

Multinational, Intercultural, Multidisciplinary & Intensive (MIMI) Methodology to Enrich Soft Skills Development in Computer Science Students

Shane DOWDALL¹, Artur HŁOBAŻ², Piotr MILCZARSKI²,
Derek O'REILLY¹, Krzysztof PODLASKI², Zofia STAWSKA²

¹*Department of Visual and Human-Centred Computing, Dundalk Institute of Technology, Ireland*

²*Faculty of Physics and Applied Informatics, University of Lodz, Poland*

e-mail: shane.dowdall@dkit.ie, artur.hlobaz@uni.lodz.pl, piotr.milczarski@uni.lodz.pl,

derek.oreilly@dkit.ie, krzysztof.podlaski@uni.lodz.pl, zofia.stawska@uni.lodz.pl

Received: November 2020

Abstract. The development of communication and other soft skills among computer science students is not usually an easy task. Often, curricula focus on technical skills, with team projects being used for the improvement of communication skills. However, these teams usually comprise solely of computer science students. In this paper, we present a didactical methodology, called MIMI, which can be used in a short, intensive, programme for undergraduate students. This methodology has been implemented in real projects that have run annually since 2014. We advocate the use of team-based projects, with an important requirement that each team is both multidisciplinary and multinational. Additionally, the period of teamwork is short and intensive. A significant role in the project is given to team mentors. A mentor is a person, usually a university lecturer, who helps the team organize their work and tracks if the team's planned didactical results are being achieved. The program has proved to stimulate an increase of soft skills among the students who participated and, in particular, among the computer science students. The detailed description of our process will allow others to implement and build similar events in their university or company environments, the focus of which is a Multinational, Intercultural, Multi-disciplinary & Intensive (MIMI) methodology approach.

Keywords: soft skills, multinational project, intercultural project, multidisciplinary project, intensive project, teamwork, problem-based learning, project assessment.

1. Introduction

Employees in IT companies often work in multinational, intercultural and multidisciplinary project teams. In many teams, there is a mix of both IT specialists and people from different disciplines Börstler and Hilburn (2015); Frezza *et al.* (2019). An IT

specialist is no longer “hidden” from the rest of the company. IT products are usually developed using one of the agile methodologies Gestwicki and Sun (2008); Ibanez *et al.* (2014); Reardon and Tangney (2015) and require very close cooperation between IT specialists and non-technical colleagues. As a result, any modern IT employee has to be able to communicate effectively with both IT specialists and non-specialists. The list of requirements for an IT specialist now contains a high level of English language skills and the ability to work and communicate in multinational, intercultural and multidisciplinary teams Gestwicki and McNely (2016); Frezza *et al.* (2019). Modern IT specialists also require the ability to present ideas clearly and professionally Pastel *et al.* (2015); Frezza *et al.* (2019); Saltz and Heckman (2018). They need to be able to properly identify an IT solution for a problem described in a non-technical way and be able to present this solution Ahmed *et al.* (2012); Herrmann and McFarland (2019); McKinsey Center for Government (2014); Sorensen and Mas (2015); Forum Word Economics (2016). Critical thinking, innovative problem-solving capabilities, cross-cultural understanding, self-awareness, and assertiveness are also desirable skills. These requirements, in turn, present a challenge to the curricula of third level institutes Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society (2013); Marques *et al.* (2014); An *et al.* (2019); Mangaroska and Giannakos (2019); Gomez *et al.* (2016).

Nowadays, many active-learning methodologies, such as Process Oriented Guided Inquiry Learning Simonson (2019), Peer Instruction Porter *et al.* (2013), Peer-Led Team Learning Murphy *et al.* (2011), Problem-Based Learning Schmidt *et al.* (2009) and Team- Based Learning Michaelsen *et al.* (2004) are primarily used for the development of soft skills. One can find many project-based elements in curricula, but, in a normal university setting, it is very rare to have a project that is based around multidisciplinary teams, where only some of the members represent computer science and the rest represent non-technical domains Beckingham (2018); Berkling *et al.* (2019); Börstler and Hilburn (2015); Capretz and Ahmed (2010); Devedzic *et al.* (2018); Hazzan and Har-Shai (2014); Hazzan *et al.* (2014); Herrmann and McFarland (2019); Johns-Boast and Flint (2013); Johns-Boast *et al.* (2009); Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society (2013); Jones *et al.* (2016); Sukhoo *et al.* (2005); Tomić *et al.* (2017); Walker and Slotterbeck (2002); Nazligul *et al.* (2017); Katarina Pažur and Arbanas (2017); Ibanez *et al.* (2014); Ivanova *et al.* (2019); Pastel *et al.* (2015); Gestwicki and McNely (2016); Frezza *et al.* (2019); Saltz and Heckman (2018); Reardon and Tangney (2015).

In this paper, we present the MIMI methodology and its reference implementation in a project, called GGULIVRR@Lodz. This project has run annually since 2014. The project is led by the University of Lodz and involves several European partner universities. To solve the presented didactical issue one has to imagine a type of IT project that is interesting for both computer science students and non-technical students. The MIMI methodology requires that the project is multinational, intercultural and multidisciplinary. It allows computer science and non-technical members to take important roles. A solution lies in gamification, where many non-technical skills are in high demand

Almeida *et al.* (2015); Kapp (2012); Ivanova *et al.* (2019); Barata *et al.* (2013); Ibanez *et al.* (2014). Moreover, Wang (2011); Barata *et al.* (2013) shows that in game-based project students put more effort and seem to be more motivated to work. The approach involves teams of students who, in a short period, prepare a prototype of a mobile gamified app. Thus the presented approach also builds a spirit of entrepreneurship in the participants Milczarski *et al.* (2021, 2018); O'Reilly *et al.* (2015). As a result, the project fulfills the following educational objectives:

1. A cooperation in achieving goals in diversified teams; communication skills needed to achieve better final results; achieving objectives gradually by moving through different gradations of a problem.
2. Decision-making processes at the operational, tactical, and strategic levels; and presentation of achievements, proceedings, and results.

The project runs in Europe, where intercultural issues are very strong. Two people with the same level of English, but with different native languages and cultures, can encounter problems properly understanding each other. Due to different backgrounds, different people will interpret some sentences and constructions differently. Accent, pronunciation, and sentence construction are important in interpersonal communication. These language-related issues are difficult to teach. They are only solved through experience.

The acronym MIMI describes the elements that the methodology is based on. Each of the words in the underlining term plays an important role in the approach. We now discuss the meanings behind each of these words:

1. Multinational comes from the requirement to have a diverse group of students. Students from each nation bring a different perspective to the project. They often have a different method of approaching a given task. This diversity enforces synergy and communication. The method requires that all participating nations work equally in the project.
2. Intercultural is connected to synergy, empathy and communication. We want the students to bring elements of their culture to the project. This promotes a friendly and productive environment. It improves communication and empathy between the participants. In some cases, it can also result in the development of a feature of the gamified app. In these ways, the final solution will usually have elements from many cultures. This serves to enrich the experience of the participants while also improving the gamified apps that are produced.
3. Multidisciplinary is the requirement that each participant in the project should represent their different domain specialisms. This brings important effects, such as empathy and mutual understanding. It also helps students learn the skills of communication between specialists and non-specialists. The goal of the approach is communication between different groups of specialists rather than an interdisciplinary approach, where each student needs to fully understand all specializations.
4. Intensive comes about from the importance that the projects simulate a real IT company environment. Therefore, we need a goal-driven process and an ele-

ment of stress. The participants need to work under a strict time regime. Finally, in many didactic processes, a loss of focus or coherence in groups of students is often observed. The short time period used in the MIMI methodology negates this issue.

The paper is organized as follows. In Section 2 the GGULIVRR@Lodz project is discussed, which is the reference implementation of the MIMI methodology. Section 3 describes the proposed MIMI methodology in detail. Section 4 describes competencies development in the project. In Section 5 a comparison to the other didactic methodologies is provided. Section 6 presents an evaluation of the authors' approach and its implementations. The SWOT analysis of the methodology is presented in Section 7. Conclusions are drawn in the final Section.

2. Project Description

The GGULIVRR@Lodz project is organized by the Faculty of Physics and Applied Informatics at the University of Lodz, Poland. The project has run annually every year since 2014. The project runs in the middle of September and lasts between ten and fourteen days (including two days of travel). During the years of running the project, we have evolved and tested the MIMI methodology presented in this paper. Each edition of the project differed somehow from previous ones. The results of each year's project were assessed and applied to the next year's project, so as to continually refine the MIMI methodology. On that basis, the core elements of the implementation of the MIMI methodology have been identified as being: topic and venue of the project, team building, mentoring, technology used, and team management.

