construct their own solutions in a case study. The central hypothesis of problem-based reasoning is that students learn more effectively when the learning process is centered on a concrete problem. Barrows [18] defined six major principles of problem-based learning:

1 Learning is student-centered.
2 Learning occurs in small student groups.
3 A tutor is present as a facilitator or guide.
4 Authentic problems are presented at the beginning of the course, before any preparation or study.
5 Problems presented used as tools to gain necessary knowledge and problem-solving skills.
6 New information is acquired through self-directed learning, objectives and assignments.
Project-Based Learning (PBL) is similar to problem-based learning. According to Bell [19], students drive their own learning through self-guided inquiry while working on a project. By working collaboratively on authentic projects, students acquire both problem-solving as well as communication skills. Instructors provide guidance, feedback and suggestions according to the needs of the students within the context of the project. Because the project is defined as an external outcome, Savery [20] criticizes that this diminishes the learner’s role in defining the problem goals and parameters. This experiment implements PBL in spite of this criticism, because projects in software engineering are almost always defined by external stakeholders rather than by the students themselves.

Richardson and Delaney [21] present the use of PBL in teaching software engineering. They applied a hybrid PBL approach in an undergraduate class which included five Irish students and three foreign exchange students. They found the approach especially effective in teaching vital soft skills. Woodward [22] developed instructional modules based on PBL to teach information systems. A combination of experiential learning, cooperative learning strategies and a dialog-based analysis of cases was shown to have a positive effect on the development of students’ soft skills. Rodrigues and Santos discuss a framework for applying Problem-Based Learning to computing education [23]. They emphasize the need to adhere rigorously to the pre-defined processes of the method. Mendes Silva, et.al. [24] found that an adapted version of PBL gave more realism to teaching software engineering. Nuha [25] found that PBL correlates quite well with Scrum [26] practices in software engineering.

3.2 Description of the International Team-Teaching Experiment

In this paper, a virtual, team-teaching experience between two universities is described: the National Polytechnic Institute in Mexico City, Mexico and the Nuremberg Institute of Technology in Germany. At each university, a team of approximately 20 students in each country cooperate on a group project to develop a software product. Two different teaching methods are compared for two iterations of the same class (Global Software Engineering) conducted in two different semesters:

1. A project-based learning approach (PBL) in March 2016,
2. Instructor-centered lectures followed by a project phase in Winter Semester 2016/2017.

For the project-based learning approach, a real-world international software development project was simulated. Students participated in an intensive, all-day block seminar which ran for seven consecutive days. Students were assigned a messy, real-world problem, without detailed instructions on how to solve it. Communication between the two groups was limited to electronic means: video conferences, chat and e-mail. The teams exchanged documents and computer code using cloud-based project management and collaboration software. Due to the seven hour time difference between Mexico and Germany, students only had about a one hour time window each day when they could communicate in real-time via video conference. All other communication was conducted asynchronously. At the end of each group’s working day, they sent a report of their progress and any questions they had to the other group. At the beginning of each group’s working day, they first reviewed the progress reports and questions which the other group sent them overnight while they were asleep. This method of distributed team work is often referred to as “Follow the Sun” [27].

During the traditional, instructor-centered lecture, students were given weekly lectures about different aspects of global software engineering. After this initial eight-week lecture period, students from each group attempted to collaborate on a group project.

3.3 Collaboration Tools for International Projects

There are different challenges when working as geographically separated teams, which need to be taken into account by selecting and properly using tools for communication and organisation between and within teams. These tools must meet certain requirements. For a student project, the tools should be available without additional cost to the participating universities. An intuitive user interface is desirable, as users tend to adapt and use these tools more willingly.

