Co-funded by the
Erasmus+ Programme
of the European Union

Accelerating the transition towards Edu 4.0 in HEIs

TEACH 4EDU4

Collaborative Teaching Methods for Joint Creative Classrooms

University of l'Aquila: Henry Muccini, Maria Vittoria Isidori, Luca Forlizzi, Giovanna Melideo.

University of Zagreb: Igor Balaban **The Open University:** Bart Rienties **The University of Belgrade:** Aleksandar Marković, Nikola Zornić **Polytechnic University of Catalonia:** Ariadna Llorens **Tallinn University:** Sirje Virkus **The Open University:** Bart Rienties **The University of Žilina:** Peter Marton

L'Aquila, Italy, 27 April 2023

Executive Summary

This document summarises the methodological approach the TEACH4EDU4 partnership followed to design and implement Joint Creative Classrooms.

First, a definition of what a Joint Creative Classroom (JCC) is has been provided and discussed among the partners. A set of adjustments have been implemented in order to satisfy the concrete constraints coming from the partners.

Then, a list of attributes to be considered when designing a JCC have been selected, discussed, and refined. Such a list formed the baseline to identify different types of JCCs. Before starting creating JCCs, a systematic literature review has been conducted, as a way to investigate collaborative teaching experience elicited from the literature. Eventually, an actionable process has been invented to create the TEACH4EDU4 JCCs.

The output of this approach is a set of eight JCCs that have been run during the 19.04.2022 - 31.01.2023 period. 275 students were enrolled in the eight JCCs. In total 162 participants completed the pre-survey, while 108 participants completed the post-survey, which is a response rate of 59% and 39% respectively. At the start of the JCCs students had high expectations in terms of effective and real-world skills development and flexibility in terms of how to study. It is noteworthy to report that on most elements of Education 4.0 the JCCs actually delivered, or perhaps even exceeded students' expectations. In particular it is recommendable that students felt that the JCCs provided skills and competences of the modern CS workplace, that there was flexibility in terms of how they could study and work together. At the same time, probably because of the relatively new innovation fewer students felt part of the (co-)creation and implementation of the JCCs, which given the context in which teachers from different institutions had to collaborate on a relatively short notice together is not entirely surprising.

In total 16 Computer Science (CS) teachers as part of eight JCCs were subsequently interviewed. Using thematic analysis three main themes emerged. First, the overall experiences and perspectives of the BDP tool were positive by most teachers as it allowed participating teachers to share and discuss their initial learning design ideas with their JCC colleagues in other institutions. Second, several positive aspects of the JCC experiences were repeatedly mentioned, including collaborating with educators from different institutions, working with other students, and seeing more motivated students as a result of the JCCs. At the same time, several negative aspects were related to bureaucracy and the requirements to invest more time to run these JCCs. Also there were some communication issues in some but not all JCCs.

Overall, the results from students and teachers suggest that the implementation of the JCCs was mostly successful, and encouraged students to develop and nurture these Education 4.0 skills in a European context. Future policy development would be needed to structurally embed and support these JCCs to facilitate workloads of teachers, and help with accreditation of students' activities.

Co-funded by the Erasmus+ Programme of the European Union

CONTENTS

Co-funded by the
Erasmus+ Programme
of the European Union

Co-funded by the Erasmus+ Programme of the European Union

1. Introduction

One of the features of Industry 4.0 are definitely virtual classrooms “in which the lecturers and students are participating from different places at the same time and the lecturers can store lecture material to be watched or aired in the future” (Beier et al., 2012). This is even more important in computer science where workers very often work in virtual teams and students should be equipped with such skills during their education (IO1). With that sense in IO3 we define Joint Creative Classrooms (JCC) which are computer science courses developed in collaboration of two or more partner universities by using LD models (IO2) with a total workload of (expected) 2-3 ECTS. JCC is fully inline with Edu 4.0 recommendations and will support innovative joint teaching and creative learning practices such as personalised and active learning, learning by osmosis, collaborative learning and existing cases from real life educational settings thus making learning more interesting and effective. Both innovative teaching and creative learning presuppose an active role for the learner and new roles for the teacher who acts mainly as mentor and facilitator of the learning process. JCC will serve as showcases of courses within Computer Science fields (e.g. Software architecture, Web development, Human computer interaction, Modelling and computer simulation, Internet/IoT security, Machine learning, Data analytics/data mining) supporting the development of teachers’ skills and competences related to Edu 4.0. In JCC, teachers from partners’ universities are co-creators of the content using different approaches detected in IO1 (e.g. blended instruction, flipped classroom and online teaching) and learning designs from IO2 because it “will make more efficient learning environments that can adapt for diversity in preparation of students” (Adya et al., 2007).