2.1. Partners and Participants

The GGULIVRR@Lodz project is multinational, intercultural and multidisciplinary in nature. It enables the inclusion of academics and students from various universities and countries as well as various fields of study to work together. It not only targets computer science students but also students from other disciplines, such as tourism, graphic design, business, history, and teacher training. To develop soft skills, the project requires the participation of academics and students from various disciplines. Students are assigned to teams in such a way as to ensure that each team is multidisciplinary and multinational. Academics from various STEM and non-technical disciplines provide high-level support to the student teams.

The universities that currently participate in the project cover Europe from north to south and east to west: Poland (University of Lodz), Belgium (AP Hogeschool Antwerpen), Finland (Centria Ammattikorkeakoulu), France (IUT de Lens – Université d'Artois), Ireland (Dundalk Institute of Technology), Portugal (Instituto Superior Politécnico Gaya), Slovenia (University of Maribor), and Ukraine (Precarpathian National University).

Each of the partner universities take approximately the same number of students to the project. The students can come from many disciplines. However, approximately half of the total number of students is from computer science departments. Over 60 students and 20 mentors participated in the 2019 edition of the project. Each of the universities is responsible for the recruitment of students specializing in the disciplines specific to that university. Some universities bring mainly technical students (e.g. University of Lodz, Dundalk Institute of Technology and Precarpathian National University), and some typically non-technical (e.g. AP Hogeschool Antwerpen, University of Maribor and Instituto Superior Politécnico Gaya). Some recruit technical and non-technical students like Centria Ammattikorkeakoulu and IUT de Lens – Université d’Artois. However, it is important that overall we end up with approximately half the students being from a computer science background and half being non-technical students. Generally speaking, a team will need a variety of skills in order to complete the project successfully. These include:

1. Technical skills – the team must build a mobile app.
2. Business assessment skills – the team has to be able to prepare at least a SWOT analysis and a Business Canvas model.
3. Presentation skills – the team will be expected to give three presentations during the project.
4. Communication skills – the team will be multinational, intercultural and multidisciplinary. The team will only succeed if its members can communicate well.
5. Leadership – the team needs a leader.
6. Graphical skills – the team should be able to create a basic graphical design.
7. Organization skills – the team needs to organize work between the members of the team.
8. Storytelling – a gamified app should have proper content. This is also true for presentations.
9. Entrepreneurship and creativity – the team should try to invent something unique with business potential.

The first two skills are the only ones that are directly connected to a study domain. The rest of the skills require soft skills, which are not connected to a specific study domain.

The didactical goals of the MIMI refer to the factors that IT companies use to recruit graduate employees. Therefore, companies have a very important role in the project. They support the project both financially and substantively. Additionally, external partners provide technical and non-technical advisors and are involved in the commercial development of some of the student projects.

2.2. Project Theme and Student Teams

In the project, students are divided into teams that contain between five and seven members. Teams have the task of creating a prototype of a gamified app. In order to make the project more real-world focused, the students must be asked to develop a prototype of

a gamified app that relates to a theme that is provided by an external stake-holder, such as a company or public organization. For our project, the theme of the project has always been connected to the City of Lodz. Themes have focused on the city's tourist attractions and culture, on the city's science museum and on the city's academic offerings.

The main rule during the formation of student teams is that each team should have at most one representative of a given country/university (ensuring the multinational and intercultural elements). Some members of each team must be computer science students (typically three), with the rest of the team consisting of a combination of other disciplines (ensuring a multidisciplinary element). Being multidisciplinary makes the teams more representative of real-world company project teams.

In their teams, computer science students' main responsibilities are connected with technical issues, while the non-technical students have tasks directed toward concept, content, business analysis and preparations for presentations. The division of labor is not strict and computer science students participate in brainstorming and presentations. Non-technical students take part in development, usually with user interface design and gamified app functionality.

Several years of observations show that the lack of integration, especially at the beginning of the project, can cause poor team collaboration. Therefore, the first day of our project always ends with a student integration activity. For our project, this is usually a game of bowling. This allows and encourages all the students who are involved in the project to move freely and easily introduce themselves to each other. It is also crucial that all students (including local students) live together in one facility and eat meals together. This encourages better communication within teams and between all teams. It also encourages all students to work more efficiently and effectively.

During the project, a mentor is assigned to each team. We have experimented with a few approaches: singular, multiple and no specific mentor at all (where all mentors help all teams). We have identified that a single mentor per team works best. The mentor is there to help the team to stay focused on the tasks of the project. The mentor can also help the team deal with any other issues that affect the team, such as resolving communication, members' workload or creative differences.

The student teams' work is presented on the last day in the form of a public final presentation. Representatives from partner companies, the University of Lodz and the city of Lodz are invited to attend this presentation.

2.3. *Team-building*

Team-building takes place on the first day of the project. During the opening meeting, participants are informed about the theme of the event. The students are given some basic organizational details, such as the project rules and daily schedule. The students are tasked with building their own multidisciplinary teams. Teams are not allowed to have two students from the same participating university unless this is impossible (a country takes more students than there are teams). Furthermore, between them, teams must include both computer science and non-technical students.

The team-building process is strengthened by the inclusion of activities that keep all of the students together and allow the free movement of students within the entire student group. In the case of our project, a variety of such activities – such as city games and bowling – are provided. Based on our experience, the quality of the team-building process has a significant impact on team integration and the quality of the teams.

On the first day, we enforce the introduction of the participants to each other. We organize activities in two parts: firstly, during the morning and afternoon, we have an introduction and integration session for all participants – secondly, during the evening, we enforce team formation and team level integration. During the introduction and integration session, the participants present their competencies. We moderate integration using methods of social integration games and workshops. The workshops are usually organized by project partner companies. Afterward, we allow participants to freely discuss team formations among themselves. The team formation and team level takes place in the evening. It takes the form of group-based fun activities. Here, it is ideal that students can play in a team, while also being able to move between teams and talk. Many activities are unsuitable for our purposes, e.g. activities in a swimming pool and most outdoor activities. We suggest sports such as bowling, snooker/pool or darts, et cetera.

At the end of this process, each student must be in a team. Furthermore, each team needs to have a complementary set of competencies, while also satisfying any other rules that we have imposed (e.g. only one representative of a given university per team).

To facilitate this, we impose the rule that until all teams are accepted, none are accepted. This ensures that all of the students work together to ensure that all students are placed on a team that satisfies the rules. This prevents the situation of having one or two people who have no team and the difficulties arising from this.

2.4. Technological Stack

In our approach, we require that the gamified apps work on mobile devices. Over the various editions of the project, we have applied different approaches to technology. These have ranged from imposing specific technologies to letting students select their own. The latter was a response to students' suggestion that they would produce a better product using technologies they already used.

However, this solution did not usually work well. Often, a student overstated their knowledge in a particular technology resulting in the team failing to complete their application as the student could not give the necessary technical support to the other computer science team members. Thus, the project organizers have decided that only the set of technologies that are connected to the competencies of the academics and support team can be used. Based on our computer science students' technical skills, the consortium has favored Cordova and Ionic technologies. Workshops in these technologies have sometimes been provided by the project's partner companies.

2.5. Organizational Constraints

The project has to be organized with access to a few rooms that are located close to each other. Ideally, if there is a big room, we can organize meetings for all participants in this one location. Each day, all participants meet at least once. This project space can also be used for group presentations and should be equipped with a projector. The teams can work in smaller rooms. If there are no small rooms available, an auditorium, gym, or long corridor can be appropriately arranged to allow teams to work. All teams should be equipped with computers, and we suggest the participants use private laptops.

The costs of the organization are non-negligible and vary on the level of support provided. Each foreign participant has to travel to the venue. Partner organizations or participants sometimes cover this cost. The next important issue in the project budget is accommodation¹. This is the most substantial part of the project costs and can be covered in a few approaches. The support may be organized from external institutions, such as the EU, country, or local administration. It can also be covered by the organizer if the institution owns dormitories. The partners or participants can also cover such expenses. In the case of lack of financial support, the foreign participants can be hosted, free of charge, in local participants' homes. Such an approach is used, for example, in project e.COAL organized by IUT de Lens – Université d'Artois, France.

The last type of financial cost relates to food expenses² and cultural events, such as the project integration event, final presentations, farewell party, et cetera. The cost of such events depends on the level of support and can be fully or partially covered by participants. The cost structure of GGULIVRR@Lodz event in the year 2019 is shown in Table 1.

The most important resource needed to organize the event is the team of mentors. The mentors usually are partner organizations' employees, and the partners cover costs connected with their participation.

Table 1
Cost structure in GGULIVRR@Lodz project in the year 2019, €/dp denotes cost in euro per day per participant while €/p denote cost in euro per participant

Cost type	Cost	Source of funds
Travel	—	Participants or their home institutions. Cost depends on partner, distance and means of transport.
Accommodation	10 €/dp	The Organizer, University of Lodz, in dormitories during summer break when dormitories are partially empty.
Food expenses	10 €/dp	The funds are obtained from City of Lodz. The funds cover lunches and breakfasts for participants.
Cultural events	70 €/p	The funds are obtained from sponsoring IT companies. This allowed to organize integration, final presentations, and farewell events.