In addition to regular e-mail communication, it is essential to support fast and direct exchange of information to define as well as to coordinate the common goals of a collaboration. Video and voice communication have shown to be the most effective during the projects described in Subsection 3.2 – especially if backed up with instant messaging for asynchronous information sharing. Considering that other countries have different service quality for internet connections, bandwidth must be considered as well. Instant messaging can also function as backup communication method if the bandwidth is too low to support video or voice chat.
To support the project described in Subsection 3.2, Google Hangouts was picked by the students as tool of choice that fulfils all of the requirements. It is similar to Facebook in look and feel, and therefore is easy to use, but focuses more on video and voice communication. To use this free tool on a desktop PC, a Google account and the installation of a browser plugin is required. Moreover, all of the common mobile devices are also supported. Google Hangouts organizes communication in groups created by users. It provides the option to engage in conference-like instant messaging as well as to call all members of a user group. Alternatively, users can also be contacted directly via a private two-person group conversation. Google Hangouts was chosen over similar tools because of the free accessibility, the number of students who already had the necessary account and the overall low bandwidth requirements [28] the tool comes with.

After the means of communication were defined, it was important to be able to organize and coordinate the shared effort of all participants. Groups of students should be able to work on shared tasks independently without having to wait for each other and with sufficient transparency to have an outline of the current work's state. To enable this type of agile development process, pin board styled tools are common. All users share common boards and each board has columns added with post-it like cards. Each card represents a task that must be completed to achieve the project's goals. During the project's execution; Trello was chosen as organisational tool, as it fulfils these requirements and the students already had experience with it. Trello can be accessed by creating a specific Trello account or by using an existing Google account, which fits well together with using Google Hangouts for communication. It is rather intuitive to set up, with a more complex rights management to be done if multiple users are supposed to share and access a single (pin) board. On a board, it is possible to create multiple columns, as seen in Figure 3.

![Figure 3: Example of a Trello pin board to organize tasks](image)

Cards can be created and assigned to columns and afterwards moved freely between columns or even different boards. Besides a title, a card includes a description field where any text with somewhat limited text design can be put into. The cards can be assigned to specific users and given a due date. To group cards together for clarification, they can be labelled with text and colour to make them easy distinguishable. Each card can be commented on, as is common on social media websites. A strict history of user actions is appended to each card. The best-known application achieving a similar agile development process is Jira, which requires a license and is not available for free.

### 3.4 Measurement of Skills and Experience Gained

Castro-Hernández, et.al. [30] compared interaction-based measures and their ability to predict task cohesion within global collaborative learning environment. They found a statistically significant correlation of linguistic characteristics and information exchange similarity with task cohesion, when controlled by culture. Quantity-based metrics, such as the information exchange rate, were an even better predictor of cohesion within distributed learning teams than similarity-based measurements.
In this work, in spite of cultural differences, it is important to detect benefits to each participant team. The German team was made up of master’s degree students, who had more experience in the following methods: agile project management methodologies such as Scrum [26], the tools used in software development projects (as described in Section 3.3), and English as a second language to communicate with people in neighboring countries. It is important to explicitly recognize these differences. For the Mexican team, the situation was quite different. Scrum in European countries is a term associated with the rugby, with a group of people pushing another group to get a goal, while in Mexico it is an unfamiliar term that needs to be explained and exemplified to be understood. English in Mexico has only recently become mandatory for bachelor degree students, therefore only few students were able to maintain a conversation. Global software engineering is not a widespread term. With the above scenario, the motivation is to exhibit the measurement of factors from the Mexican side involved in the global software development experience to produce a Web site that provides local information to foreign students during their mobility.

The factors considered in this study are:

1. Global team coordination.
2. Experience using a common language to collaborate with foreign developers.
3. Usage of tools that support Global development, in this case: Trello(R) for activities management, Slack(R) for communications purposes, and GitHub(R) to share the software developments.
4. Experience presenting result on fora.
5. Experience developing software using agile methodologies.

At the beginning and at the end of each course, students filled out a questionnaire asking them to judge how important the following seven factors were for the success of a global software engineering project:

1. Geographical distance
2. Time zone difference
3. Different native languages
4. Proficiency in a third, shared language
5. Cultural differences
6. Familiarity between teams
7. Trust between teams.

4 RESULTS

4.1 Results from the project-based learning method

As two quite different cultures worked together on this collaboration, it is interesting to have a look at the results of the other team involved. Right from the start, it was expected that the results on the German side would be seen from a different perspective than on the Mexican side.