Tasks leading to the production of the intellectual output:

- *JCC conceptualization*: at first, a workshop will be organised with all partners to understand the individual teaching rules and constraints. The output of this action will be a clear understanding of rules and desiderata from all partners. This activity may also bring to the creation of topic-specific contents (e.g., software verification and validation for industry 4.0, model-based engineering industry 4.0 applications, architecting industry 4.0 IoT systems) to be taught collaboratively by different partners;
- *Definition of collaborative teaching methods*: preceded by a literature search, the consortium will define new collaborative teaching methods. Learning by osmosis (Lago, et al., 2012) can be one of such methods, where students attending different (but related) courses, provided by different members of the consortium, work together on a joint project and learn by each other.
- *Design of joint creative classrooms*: once the collaborative methods are identified, the consortium will have to identify those classrooms that may run in a collaborative way. After that stage each JCC will be planned according to the learning design model developed under IO2.
- *Feedback analysis and final review*: milestones for feedback analysis will have to be defined to evaluate possible shortcomings, difficulties, or barriers. Two or more monitor and replan rounds will be considered. A final review will also assess the students’ liking of the joint classroom as well as their acquired knowledge, and compare the output with the traditional (non collaborative) model.

Co-funded by the Erasmus+ Programme of the European Union

- *Applied methodology*: workshops, qualitative and quantitative methods will be used.

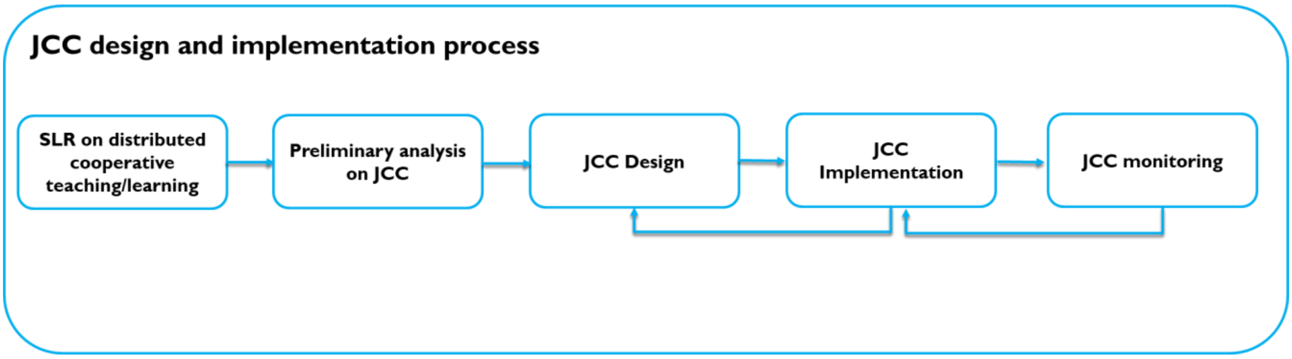


Figure 1.1 JCC design and implementation process

2. Preliminary analysis on JCC

IO3 supports innovative joint teaching and creative learning practices such as personalized and active learning, learning by osmosis, collaborative learning and existing cases from real life educational settings thus making learning more interesting and effective. In this project, within IO3, previously developed learning design models were piloted in the form of Joined Creative Classrooms (JCCs).

The main idea of Collaborative Teaching Methods for Joint Creative Classrooms (JCCs) was to define Joint Creative Classrooms (computer science courses) developed and held in collaboration with 2 or more partner universities by using Learning Design models (LD models) with a total workload of 2-3 ECTS.

JCC can be seen as a unique student-centred environment that enable students from different partner institutions to join the same classroom and in addition to domain knowledge, to gain mobility from home experience and transferable skills (Communication, Reliability, Teamwork, Organization, Adaptability, Leadership and Technological Literacy).