¹ The cost of accommodation is estimated at around 30 € per day per participant in most of European countries.

² The cost of food is estimated at around 12 € per day per participant in most of European countries.

3. Methodology

A detailed description of the project workflow is presented below. This will allow others to recreate and apply it in their own project.

The student teams' main task is to create a gamified app idea, a business and marketing plan and a working prototype. Since 2014, we have used a continuous self-improvement process to hone in on the key factors that achieve a high-quality project outcome. The project schedule must include several unchangeable key actions and a changeable general workflow. The project's key actions, general workflow and expected results are presented in Tables 2 and 3.

The MIMI methodology relies heavily on student team creation, which is described in section 2.3 and the initial preparations for the project (i.e. learning about the topic of the event, the outcomes and its venue). Getting this right will result in team formation that is relaxed and student-led.

3.1. Preparation

Prior to the start of the project, we take a few steps to ensure the project runs smoothly. Items such as the general schedule, dates, and the length of the project play a key factor in the success of the teams and need to be discussed by the project partners. The duration of the project is discussed and agreed upon. Currently, the project lasts eight full days, starting Thursday morning and finishing the following Thursday evening. This includes a free Sunday to separate the conceptual phase from the production and implementation phase. Specifically, the following items and their timings are agreed in the preparation steps.

Table 2
Key elements of the daily schedule for a MIMI project

Time	Activity
08:00	Common breakfast
08:45	Mentors' meeting
09:00	General meeting of all mentors and students
09:15	Each team meets with their mentor
09:30	Students work in teams or attend workshops
11:30	Each team meets with their mentor
13:00	Lunch
14:00	General meeting of all mentors and students
14:15	Each team meets with their mentor or/and work in teams
16:00	Mentors' meeting, each team meets with their mentor – summary of the tasks
17:00	Teams work on their own; leisure time

The project's starting date is agreed upon. In our case, it takes place in September. The invitations along with the project topic and rules are sent in February/March five months prior to the start of the project (Month -5 in Table 3).

In the partner universities, the application process is finished by July. The students fill in an electronic form highlighting their skills (Month -3 in Table 3).

The core of the daily schedule is the same each day and it focuses students on specific tasks, so they are not diverted from the main goals of the project (Table 2). It is published on the project webpage around August (Month -2 in Table 3). The daily schedule is discussed in detail in section 3.2.

A table outlining each student's skills is disseminated prior to the start of the project (Month -1 in Table 3).

After each project, we gather data from the mentors, business partners and students. We analyze the data to help us make improvements for future iterations of the project. The draft of the prospective projects and preparation for future development is presented in section 3.4.

Table 3
Workflow in the MIMI implementation

Day/ Mth	Main activities	Students main tasks	Mentors main tasks	Deliverables Milestones
Mth -5	Topic, venue and rules	Read the documents, acknowledge the rules	Present the project idea	
Mth -3	Applications for the project	Enroll to the project	Students selection	List of students
Mth -2	Initial schedule	Get acquainted with the schedule	Get acquainted with the schedule	
Mth -1	Skills sharing	Show their skills in the spreadsheet	Show their skills	List of skills
Day 1	Opening, the summary of the rules, project idea and venue; Profiled workshops; Integration event	Attend workshops; Look around for the group members; Create the group	Divide the students into: creative, management and technological groups; Initial group assessment	Group creation rules; Group division
Day 2	Group confirmation; Working on the game ideas; Workshops (soft-, creative-, technological-, management-skills)	Working on the game ideas, preparation of the short presentation; Self-assessment of the ideas after presentations	Mentors and the groups work on the projects ideas; Assessment of the game ideas and their improvements; Selections of the mentors	1st presentation of game ideas; Initial group ideas; Mentor selection
Day 3	Clarifying the game idea; Divisions of the work	Preparing the draft of the game scenario; Choosing the task to develop and the technology; Appointing the roles to the group members; Preparation of work schedule	Conducting the group works on the project; Preparation of the tasks for the groups; Mentoring the given group and their progress; To be accessible to other groups if needed	Technological stack; Clear game idea; Draft of the game and developed task; Detailed group schedule

Continued on next page

Table 3 – continued from previous page

Day/ Mth	Main activities	Students main tasks	Mentors main tasks	Deliverables Milestones
Day 4	Free day; No schedule	Free time	Free time	Relax
Day 5	Preparing content; Working on the business aspects; Prototyping the chosen task	Working on their personal or subgroup tasks; Clarifying the game idea; Working on the scenarios, scoreboard, gaming rules e.g. points; Preparing group adverbs	Mentoring the groups; Checking progress; Making students to avoid deflections from the main task; Preparing the feedback questionnaires	The game idea is thought over and fleshed out; Feedback questionnaires
Day 6	Working on the game details: business, monetization, playability, preparing presentations etc.	Preparing content; Working on the business aspects; Prototyping the chosen task; Practicing presentation; Polishing the content and adverbs; Testing the prototype	Mentoring the groups; Checking the progress of the tasks; Assessment of the 2nd presentations	2nd presentation of games; Business; Monetization idea; Content; gamified app layout design; Main task prototype
Day 7	Preparing the prototypes and final presentations	Preparing the prototype for the afternoon open testing and final presentations	Checking the progress; Presentation rehearsal	Mobile app; Working prototypes of games
Day 8	Final presentation; Project and idea assessment; Farewell party	The final presentation; Self-assessment; Uploading sources codes, presentations, videos, etc.; Filling in the questionnaires	Rehearsal of the final presentations with team's mentor; Assessment of the final presentation; Collecting feedbacks	Final presentation; Source codes, presentations, videos, gamified app layout design
Mth +1	Appointing the games/apps for the further development	Prepare a detailed description of the gamified app scenarios and tasks	National and group mentors coordinate the work	Scenarios for the future development
Mth >1	Development of the game/app	Create level ideas, testing	Testing	Final game or gamified app

3.2. Workflow During Project

Each day starts with a common breakfast, mentors' meeting and an obligatory general meeting of all mentors and students. This is followed by a meeting of each team with its mentor. During the project, students are given a small number of technical and non-technical workshops relating to technology, business and game design. Not all students need to attend every workshop. When not attending workshops, students work in their teams. Next comes a common lunch followed by another general meeting of all mentors and students. After this, the students return to working in teams. During the evenings, students tend to do some work on their projects and to socialize with the other students from all of the teams.

We propose that teams work with the use of an incremental methodology or one of the agile approaches. Each agile iteration usually lasts one day. During each day, students are assigned tasks to achieve and are told of the expected results and deliverables appropriate to the milestones of that day (Table 3).

On the first day of the project, before the project opening ceremony, we have a brief welcoming mentors' meeting. Here, we present mentors to each other, summarize our skills and repeat the rules of the project. During the opening session, we summarize the rules to students, briefly present the main project theme and goal. We explain the key points of the day in detail. The mentors describe their domain of expertise to the students.

After the first day's lunch and the workshops, we organize some common activities, so the students learn more about each other and try to build their teams. On the first evening, we have an integration and team-building event. This lasts three to four hours and consists of a relaxed team activity, such as bowling. During this activity, students must create their teams according to the rules provided in the morning session. Importantly, every team must contain members who have IT skills and others who have non-technical skills. The proposed teams are assessed by the mentor council (MC) carefully. The first day's milestone is the creation of the teams. The first day also helps students to develop their communication skills.

At the general meeting of all mentors and students on the second day, the students are informed if their teams have been validated by the mentors. If teams are not approved, then the mentors will assign them. Once the teams are approved, student teams can work on their gamified app. All the mentors are accessible to all teams. The mentors usually circulate on their own between the teams. On the second day, students participate in soft skills and technological workshops and deliver a short (three-five minutes) presentation, where they explain their initial gamified app idea, show their team skills and describe their team task division. The computer science students choose a technology. If they want to use a technology that is not supported by the academics or workshops, this must be granted by the MC. Acceptance or rejection of the student's technology proposal is reported to students after the mentors' meeting. After lunch, students can prepare for their presentations. At the presentation session, one mentor plays the role of a moderator.

In the first presentation, the teams outline their initial idea. The students present their main storyline, the target team, and the initial work division of the team members. Each presentation is video-recorded. This recording is only used to help the teams assess their own presentation performance. After each presentation, the mentors and other students ask questions to help clarify a student team's idea. After all the presentations are done, there is a mentors' meeting to discuss and assess the ideas. Feedback from the mentors' meeting will be given to the teams. In rare cases, the mentors' meeting will recommend that a student team completely change their idea. During this mentors' meeting, there is also a pitch of mentors for the teams. We usually rely on our previous experience in team mentoring in order to appoint an appropriate mentor for any team that contains strong personalities. In parallel to the mentors' meeting, the students are asked to discuss their

ideas and to self-assess their work. The second day's milestones are the outline of the game idea and mentor assignment to each team. The second day also helps students to develop their presentation skills.