According to Hofstede [8], the high value for uncertainty avoidance means that ambiguous situations can be extremely disquieting for Germans. German students usually expect detailed specifications and clear instructions. To test this hypothesis, the PBL group [18], was intentionally assigned a vague, messy project. This forced them to improvise and organize without specific instructions about the role they should play, which resulted in the group feeling uneasy. This strongly influenced initial contacts with the Mexican collaboration partners. German students felt a need for clarity. For instance, during the first video-call with the Mexican students, the German students predominantly dealt with the need to gather information about how the collaboration should proceed and what the roles of each team should be. This typical attention to “business first” intimidated the Mexican students, who expected an initial phase of social contact to ease team-building.

The PBL method caused an intense demand for communication both within each team as well as between the two teams. With a focus on the intra-team workload organization, it can be concluded that PBL had a positive impact on the German team. As the project progressed, the team identified what
the difficulties in this intercultural collaboration were and how to alleviate them. The primary focus was on cultural specifics and a clear definition on how the collaboration would take place, including organizational and capacity-affecting topics. The most important adjustments implemented were:

- Discussion agenda: After the first meeting with the Mexican team, the German team understood that the timeframe to communicate was shorter than planned and therefore had to be organized more stringently. Meetings were planned based on a written discussion agenda, which each group received ahead of time.
- E-mail communication: E-mails often remained either unanswered or were not read reliably outside of class time. Since e-mails could only serve as a one-way information channel, any coordination agreements had to be made during video conferences.
- Excursiveness and spontaneity: In contrast to habitual German inflexibility, Mexican students improvised ideas and goals quite agilely. German students were surprised by sudden requests for additional requirements during the project, which had not been initially agreed on at the outset. They had to learn to break with their habit of sticking to detailed plans and to try to be as agile as their Mexican collaboration partners, where necessary.

4.2 Results from the lecture-based method

During the winter semester of 2016/2017, the next iteration of students were first introduced to the topic of Global Software Engineering through a series of traditional lectures. The class met once a week for four hours, with an initial lecture period which lasted eight weeks, followed by a project cooperation period of four weeks. The lecture topics presented the theory and methods central to global software engineering, including IT offshoring, nearshoring and reshoring, adaptation of agile software project management methods to distributed teams, intercultural aspects of global software engineering, as well as team-building and conflict resolution in international projects.

The initial and final results from the Mexican team are presented in Table 1. The results are also presented in Figure 4, showing important benefits to the Mexican team, who for the first time interacted with a foreign team using a different language, tools, and a project that needed collaboration to produce a shared result.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Measure</th>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global team coordinators</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Common language usage</td>
<td></td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Global teams tools usage</td>
<td></td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Global programming</td>
<td></td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Forum experience</td>
<td></td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

When comparing their respective contributions to the final project, it was observed that the more experienced students on the German side contributed more than the Mexican participants. The formula (1) is proposed to give a measure of the participation that in such a case could serve to give an idea of contributions and if it is the case to divide benefits. Ideally, this formula should have 50% for each of the factors when teams share similar skills and experience.

\[ F_1 + F_2 = 1 \]

The factors \( F_1 \) and \( F_2 \) are used to equilibrate the contributions to the product. In this case the values are estimated as \( F_1 = 3/4 \) for the German team, and \( F_2 = 1/4 \) for the Mexican team. These measures indicate that the German team tripled the contributions considering their Mexican counterpart. If more teams participate, additional factors should be included, resulting in lower fractions.
4.3 Comparison of results from project-based vs. lecture-based methods