JCC is fully in line with Edu 4.0 recommendations and will support innovative joint teaching and creative learning practices thus making learning more interesting and effective. Both innovative teaching and creative learning presuppose an active role for the learner and new roles for the teacher who acts mainly as mentor and facilitator of the learning process.

Elements of innovation are shown as follows:

- Content and Curricula: Less extensive curricula covering fewer topics in more depth is introduced, besides factual knowledge, transversal soft skills such as problem-finding, problem solving, and collaboration are also developed, Industry is involved in creation of some topics.

Co-funded by the Erasmus+ Programme of the European Union

- Assessment: Assessment for creative learning and innovative pedagogies provides valuable insights into individual student's learning and accounts for individuals' progress as creative learners.
- Teaching practice: Recommendations for teachers on effectively playing new roles as mentors and facilitators of learning and acting as role models of creativity and innovation, Recommendations for teachers on managing a classroom during group work or peer assessment, and make innovative use of ICT for creative learning, Recommendations on how to handle students that are distributed across the globe and in different time-zones, how to effectively teach and conduct assessments (especially, written exams) by using technologies and approaches appropriate for distance education in JCC context.

In this Project, developed JCCs serve as showcases of courses within Computer Science fields (e.g., Software architecture, Web development, Human computer interaction, Modeling and computer simulation, Internet/IoT security, Machine learning, Data analytics/data mining) supporting the development of teachers' skills and competences related to Edu 4.0. In JCC, teachers from partners' universities are
co-creators of the content using different approaches detected in IO1 (e.g., blended instruction, flipped classroom and online teaching) and learning designs from IO2.

Originally it was planned to conduct all developed JCCs from June 2021 to July 2022. Due to the COVID-19 pandemic, as well as the need for harmonization among numerous institutions, study levels, teaching dates, as well as the desire of a larger number of teachers to participate in the realization of the JCCs, this time frame has been extended until February 2023.

The main idea of JCCs is shown in the picture below.

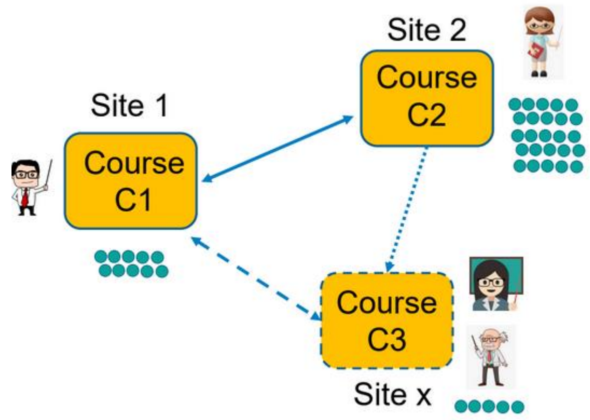


Figure 2.1 Different sites, courses, and collaboration between them

We have different courses (Course 1, Course 2, Course 3) offered by different universities (Site 1, Site 2, Site X) by one or more teachers. The green dots represent students. Their number can vary significantly from one institution to another. There are classes with a small number of students, classes with more students and so on. Arrows represent connections between different courses.

Co-funded by the
Erasmus+ Programme
of the European Union

Different arrows indicate that the relations between individual courses may be different. This actually means that the way in which for example Course 1 and Course 2 collaborate may differ from the way in which the other two observed courses collaborate. In the initial phase of creating the JCC, it was considered how the courses would cooperate with each other.
During the development of the JCCs we considered a few valuable information sources when considering what a JCC can be: conclusions obtained from a systematic literature review; teachers' personal experience and knowledge; or asking TEACH4EDU4 partners to come with their own understanding of JCCs.

Let's now consider how this collaboration works.