The third day starts as usual (see Table 2). Students meet in teams with their newly appointed mentors to assess their idea and divide it into tasks. Tasks include prototyping, deciding on game app layout design and content, discussing business and marketing issues, researching target markets, finding ways to monetize the gamified app, et cetera. The students prepare the electronic schedules of their tasks, taking into account the current gamified app idea. After lunch, they present the schedule of tasks to their mentor and a final version of their tasks is agreed upon. The schedule of their project and final technological stack are the milestones of the day's iteration. The students now have one and a half days free to socialize and to explore the host city. However, many students tend to spend some of this time preparing part of their work for the fifth day. All three days (days three to five), as well the following days help students to develop teamwork, problem-solving, and communication skills.

The fourth day is free. Students are encouraged to take time out to be tourists in the host city and to relax.

On the fifth day, students and mentors follow the standard routine (see Table 2). The teams prepare content and app layout design. They produce a business plan and detailed information how they can monetize parts of their gamified app. Students do surveys for their target groups and work on the style of their presentation (including video, graphics, and audio). Teams often meet with their mentor to assess the team's progress. Mentors circulate to help any team that might need some help in a particular mentor's domain of expertise. The fifth day's milestone is an improvement of the working gamified app, the layout for the 2nd presentation, content, business and monetization assessment, and the marketing research of the target group.

On the sixth day, students work on their game tasks. They practice with mentors for their second presentation. The ten-minute presentations give each team time to explain their ideas thoroughly. The session moderator allows a five-minute Q&A session for mentors, students, and invited external experts. The presentations are recorded so that teams can review and self-assess their performance. During the sixth day, students usually try to finish their prototypes for the next day's testing. The sixth day's milestone is improvements of the prototype, business and monetization models and presentation extras, such as video, audio, et cetera.

On the seventh day, the technical students work to finish their prototypes. The creative team members work on the final presentation. In the late afternoon, there is a session where the various teams' prototypes are tested by the mentors. The IT mentors choose the best prototypes, taking into account only the technical quality of the prototypes. The non-technical mentors make a selection of the best prototypes using non-technical criteria (such as business plan, gamified app layout, content and creativity). In the late evening, pizza or similar "student food" is provided for a hackathon night. This event allows students to improve their prototypes and the final presentation. The seventh day's milestone is a gamified app prototype. The seventh day also helps students to de-

velop their presentation skills. By the end of this day, the mentors should have prepared the project's final survey.

The eighth and final day starts with a short general meeting of all mentors and students. Here, the mentors provide some last encouragement to the students. After this, the students practice their presentations with their mentor or students work in teams. The main event of that day usually starts at 11:00 AM. Students check all of the needed connections in the presentation auditorium and solve any issues with the technicians. Representatives of the host university, the City of Lodz and partner companies are invited to the final presentation. The moderator for the final presentation is usually one of the non-mentoring academics. The moderator will try to target the asking of questions toward the external guests. The mentors write down their remarks for future assessment. The recorded presentation plus questions and feedback usually lasts approximately 12–15 minutes per team. The external experts assess each student team's project during their presentations.

After the presentations, there is a lunch for all of the people who were at the presentation. Here, the students, mentors and external guests have time to talk freely among themselves. This allows students to get additional feedback on their presented solutions. After that, the final general meeting of all mentors and students is held. A student survey is conducted. The student teams upload the final source code, presentations, and other documents to preprepared repositories. In the evening, all the participants and guests take part in the farewell party.

The schedule presented above allows us to increase students' soft and communication skills. The process of developing a gamified app in short sprints by a team (work, present obtained results, repeat), facilitates the participants working in a multidisciplinary team to achieve a defined goal.

3.3. *Mentoring*

A crucial component of the project is to provide all teams with adequate mentoring. The goal of mentoring is to ensure project participants get support in two areas: those related to the technologies used and, more importantly, to the development of the project idea. The mentor's task is to supervise the course of work, project development, and cooperation in the team. A mentor is the person to whom a student or team can turn to when they encounter problems. Problems are usually related to technical, communication or organizational issues. The mentor's task is to support the team, indicate the possible directions of development, correct unnecessary or dangerous shortcomings, and help solve various problems. The mentors should not impose their ideas or solutions on the team. He/she is not expected to know any particular field, but should be open to new ideas and be enthusiastic about their team and the project. The mentors should create a partnership with their team that promotes student inspiration and stimulation. They should also help the team to discover and develop the team's capabilities. A mentor should not solve problems directly, but help the team to solve issues

themselves. Sometimes, a mentor might even allow a team to make a mistake, so that they can learn from this.

Each university proposes one or two lecturers to participate in the project. The team of mentors consists of these lecturers. The team of mentors is diverse in terms of competence. This is due to the assumption that the project is multidisciplinary. This feature is ensured by the participation of students of various faculties, both technical and non-technical. Each university participating in the project has a different specificity, thanks to which mentors coming from these universities have a diverse and wide range of competencies. Having a diverse team of mentors ensures that we can help teams overcome the various technical and non-technical difficulties that they might encounter. The project assumes that each team has one mentor who looks after it throughout the project. However, this does not exclude the possibility of using the assistance of another mentor, if their competencies could help solve a team's problem. For example, if a team led by a nontechnical mentor has a technical problem, they may seek the assistance of another mentor with appropriate competencies. In this way, the mentors' competencies complement each other, providing the teams with extensive support.

Cooperation between a team and their mentor begins by assigning mentors to the teams after the first presentation. This is always preceded by a discussion among the team of mentors. Each mentor chooses a project that they are interested in. The team does not influence the choice of mentor. Throughout the project, the mentor remains at the team's disposal. Each day begins with a meeting between a team and their mentor to discuss any problems that arose during the previous day's work and the plan for the current day. In addition to the morning meeting, during the day at least one more meeting with the mentor is scheduled. During this meeting, the mentor has the opportunity to check the progress of work and can discuss any emerging ideas and ways to implement them with the team. Although the mentor is at their disposal, most of the time students will work independently of their mentor. In addition to the designated meetings, the team may ask their or another mentor for consultation at any time. The mentor also helps the team prepare the final presentation. Students often do not have experience in presenting their achievements. Their mentor helps them extract and show the strengths of their project.

Different mentors have different competencies. Each mentor has a different character, uses distinct methods of work and motivation and has a varied view on a given problem. Each project team also encounters different difficulties, develops its methods of work and communication and has different requirements for support. For this reason, it is very difficult to assess the impact of mentoring on the final effects of the project. It is influenced by various, often impossible to measure, factors. That said, surveys have shown that the students view the mentors' role as critical to the success of their projects.

Several years of experience shows that all teams, regardless of which mentor they work with, achieve the intended outcomes. We attribute this partially to the extensive teaching sophistication of mentors as they are academic teachers with many years of experience.

3.4. *Post-project Final Product Development*

The potential that their prototype will be further developed into a commercial product is one of the incentives for students to work hard at creating a real-world solution. After the final presentation, some projects may be chosen for further development. The choice of the projects takes into account: the gamified app idea, their business impact, monetization, complexity, et cetera. The supporting technological partners are usually the ones who take the project to fruition, but any university partner or student team can also continue with their project.

In 2014 we had two business and university consortiums for commercial app development: “Flappy Bear” for the Se-ma-for and “Urban Legend” for Urban Forms NGO. It took one year to finish the development. Se-ma-for and Urban Forms were the product owners and the developer teams consisted of Irish and Polish students and mentors.

The other examples are the games “Unknown” and “bTrail”. A games development company from Lodz, called 9bits Development Studio, produced commercial products based on these two games. So far we have finished the development of six games. Some projects, such as “bTrail”, are very complex and it would be hard to develop them without extra commercial resources and maintenance. That is why these projects were postponed until a supportive technological company was found.

4. **Competencies Development in the Project**

During the project, participants develop various competencies. We can divide these competencies in two areas: a domain of the study competencies and soft skills.

The domain competencies are mostly connected to the participants’ field of study. Every student can develop appropriate domain competencies during the project. Computer science students need to use their development skills to build the team’s prototype gamified app. Business students use their skills in business assessment, management. The same can be said about other fields of study.

This paper is focused mainly on the soft skills that are developed during every phase of the project. The project uses the Multinational, Intercultural, Multidisciplinary, and Intensive (MIMI) methodology – this enforces the development of communication skills in the participants. Specialists often use a specific domain language that is incomprehensible to others. For example, IT-specific words, such as factory method, JSON data, test coverage, et cetera. The project enforces empathy in the communication, as people have to translate domain-specific terms in normal, lay-person, language. Moreover, teamwork on the common goal is almost impossible without cooperation and task planning. As the teams in the project tend to achieve a high-quality result, we can deduce that the teamwork phase is conducted properly.