Prior to and after completing the learning project, the German students were asked to complete a questionnaire to measure their impression of factor relevancy for international collaboration. The questionnaire’s scale reaches from 1 (not important) up to 4 (very important). Although the number of participants in each survey (19 for project-based, 14 for lecture-based) was not large enough to be statistically significant, differences between the two groups can be discerned.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Method: PBL</th>
<th></th>
<th>Method: Lecture</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial</td>
<td>Final</td>
<td>Initial</td>
<td>Final</td>
</tr>
<tr>
<td>Geographical distance</td>
<td>2.47</td>
<td>3.00</td>
<td>2.79</td>
<td>2.93</td>
</tr>
<tr>
<td>Time zone difference</td>
<td>3.63</td>
<td>3.75</td>
<td>3.50</td>
<td>3.29</td>
</tr>
<tr>
<td>Language differences</td>
<td>3.21</td>
<td>2.60</td>
<td>2.93</td>
<td>3.43</td>
</tr>
<tr>
<td>Proficiency in shared language</td>
<td>2.84</td>
<td>3.00</td>
<td>3.86</td>
<td>3.50</td>
</tr>
<tr>
<td>Cultural differences</td>
<td>2.52</td>
<td>2.95</td>
<td>3.00</td>
<td>2.79</td>
</tr>
<tr>
<td>Familiarity between teams</td>
<td>2.58</td>
<td>2.90</td>
<td>3.00</td>
<td>3.29</td>
</tr>
<tr>
<td>Trust between teams</td>
<td>3.47</td>
<td>3.55</td>
<td>3.79</td>
<td>3.50</td>
</tr>
</tbody>
</table>

For the students in the PBL Semester in March of 2016, 10 of the 19 German participants already had experience in working with teams from other nations. None of the students could speak Spanish, most students judged their English skills to be “advanced”. All of them were native German speakers. The perceived importance of “geographical distance” and “cultural differences” increased the most. These aspects have proven to be more important than initially anticipated. Their experiences with the Mexican team changed their ideas about the importance of punctuality and communication. Also, the factor “familiarity between teams” was deemed more important at the end of the project. The importance of language differences declined over the duration of the project. This is a result of the insight that if both teams speak a shared language, the differences in language matter less. At least one of the collaboration partners on each side should be sufficiently proficient in the shared language to enable effective communication.

During the lecture-based semester in the winter of 2016/1027, one student from the Mexican team could speak excellent English and served as the manager of communication for the team in Mexico. On the German team, one student could speak Spanish, three students were from countries other than Germany: Denmark, Portugal and Russia. Most of the students on the German team classified their English skills as “advanced”. This may explain the large discrepancy in the opinions about the importance of language differences between the classes in two different semesters. The diversity of
languages within the German team gave them a greater appreciation of language differences overall. The opinion of importance of familiarity between teams also increased similarly to the students from the PBL semester. The changes in opinion regarding the importance of the other factors, such as geographical distance, time zone differences, trust between teams and cultural differences was not as drastic as during the PBL semester. During the lecture phase of this course, students did not experience levels of stress comparable to the PBL semester. Students reported passively consuming the information presented, without necessarily internalizing it.

5 CONCLUSIONS

As a result of the experience gained during two semesters of the same class taught by PBL in March of 2016 and by instructor-based lectures in the winter of 2016/2017, the following conclusions are presented: Students reported that PBL was much more challenging than a traditional lecture. Held as an all day, one week course during the semester break, it allowed students to participate in a realistic simulation of an international project, in two countries with a seven hour time gap, different languages and cultures. At the beginning of the experiment, some of the German students reported feeling anxious due to the lack of specific instructions on how to achieve the project goals. At the end of the seminar, many team members complained of exhaustion. The evaluation and feedback meeting after the project’s end showed that the team valued this real-world experience working on an international project, instead of just learning about it theoretically.

It should be noted that the students enrolled were either in the final year of their bachelor’s degrees or were graduate students. Although PBL could theoretically be used with less experienced participants, the danger of cognitive overload, as described by Kirschner [30], should not be ignored.

In conclusion, PBL had quite a positive effect on the project members’ performance and on their learning success. Although none of the students expressed a desire to have all of their classes taught exclusively according to PBL, they judged it to be a much more effective method to deepen knowledge and gain experience international projects than conventional, instructor-based lectures. Measured by the subjective experience and learning success, students felt they had gained more relevant knowledge by experiencing the situation firsthand.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation for the support of their two universities: the National Polytecnich Institute of Mexico and the Nuremberg Institute of Technology in Germany.

REFERENCES