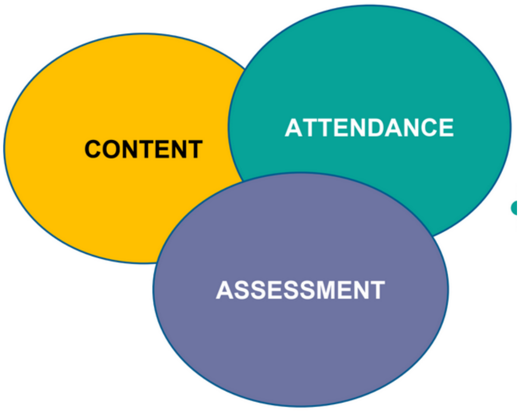


Figure 2.2. JCC collaboration dimensions

When considering possible forms of collaboration between courses, three key dimensions should be considered:

- Content – Here we should answer the question “How to put together different course syllabuses?”. There are a few possible ways to do it:
  - Transfer
  - Fusion
  - Expansion
  - Complement
  - Complement theory/practice
- Attendance – Here we should answer the question “How are students attending the joint course? Do they attend the whole course or parts of multiple courses?”. Types of attendance may be:
  - Fusion
  - Expansion
  - Complement
  - Complement theory/practice
- Assessment - Here we should answer the question “How do we grade students, at which period during the project?”. Types of assessment may be:
  - Fusion

Co-funded by the Erasmus+ Programme of the European Union

- Expansion
- Complement
- Complement theory/practice

By reviewing the literature, as well as based on consultations with colleagues from other universities, four different models of collaboration were identified in term of content (Figure 2.3) - orange part. It should be emphasized here that these are not the only four possible models of collaboration.

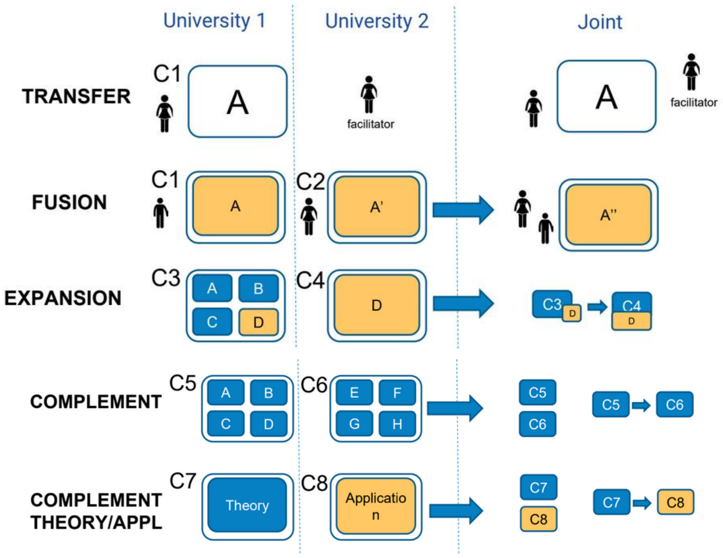


Figure 2.3. Four identified collaboration models in term of content

The first form is fusion. Looking at the content, we are talking about two very similar courses. When reviewing the initial JCC topic proposals, it was noted that there are courses provided by more than one institution that have similar or the same keywords (for example software engineering) or similar content. The idea behind fusion is that two or more universities create one integrated module that will be offered to all students. Teachers collaborate in teaching and students collaborate in learning and projects, on one integrated course. One possible variation of this model can be when one university doesn't have some course at all but collaborates with professors from another university to bring that course to their students.

The second model is expansion. There are courses with multiple topics, where one of them is covered as a separate course at another university. A course in which one topic is developed as an introduction can be linked to a course at another university in which that topic is covered in detail.

The third form of collaboration in terms of content is complement. We have two courses that are quite different in terms of content. There is no overlap of content here (there is nothing orange in intersection), but they look at the same topic from different perspectives. The idea is that students work together in these courses to cover the topic from multiple angles and put together and complement two courses.

Co-funded by the
Erasmus+ Programme
of the European Union

The fourth model is complementing theory/application collaboration. This model refers to the complementarity of two courses where one topic is considered through theoretical foundations and the other through practical application. Such courses can also be integrated through a joint JCC.

FUSION
C1 and C2 are different instances on the same topic
Concrete Example:
- C1= Sw Engineering @Zagreb
- C2 = Sw Engineering @Belgrade
- C2= Sw Engineering @Zilina

COMPLEMENT:
C5 and C6 cover two views of the same topic
- C5 = tech design of mobile apps
- C6 = business plan for mobile apps development

EXPANSION:
C3 expands on a topic of C4
Concrete Example:
- C3 = Sw Engineering with 3 lectures on agile sw development
- C4 = Agile Software Development

COMPLEMENT THEORY/APPLICATION:
Both courses are on the same topic, but C7 on its theory, C8 on its practice
- C7 = formal methods
- C8 = tools and practice of formal methods

Figure 2.4. Examples of possible collaboration

When it comes to the dimensions of attendance and assessment, it is possible to distinguish several types of collaboration (Figures 2.5 to 2.8):

- Fusion,
- Expansion,
- Complement,
- Complement theory/practice.