We can define the main elements of the project as: teamwork, idea creation, idea evolution, presentation, communication within a team, focusing on the goal. For these

activities, we show a competencies matrix (Table 4), which shows the soft skills that are developed by each activity.

We do not evaluate the soft skills of participants before and after the project. Nevertheless, on the basis of the teams' final presentations and prepared prototypes, we can conclude that high-quality results could not be possible if teams did not have a high level of soft competencies. These soft competencies were heavily used during the project (Table 4, Fig. 1). Thus, we can be confident that the students improved the required soft skills while participating in the project.

Table 4
Soft competencies matrix in MIMI

Skills	Activities					
	Teamwork	Idea Creation	Idea evolution	Presentation	Communication within a team	Focusing on the goal
Active listening			X		X	
Analysis		X	X			X
Brainstorming		X				
Clarity		X	X	X	X	
Conflict management	X	X				X
Constructive feedback						X
Cooperation	X		X		X	
Critical observation		X				X
Cultural intelligence		X		X	X	
Curiosity			X			
Empathy	X	X		X	X	
Idea exchange	X		X	X		
Imagination	X	X		X		
Initiative	X			X		
Innovation		X				
Logical reasoning			X		X	X
Non-verbal communication				X	X	
Open-mindedness	X	X	X		X	X
Optimism			X	X		X
Planning	X					X
Prioritizing	X		X			
Public speaking				X		
Self-confidence				X	X	X
Stress management	X					X
Verbal communication					X	

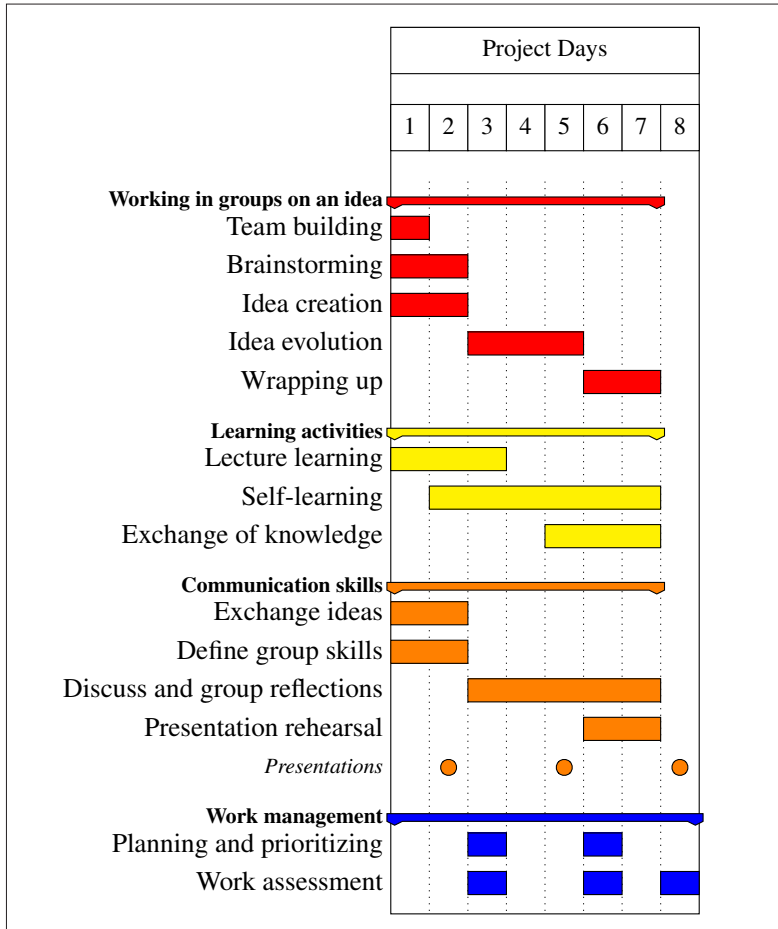


Fig. 1. Development of selected soft-skills in different phases of the project.

5. Comparison to Related Didactic Methodologies

Using innovative methods, such as creating games during the teaching process, should help students become deep learners instead of just surface learners. Furthermore, producing gamified apps for authentic, real-world, problems and using a team-based learning methodology encourages peer interaction as students learn.

According to Chickering and Gamson (1987); Woods (2013); Gomez *et al.* (2016); Barata *et al.* (2013), there are several fundamental ways to improve student learning. These include the use of active learning, cooperative learning, time-on-task, prompt feedback, delivery of student success, catering to different learning styles, and quality interaction between teacher and learner. Some other options include knowing the names of students, motivating and empowering students with the learning process and with their assessments.

The usual activities in the teaching and learning process include:

1. Picking a problem.
2. Identifying the learning issues.
3. Setting learning goals/criteria or learning objectives (LO).
4. Planning and using a strategy.
5. Picking resources from which to learn.
6. Sharing information or giving lectures.
7. Learning.
8. Solving the problem.
9. Creating the assessment.
10. Doing the assessment.
11. Embedding new knowledge in previous knowledge by elaboration.
12. Reflecting on the process.

The major learning outcomes from various team-based learning methodologies vary because of the emphasis on what students acquire by using that methodology. Major outcomes include:

- (a) Expanding/increasing soft skills e.g. communication, brain-storming.
- (b) Acquiring new knowledge.
- (c) Acquiring new process skills, such as critical thinking or design. Although students might acquire some new knowledge, the prime purpose is to develop a specific skill.
- (d) Equal knowledge and processing skills learned.
- (e) Uncovering and correcting misconceptions and developing deep learning.

Many different learning approaches can be used to help students to gain the above outcomes. For example, product-based learning, problem-oriented learning, problem-based learning, project-based learning, process-oriented guided inquiry, peer lead team learning and model-eliciting activities. Different didactic methodologies support students to different degrees and may have different outcomes. In the list below we show some of the different learning approaches, the learning activities (LA) that they strengthen and the major learning outcomes that they result in:

1. Problem-assisted (lecture followed by practical experience) [activities 7, 8; outcome b].
2. Problem-driven research/inquiry without lectures (problem posed, plan given, students follow the plan and solve issue) [activities 7, 8, 11, 12; outcomes a, d].
3. Problem-driven inquiry/research (problem posed, lecture, then inquiry/research) [activities 2, 3, 5, 7, 8; outcomes a, d].
4. Problem-driven inquiry with small team and skill focus (POGIL process-oriented guided inquiry learning: problem posed, no lectures, small team facilitated for deep learning Simonson (2019)) [activities 2, 3, 5, 7, 8, 11, 12; outcomes a, e].
5. Problem-centered discovery (identify past experience/ problem Kolb and Kolb (2005)) [activities 1, 2, 3, 5, 7, 8, 9, 10, 11, 12; outcomes a, e].
6. Problem-initiated with students generating learning objectives (problem posed, students identify learning needs and objectives, lecture, solve problem) [activities 2, 3, 7, 8; outcomes a, b].

7. Problem driven action learning (student poses a problem, small team asks questions and reacts to improve individual's learning and answer own problem) [activities 1, 2, 3, 11, 12; outcomes a, d].
8. Problem-based learning given learning objectives (pose problem, give objectives, students research, teach, discuss solve, react) [activities 4, 5, 6, 7, 8, 9, 10, 11, 12; outcomes a, c].
9. Problem-based knowledge and skills focus scaffold (knowledge in single subject and skills to use knowledge, Schmidt *et al.* (2009) type II; process skill prerequisite; scaffold additional process skill development) [activities 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12; outcomes a, c].
10. Project-led problem-based learning. PL-PBL Open-ended trigger posed to students, they create proposals, obtain approval for work breakdown schedule, goals, budget. Then the PBL phase, where students identify what they need to know, learn, and teach each other. Then, return to the project and complete the work schedule. [activities 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11; outcomes a, c].
11. Problem-based multidisciplinary knowledge in integrated subjects and skills focus Savin-Baden (2007, 2005) (students make the connections of interrelationships among subjects) [activities 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12; outcomes a, c].
12. Problem-based multidisciplinary learning with knowledge, skills, and attitudes Savin-Baden (2007, 2005) (the boundaries between subjects exist but are somewhat arbitrary; encourage deep learning independent of subject "discipline"; critically think about knowledge, themselves, and peers Savin-Baden (2005)) [activities 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12; outcomes a, c].
13. Task-based (problems solved in real-time in a setting Ozkan *et al.* (2006); Harden *et al.* (2000)) [activities 5, 6, 7, 8, 9, 10, 11; outcomes a, d].

Taking into account the presented above examples of the didactic methodologies we can derive that Multinational, Intercultural, Multidisciplinary & Intensive (MIMI) is a team-based one (mentors give objectives, students research, discuss, solve, present and create a prototype) [activities 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12; outcomes a, b, c, d, e].

The MIMI is multidisciplinary and intensive and allows us to achieve our defined goals. From the above activities, we are interested in the ones that require the involvement of not only students and mentors, but also of experts and the project organizers. The student, mentor, organizer and expert activities are summarized in Table 5 and are described thoroughly in Sections 2 and 3.

Comparing MIMI to other known didactical methodologies, we can show similarities and differences. The MIMI methodology uses as its scaffolding Cooperative Problem-Based Learning (see Section 2 and 3) and Challenge-Based Learning or Scaffolded PBL. PBL gives an authentic challenge problem. Students work in teams of five to seven and each team works one to two weeks on a challenging problem. Students, guided by mentors and experts, are encouraged to share information across teams. During the GGULIVRR@Lodz project, short workshops are given when students realize they need more understanding of the required skills.

Table 5
 Participation in the learning activity by
 students (S), mentors (M), organizers (O), and experts (E)

ID	Learning activities	S	M	O	E
1	Pick problem	x	x	x	
2	Identify the learning issues		x	x	
3	Set learning goals/criteria or learning objectives, LO		x	x	
4	Plan and use a strategy	x	x		
5	Pick resources from which to learn	x	x	x	
6	Share information or give lectures	x	x	x	x
7	Learn	x			
8	Solve the problem	x			
9	Create the assessment		x	x	x
10	Do the assessment	x	x	x	x
11	Embed the new knowledge in previous knowledge by elaboration	-	-	-	-
12	Reflect on the process		x	x	x

6. Evaluation of MIMI Methodology Implementation During Years 2014–2019

Assessment of the soft-skill, didactic and technological results of the project is conducted in a threefold way: questionnaires for the students and mentors, surveys for the external companies and non-binding, casual atmosphere meetings/talking with companies, mentors and students. The surveys and questionnaires are revised and improved each year. The electronic version of the questionnaire is made accessible after the final presentation session. The external partners from the companies are asked to grade each project at three levels: the gamified app idea and its presentation, business and marketing impact, and proficiency of the technical solution. General remarks about the projects and overall impression are also documented. After the final presentations, there is a meeting where the organizers and mentors discuss the project’s outcomes.

Ongoing assessment of the project is conducted during the mentors’ meetings that happen each day. The notes and minutes are written down during each day’s mentors’ meeting and general meeting of all mentors and students are analyzed after the project to aid in the development of the project in the following years. If needed, the rules and other documents are also clarified and updated.

Surveys show that the project is very stimulating and interesting to the students but, above all, very different from their previous teaching experience. Most students have reported that they see the great benefit offered in working with multinational, intercultural and multidisciplinary teams.

In the last three editions (2017, 2018, 2019) of GGULIVRR@Lodz, 159 students completed the anonymous surveys. The students have been asked to point whether they come from computer science and non-technical departments. As it was described

in Section 2, approximately half of the students are from the computer science discipline. The summary of the results is presented in Fig. 2. The results confirm that 100% of students said that active learning is a good way of learning and 95% of students responded that active learning allows them to learn more than traditional teaching allows. Most students (85%) declared that they prefer teamwork rather than individual work. More than 92% of the students declared improvement in their presentation skills.

Interesting conclusions can be drawn from the answer to the open question “What did you like the most in the GGULIVRR@Lodz project?”. The most common answers are: the ability to improve communication skills (13%), working with people from different countries (15%) and working in teams (20%). The most common answer is working with people (46%).

The workshops were highly rated by participants. However, only 5% of students listed this as something that they liked most in the project. This suggests that project elements similar to traditional forms of teaching are not as valuable to participants as the opportunity to work in a very diverse team of people.

Participants also positively assessed the open formula of the project. The project assumes that we do not impose too rigid constraints on participants. We only define the general theme, leaving its interpretation to individual teams. This approach is conducive to creativity, as evidenced by really interesting and fresh ideas for various applications. 93% of surveyed students said that they liked this form of the project and 75% said that it was more valuable than working on a defined project.

Fig. 2 below shows the percentage of students who gave positive answers to the following seven questions:

Q1. The project motivated me to learn something new and to go deeper with my knowledge.

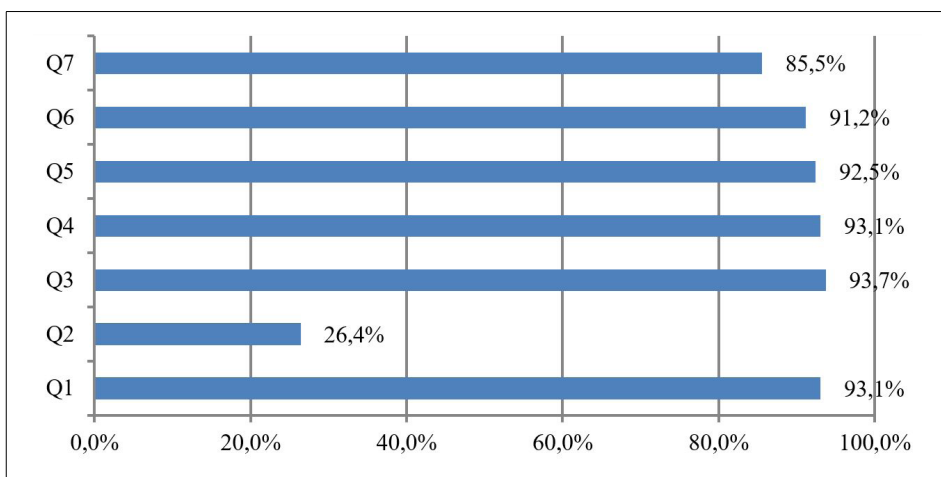


Fig. 2. Percentage of positive answers to Q1–Q7.

- Q2. Working in a team with students from different studies was very difficult.
- Q3. I felt that I really learned how to collaborate with students from different studies.
- Q4. Working in a team, during the project, really motivated me.
- Q5. My presentation skills improved.
- Q6. My self-confidence improved as a result of taking part in this project.
- Q7. As a result of participating, I feel I am more likely to seek out team projects over individual projects.

The industrial partners' questionnaire results show that gamified app prototypes that are developed during the GGULIVRR@Lodz project have more than just academic value. Most gamified apps were highly rated by the industrial partners, as expressed in their answers to the separate business survey. Those surveys concerned three aspects: interesting presentation, promising game idea and overall impression. Most teams were rated well and a few teams were rated highly by the industrial partners.

During the project, we observe the increase of soft competencies of the student participants. The intense communication, goal-driven teamwork has a significant impact on the students. Student teams are observed solving their problems better with each passing day. Mentors observe an increase in the quality of presentations and a growth in students' general communication skills.

The framework of the project requires flexibility. Based on presentation feedback or feedback from their mentor, students sometimes radically modify their project idea.

The mentors have observed that, after the project, the participants display more active and focused behavior in the following academic years, when compared to students that did not take part in the project.

All this leads to the conclusion that there was an increased level of soft skills and self-esteem for participating students.

7. SWOT Assessment of the MIMI Methodology

The proposed MIMI methodology is heavily reliant on the mentors' competencies and skills to manage the teams, regardless of their expertise domain. Even first time mentors can easily succeed with the help of the mentors' board. The SWOT analysis of the MIMI methodology is presented in Table 6.

It can be derived from the SWOT of the MIMI that the MIMI methodology and the mentors' competencies results in a more effective way of managing the students' development of the gamified apps. Additional reflections are as follows:

1. We have optimized the length of the project to 10 days: 2 days for travel, 7 working days and one day (one and a half days) free in the middle of the project. Thursday to Thursday seems to us as the best one.
2. The break in the middle has a positive impact on the group ideas. Students need some time for retrospection and thinking over the game idea. They also have time to learn the venue they are usually working about.

Table 6
SWOT analysis of the MIMI methodology

S	<p>The rules are known in advance; A well balanced and diverse team of computer science and non-technical students; The workflow has a clear and set backbone; Widely used technology is known in advance; Complementary competencies of mentors; First-time mentors can easily apply it; Small teams increase impact on each participant.</p>	<p>The mentor should support not drive the team; Students English should be satisfactory; High dependence on student and mentor diversity; No time to recover after bad team's choices; Risks of over-confidence in strong personalities; Non-negligible organization cost; Due to the small team sizes, it would be difficult to offer the project to every student on a degree and their work.</p>	W
O	<p>IT labor market interested in soft skills; Students look for active education; Universities and staff implement new approaches eagerly.</p>	<p>Random events (sickness, accidents, etc.); Dependence on provided workshops; Requirement of proper place (accommodation and work).</p>	T

3. The presented gamified apps often have a commercial and monetization potential. Their prototypes achieve the proper technological level.
4. The technological stack and supporting materials must be defined in advance. It helps the technological students to prepare for the intensive development of their tasks.
5. The students can choose their own technological stack if when they are proficient in their chosen technologies and there is at least one mentor with the knowledge of that technological stack.
6. The best game ideas are students' own ideas because there is an emotional bond between students and their work.
7. The most important resource needed to organize the MIMI event is the mentor's team and their time. We suggest having one mentor per team where each team contains 5–6 students.
8. The project could be organized as a multidisciplinary university-wide course. Even in this case, it would need to be a non-mandatory course, as it is almost impossible to organize such an event for all students on a programme. This kind of event is directed to a subset of students on selected programmes. If someone plans that all students from a programme take part in such an event, we would suggest the creation of a series of independent events and each student takes part in one of these.