With fusion, it is common to offer one course in the same time frame (for example, same semester). Typically, we have the single joint team project with teams comprising of students from all universities (Figure 2.5).

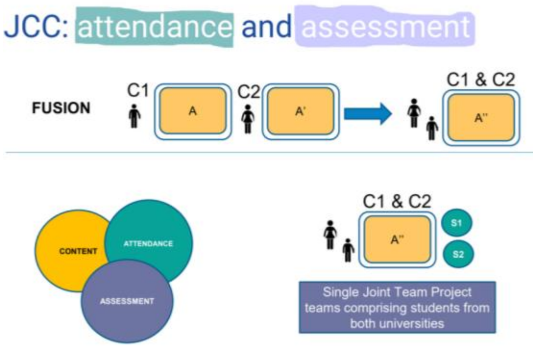


Figure 2.5. Collaboration model in terms of attendance and assessment – Fusion

Co-funded by the Erasmus+ Programme of the European Union

Looking at the expansion model, it is possible to distinguish several different sub-models (Figure 2.6). Suppose that C3 is a course of one university, C4 is a course of another university, where one topic of the C3 course is the essence of the C4 course.

- The first sub-model: From the point of view of attendance, it implies that students from both universities listen to both courses, while from the point of view of assessment, students in a mixed team produce a unique project.
- Second sub-model: From the point of view of attendance, students attend courses at their universities, but mixed teams produce a unique project.
- Third sub-model: From the point of view of attendance, students attend courses at their universities, but regarding assessment, students S3 produce project 3 (PR3) and pass it to students S4. In this way, students transmit their knowledge to other groups of students, or they participate in another project.

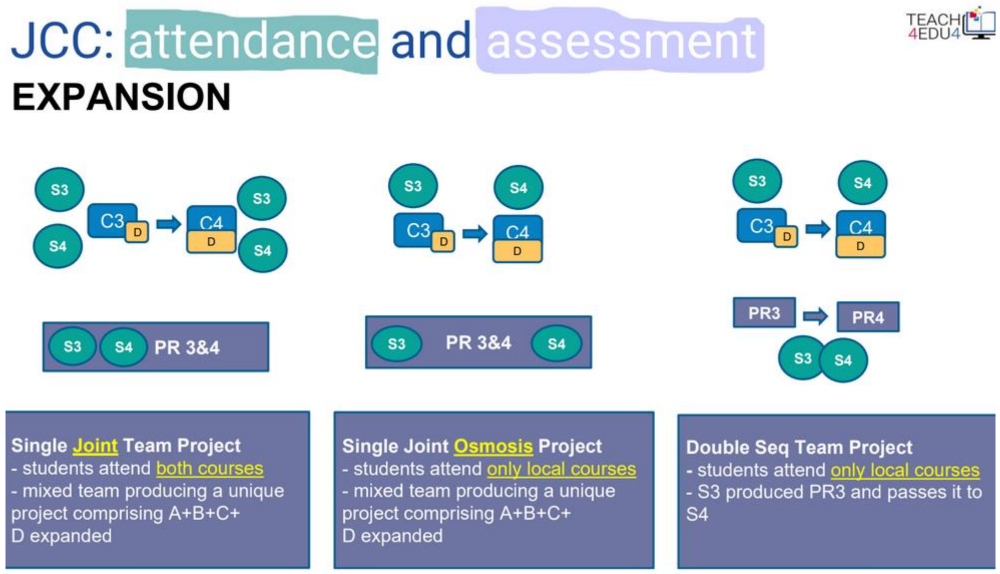


Figure 2.6. Collaboration model in terms of attendance and assessment - Expansion

Regarding the complement model, it is possible to distinguish several different sub-models (Figure 2.7). Suppose that C5 is a course of one university, C6 is a course of another university, where both courses cover the same topic from different points of view.