8. Conclusions

The reference implementation of MIMI achieves its main goal, which is to increase the level of soft skills of participating computer science and other students. The uniqueness of the approach is how we build multinational, intercultural and multidisciplinary teams and the intensive schedule. These combine to make the project resemble the way modern companies work.

Almost half of the participants in the GGULIVRR@Lodz project study computer science. The project is developed in a way that enables all students to gain learning that is appropriate to them. The competencies that students develop are not exclusive to IT. All student participants develop the skills needed in their field of study.

After analysis of the GGULIVRR@Lodz project's outcomes from 2014 to 2019, we can conclude that the MIMI methodology presented in this paper results in an effective way of teaching.

We conclude that the best-gamified app ideas are the students own ideas. This is because there is an emotional bond between students and their own creation.

From observations of students' behavior and progress the mentors assessed that:

1. Computer science students are motivated to be open to non-technical students' ideas, which also results in an increase in the self-confidence of all participating students.
2. The optimal length of the project is ten days with eight working days and one day free in the middle of the project. The free day gives students some time for retrospection and time to think over their team's gamified app idea. The free day also gives students time to relax with each other and to enjoy the host city.
3. The prototype test session (7D) prevents students from neglecting technological quality. Without this event, teams focus too much on presentation and overuse mockups.
4. The technological stack and supporting materials must be defined in advance of the project. It helps the computer science students to prepare for the intensive development and results in them achieving a high-quality prototype. The students can choose their technological stack when they are very proficient in their chosen technologies and there is at least one mentor with the knowledge of that technological stack.

Based on the industrial partners' feedback, we can state that the presented gamified apps have real commercial and monetization potential, the prototypes achieve a high technological level and that the students develop a high level of innovative, communication and management skills.

The project can be recreated by other universities. The technologies and themes of the event can be redesigned according to the disciplines and skillset of the team of mentors that takes part in the project.

We recommend that a given mentor is appointed to only one team. The mentor and the team choose the tasks for the prototyped story after a thorough discussion. Ideally, the team of mentors should have some general knowledge and skills in project management, monetization, and business aspects. Team mentors should have complementary competencies and they should support all teams and in aspects connected with their domain of expertise. We note that the requirement that each student team reports their progress to their mentor at least twice a day is very important and that it helps in achieving each student teams' goals.

The prepared and described MIMI methodology can be adapted for use in the form of a university-wide course (i.e. a course for any student from any discipline).

Acknowledgment

The authors would like to thank all the participants of GGULIVRR@Lodz events, the students and mentors. There are so many of them that we can name only a few: Gilles Audemard, Michał Beczkowski, Hiram Bollaert, Norbert Borowski, Virginie Hayenne-Cuvillon, Mykola Kozlenko, João Monteiro, Phillipe Possemiers, and Grzegorz Szewczyk. Additionally, we want to thank all the companies: 9Bits, ABB, Accenture, ATOS (formerly AMG.net), Comarch, Ericpol, Fujitsu, Google, IBM, Mobicca and Technitel for their longtime support.

References

- Ahmed, F., Capretz, L.F., Campbell, P. (2012). Evaluating the demand for soft skills in software development. *IT Professional*, 14(1), 44–49.
- Almeida, F., Bolaert, H., Dowdall, S., Lourenço, J., Milczarski, P. (2015). The WalkAbout framework for contextual learning through mobile serious games. *Education and Information Technologies*, 20(3), 415–428.
- An, B., Brown, D., and Guerlain, S. (2019). The evaluation of a serious game to improve cross-cultural competence. *IEEE Transactions on Learning Technologies*, 12(3), 429–441.
- Barata, G., Gama, S., Jorge, J., Gonçalves, D. (2013). Improving participation and learning with gamification. In *Proceedings of the First International Conference on Gameful Design, Research, and Applications – Gamification '13*. ACM Press.
- Beckingham, S. (2018). Baseline skills – scaffolding soft skills development within the curriculum. In *Higher Education Computer Science*, pages 221–242. Springer International Publishing.
- Berklings, K., Incekara, M.A., Wolske, T. (2019). A study of dispositions according to the IEEE information technology curricula 2017 for german industry and student population. In *2019 IEEE Global Engineering Education Conference (EDUCON)*. IEEE.
- Börstler, J., Hilburn, T.B. (2015). Team projects in computing education. *ACM Transactions on Computing Education*, 15(4), 1–5.
- Capretz, L.F. and Ahmed, F. (2010). Why do we need personality diversity in software engineering? *ACM SIGSOFT Software Engineering Notes*, 35(2), 1–11.
- Chickering, A.W., Gamson, Z.F. (1987). Seven principles for good practice in undergraduate education. *AAHE bulletin*, 3:7.
- Devedzic, V., Tomic, B., Jovanovic, J., Kelly, M., Milikic, N., Dimitrijevic, S., Djuric, D., and Sevarac, Z. (2018). Metrics for students' soft skills. *Applied Measurement in Education*, 31(4), 283–296.
- Forum Word Economics (2016). Global challenge insight report: The future of jobs, employment, skills and workforce strategy for the fourth industrial revolution.
- Frezza, S., Daniels, M., Wilkin, A. (2019). Assessing students' IT professional values in a global project setting. *ACM Transactions on Computing Education*, 19(2), 1–34.
- Gestwicki, P., McNely, B. (2016). Interdisciplinary projects in the academic studio. *ACM Transactions on Computing Education*, 16(2), 1–24.
- Gestwicki, P., Sun, F.-S. (2008). Teaching design patterns through computer game development. *Journal on Educational Resources in Computing*, 8(1), 1–22.
- GGULIVRR@Lodz, P. (2020). web page: <https://ggulivrrr.uni.lodz.pl>
- Gomez, J.E., Huete, J.F., Hernandez, V.L. (2016). A contextualized system for supporting active learning. *IEEE Transactions on Learning Technologies*, 9(2), 196–202.
- Harden, Crosby, Davis, Howie, Struthers (2000). Task-based learning: the answer to integration and problem-based learning in the clinical years. *Medical Education*, 34(5), 391–397.
- Hazzan, O., Har-Shai, G. (2014). Teaching and learning computer science soft skills using soft skills. In *Proceedings of the 45th ACM technical symposium on Computer science education – SIGCSE '14*. ACM Press.

- Hazzan, O., Lapidot, T., Ragonis, N. (2014). *Guide to Teaching Computer Science*. Springer London.
- Herrmann, A., McFarland, C. (2019). Future workers: computer science, apprenticeships and soft skills: lessons in education and training from the united states.
- Ibanez, M.-B., Di-Serio, A., Delgado-Kloos, C. (2014). Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, 7(3), 291–301.
- Ivanova, G., Kozov, V., Zlatarov, P. (2019). Gamification in software engineering education. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE.
- Johns-Boast, L., Flint, S. (2013). Simulating industry: An innovative software engineering capstone design course. In *2013 IEEE Frontiers in Education Conference (FIE)*. IEEE.
- Johns-Boast, L., Flint, S., et al. (2009). Providing students with 'real-world' experience through university group projects. In *20th Annual Conference for the Australasian Association for Engineering Education, 6–9 December 2009: Engineering the Curriculum*, page 299. Engineers Australia.
- Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society, editor (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc.
- Jones, K., Leonard, L.N.K., Lang, G. (2016). Desired skills for entry level IS positions: Identification and assessment. *Journal of Computer Information Systems*, 58(3), 214–220.
- Kapp, K.M. (2012). *The Gamification of Learning and Instruction*. John Wiley & Sons Inc.
- Katarina Pažur, A., Arbanas, K. (2017). Prestige and collaboration among researchers in the field of education and career development of ICT graduates. *Journal of Information and Organizational Sciences*, 41(2), 231–262.
- Kolb, A. Y., Kolb, D. A. (2005). Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning & Education*, 4(2), 193–212.
- Mangaroska, K., Giannakos, M. (2019). Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning. *IEEE Transactions on Learning Technologies*, 12(4), 516–534.
- Marques, M. R., Quispe, A., Ochoa, S. F. (2014). A systematic mapping study on practical approaches to teaching software engineering. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE.
- McKinsey Center for Government (2014). *Education to employment: Getting Europe's youth into work*. McKinsey & Company.
- Michaelsen, L. K., Bauman Knight, A., Dee Fink, L., editors (2004). *Team-Based Learning: A Transformative Use of Small Groups in College Teaching*. Stylus Publishing.
- Milczarski, P., O'Reilly, D., Podlaski, K., Almeida, F. L., Dowdall, S., Hlobaz, A., Bolaert, H., Lourenco, J. (2018). How participation in an intensive project can increase 3rd level students' awareness of entrepreneurship. In: *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI)*, pages 394–404.
- Milczarski, P., Podlaski, K., Hlobaz, A., Dowdall, S., Stawska, Z., O'Reilly, D. (2021). Soft skills development in computer science students via multinational and multidisciplinary gamedev project. In: *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, SIGCSE '21*, New York, NY, USA. Association for Computing Machinery.
- Murphy, C., Powell, R., Parton, K., Cannon, A. (2011). Lessons learned from a PLTL-CS program. In: *Proceedings of the 42nd ACM technical symposium on Computer science education – SIGCSE '11*. ACM Press.
- Nazligul, M.D., Yilmaz, M., Gulec, U., Gozcu, M.A., O'Connor, R.V., Clarke, P.M. (2017). Overcoming public speaking anxiety of software engineers using virtual reality exposure therapy. In *Communications in Computer and Information Science*, pages 191–202. Springer International Publishing.
- O'Reilly, D., Almeida, F.L., Podlaski, K., Bollaert, H., Milczarski, P., Dowdall, S., Hlobaz, A., Lourenço, J. (2015). Using intensive games development projects to teach entrepreneurial skills to third level students. In: *Proceedings of Multinational Conference on Education and New Developments (END 15), Porto, Portugal*, pages 189–193.
- Ozkan, H., Degirmenci, B., Musal, B., Itil, O., Akalin, E., Kilinc, O., Ozkan, S., Alici, E. (2006). Task-based learning programme for clinical years of medical education. *Education for Health: Change in Learning & Practice*, 19(1), 32–42.