- The first sub-model: From the point of view of attendance, it implies that students from both universities listen to both courses, while from the point of view of assessment, students in a mixed team produce a unique project.
- Second sub-model: From the point of view of attendance, students attend courses at their universities, but mixed teams produce a unique project.
- Third sub-model: From the point of view of attendance, students attend courses at their universities, but regarding assessment, students S5 produce project 5 (PR5) and pass it to

Co-funded by the
Erasmus+ Programme
of the European Union

students S6. In this way, students transmit their knowledge to other groups of students, or they participate in another project.

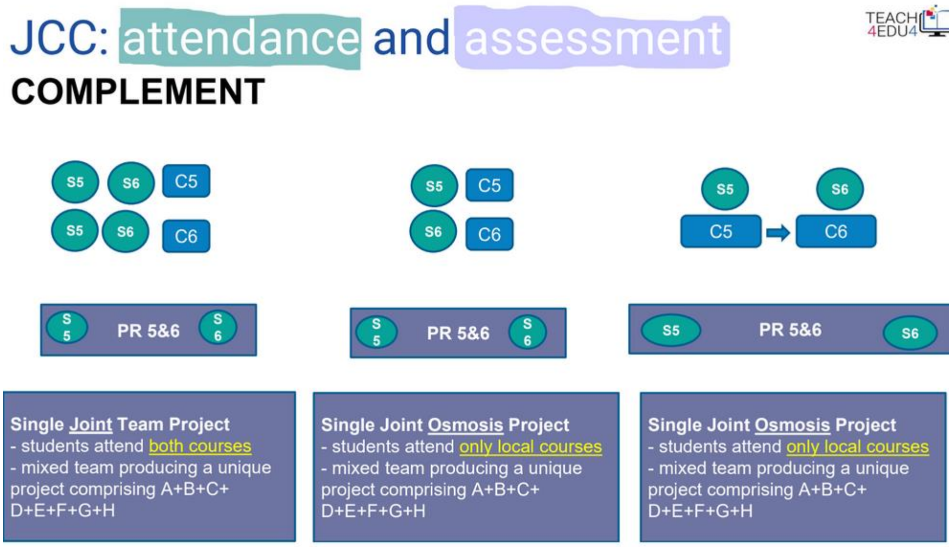


Figure 2.7. Collaboration model in terms of attendance and assessment – Complement

When it comes to the complement T/P model, it is possible to distinguish several different sub-models (Figure 2.8). Suppose that C7 is a course of one university covering theoretical foundations, C8 is a course of another university, dealing with practical application of the same topic.

- The first sub-model: From the point of view of attendance, it implies that students from both universities listen to both courses, while from the point of view of assessment, students in a mixed team produce a unique project consisting of Theory and Application.
- Second sub-model: From the point of view of attendance, students attend courses at their universities, but mixed teams produce a unique project consisting of Theory and Application.
- Third sub-model: From the point of view of attendance, students attend courses at their universities, but regarding assessment, students S7 produce project 7 (PR7) and pass it to students S8. In this way, students transmit their knowledge to other groups of students, or they participate in another project.

Co-funded by the Erasmus+ Programme of the European Union

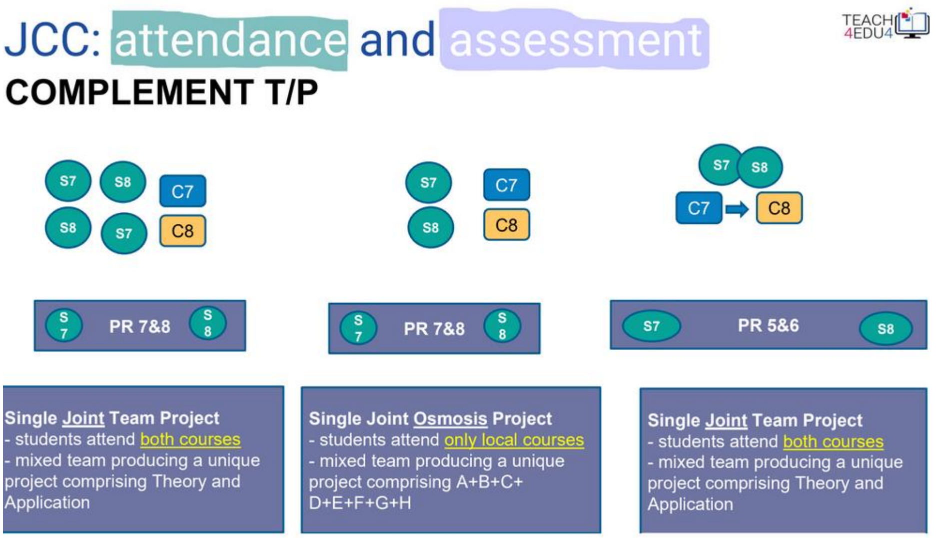


Figure 2.8. Collaboration model in terms of attendance and assessment – Complement T/P

JCC attributes

When creating courses (JCC), special attention is paid to the following requirements of courses (Figure 2.9):

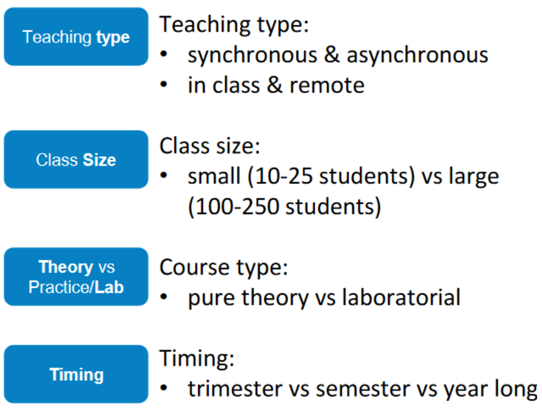


Figure 2.9. Important JCC attributes

Teaching Type - When considering this attribute, there are several different alternatives:

Teaching can be *synchronous* or asynchronous.

Synchronous learning takes place in real time and according to a specific schedule with the help of applications such as Microsoft Teams, Zoom, Google Meet, Moodle, etc. The teacher

Co-funded by the Erasmus+ Programme of the European Union

and students can discuss the lesson together, answer questions, watch slides - simply, it's teaching that gives the feeling of teaching in a classroom.

Asynchronous learning is independent learning without a strictly defined time and place for classes. Students choose when to review the material sent to them by the teacher. The material can be in the form of pre-recorded video lessons, assignments, presentations, documents, etc.

Depending on whether the students gathered in the classroom or in a remote location, the teaching can be *in class* or *remote*.

Class size – *small* vs *large*.

When creating the JCC, we should keep in mind the size of the group. Some classes have a small number of students (10-25), while some similar can have many students (100-250). Special efforts are needed to form JCC in groups with a significant difference in the number of participants.

Theory or Practice – *pure theory* vs *pure practice* or a *mixture* of those.

When creating JCC these are important elements that should be considered.

Timing – *trimester* vs *semester* vs *year long*.

One of the major problems during the development of the JCCs on this project was timing. Some courses were realized in different semesters at different universities. Also, some courses last in one trimester, while others last a semester or a whole year.

An additional problem is the fact that certain subjects appear at different levels of study. This can be observed as an additional attribute to taking care of the formation of the JCC.

JCC - first steps in TEACH4EDU4

The procedure for creating the JCC on this project was realized as follows:

- Each partner named one or two facilitators from their institution (Table 1).
- Each facilitator had a task to arrange a short meeting (30 min. or so) with all university teachers who expressed their interest in the assigned topics.
    a. facilitators were assigned to two or three topics where at least one university teacher is from the same institution as they are;
    b. facilitator shortly described possible cooperation within the JCC;
    c. facilitator filled in the specially prepared spreadsheet based on the received information.
- Status of each JCC was examined and updated in the online document (moderated by prof. Henry Muccini)