- Pastel, R., Seigel, M., Zhang, W., and Mayer, A. (2015). Team building in multidisciplinary client-sponsored project courses. *ACM Transactions on Computing Education*, 15(4), 1–23.
- Porter, L., Lee, C.B., Simon, B. (2013). Halving fail rates using peer instruction. In *Proceeding of the 44th ACM technical symposium on Computer science education – SIGCSE '13*. ACM Press.
- Reardon, S., Tangney, B. (2015). Smartphones, studio-based learning, and scaffolding. *ACM Transactions on Computing Education*, 14(4), 1–15.
- Saltz, J.S., Heckman, R.R. (2018). A scalable methodology to guide student teams executing computing projects. *ACM Transactions on Computing Education*, 18(2), 1–19.
- Savin-Baden, M. (2005). Learning spaces, learning bridges and troublesomeness: The power of diverse approaches to problem-based learning. In *Problem-based Learning: New Directions and Approaches*, pages 3–16. Learning Academy Temasek Polytechnic.
- Savin-Baden, M. (2007). Challenging models and perspectives of problem-based learning. In *Management of Change*, pages 9–29. Brill | Sense.
- Schmidt, H.G., Molen, H.T.V.D., Winkel, W.W.R.T., Wijnen, W.H.F.W. (2009). Constructivist, problem-based learning does work: A meta-analysis of curricular comparisons involving a single medical school. *Educational Psychologist*, 44(4), 227–249.
- Simonson, S.R., editor (2019). *POGIL: An Introduction to Process Oriented Guided Inquiry Learning for Those Who Wish to Empower Learners*. Stylus Publishing.
- Sorensen, K., Mas, J.-M. (2015). A roadmap for the development of labor market information systems. Technical report, USAID.
- Sukhoo, A., Barnard, A., Eloff, M.M., Poll, J.A.V.D., Motah, M. (2005). Accommodating soft skills in software project management. *Issues in Informing Science and Information Technology*, 2:691–703.
- Tomić, B., Jovanović, J., Milikić, N., Devedžić, V., Dimitrijević, S., Đurić, D., Ševarac, Z. (2017). Grading students' programming and soft skills with open badges: A case study. *British Journal of Educational Technology*, 50(2), 518–530.
- Walker, E.L., Slotterbeck, O.A. (2002). Incorporating realistic teamwork into a small college software engineering curriculum. *J. Comput. Sci. Coll.*, 17(6), 115–123.
- Wang, A.I. (2011). Extensive evaluation of using a game project in a software architecture course. *ACM Trans. Comput. Educ.*, 11(1).
- Woods, D.R. (2013). Problem-oriented learning, problem-based learning, problem-based synthesis, process oriented guided inquiry learning, peer-led team learning, model-eliciting activities, and project-based learning: What is best for you? *Industrial & Engineering Chemistry Research*, 53(13), 5337–5354.

S. Dowdall is a mathematics lecturer in Dundalk Institute of Technology, Ireland. He is passionate about teaching mathematics and computing and in particular Artificial Intelligence and Machine Learning. He has been involved with organizing and teaching mathematics to all age groups and skill levels from his work in DCU's Centre for Talented Youth to DkIT's Mathematics Learning Centre as well as Ireland's National Maths Week annual event. As well as mathematics he enjoys working on Games Development projects including Erasmus+ projects such as GGULIVRR@Lodz.

A. Hlobaž is an Assistant Professor at the Faculty of Physics and Applied Informatics at the University of Lodz since 2009. He has PhD in Technical Sciences in the field of Computer Science. He was involved in several research and didactical projects like GENIUS or BUSIT. He has experience in organizing didactic Intensive Programme Projects like GGULIVRR@Lodz 2014-2020 and has over 10 years of experience in teaching in the field of IT. He teaches programming, computer networks (Cisco CCNA and CCNP Instructor) and subjects related to the websites designing. He is also department coordinator of cooperation with employers and business.

P. Milczarski – Assistant Professor at the Faculty of Physics and Applied Informatics at the University of Lodz. Piotr is open to the new challenges of the 21st century. His research interests are connected with image processing, machine learning and AI and their applications in the psychology and sociology, medicine, cybersecurity, social media and education etc. From the beginning of his work, Piotr is involved in many multidisciplinary, national and international projects where students and mentors look for the ideas and their implementations as applications to solve world's problem: immigration and integration and food wasting, e.g. Erasmus + Genius, Erasmus + Future Time Traveller, CFOOD etc., GGULIVRR@Lodz (2014-2020). Piotr has more than 50 articles journal or international conference proceedings. Piotr has been a member in several organizing or program committees of the international conferences.

D. O'Reilly has 30 years of experience teaching in computing science. He is a Senior Lecturer. He holds post-graduate degrees in both Computer Science and Education. Derek has published numerous conference papers, journal papers and book chapters. Derek has chaired several conference panels. Derek is an expert in the areas of mobile, web and computer games development. Derek has sat as an expert on validation panels in several Irish Higher Education Institutions Higher Education Institutions. He is an external examiner for other Irish Higher Education Institutions. Derek has served as an elected member on his college's Academic Council. He is vice-chair of his school's School Board, where he is chair of the Academic Regulations group and the Internationalisation sub-group. Derek is his department's International & Erasmus Coordinator. Derek has extensive experience in participating in international student intensive projects, including the Erasmus+ projects "WalkAbout", "Serious Gaming for a Better Europe", "GENIUS", and "JEU", and the non-Erasmus+ funded projects "GGULIVRR@Lodz" and "eCOAL". Derek has been involved in the promotion of student participation in international coding competitions, such as Microsoft's "Image Cup" and Abertay University's "Dare to be Digital". Derek has delivered classes in various European colleges and has organised for European academic colleagues to visit his college on staff mobilities.

K. Podlaski is an Assistant Professor at the Faculty of Physics and Applied Informatics at the University of Lodz, Poland. He obtained PhD in Theoretical Physics. Later he switched his research interests into Computer Science. Krzysztof Podlaski's research focuses on reversible computation, optimization methods, and soft-skills development in computer science students. For many years Krzysztof Podlaski is working at the University of Lodz and conducts didactics in Computer Science, mostly on programming methods. He took part in many national or international educational and scientific projects, being twice the project manager. Krzysztof Podlaski has more than 30 articles journal or international conference proceedings publications with more than 100 citations (in Scopus). He was a member of the Organizing or Program Committee of several international conferences and reviewed articles to various scientific journals.

Z. Stawska is an Assistant Professor at the Faculty of Physics and Applied Informatics at the University of Lodz, Poland. She is a graduate of the FTiMS department of the Lodz University of Technology. She obtained PhD in Technical Sciences in the field of Computer Science. Her main topics of research interests are image processing and recognition, application of computer science in medicine, and soft-skills development in computer science students. She has more than 20 journal or international conference proceedings publications. She has many years of didactic experience in the field of IT. She teaches mainly programming in Java, Python, and C ++.