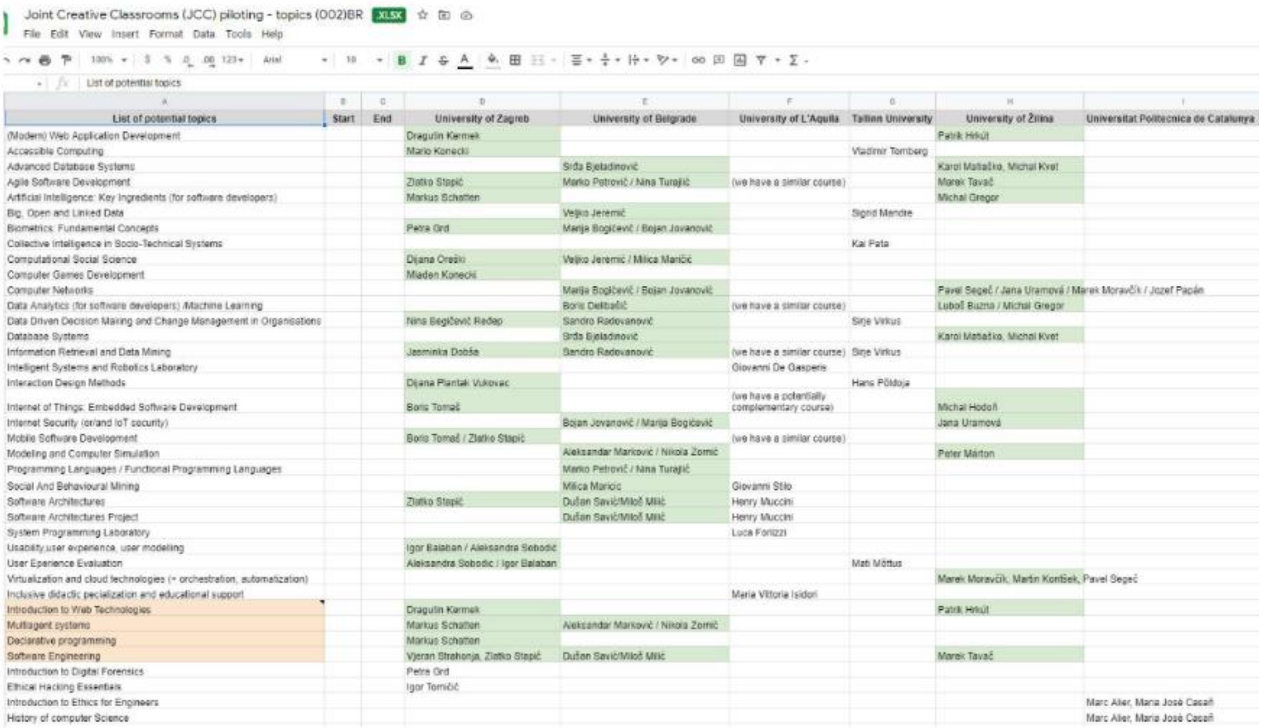
Co-funded by the Erasmus+ Programme of the European Union

- During the preparation of JCCs, partners continuously talked to their teachers and to the team leaders for the team they were facilitating (team leader for each JCC is one of the teachers from that JCC) and observed JCC progress (whether they can manage the workflow, motivate them to plan their JCCs, collect problems and issues).

JCC name	**Facilitators**
Information Retrieval and Data Mining	Goran Hajdin (FOI)
Machine Learning School	Peter Márton (FRI UNIZA)
Modeling and Computer Simulation	Aleksandar Marković & Nikola Zornić (FON)
Internet of Things: Embedded Software Development	Peter Márton (FRI UNIZA)
Robotics: Embedded Software Development	Peter Márton (FRI UNIZA)
Internet Security and Trustworthiness	Igor Balaban (FOI)
Advanced Database Systems	Aleksandar Marković & Nikola Zornić (FON)
Software Architectures Analysis and Design	Henry Muccini (AQ)

Table 1. Selected JCCs and their facilitators

Co-funded by the Erasmus+ Programme of the European Union



Joint Creative Classrooms (JCC) piloting - topics (002)BR

List of potential topics	Start	End	University of Zagreb	University of Belgrade	University of L'Aquila	Tallinn University	University of Žilina	Universitat Politecnica de Catalunya
(Modern) Web Application Development			Dragutin Kermek				Patrik Hrkút	
Accessible Computing			Mario Konecki			Vladimir Tomberg		
Advanced Database Systems				Srđa Bjeladinović			Karol Matiaško, Michal Kvet	
Agile Software Development			Zlatko Stapić	Marko Petrović / Nina Turajlić	(we have a similar course)		Marek Tavač	
Artificial Intelligence: Key Ingredients (for software developers)			Markus Schatten				Michal Gregor	
Big, Open and Linked Data				Veljko Jeremić		Signd Mendre		
Biometrics: Fundamental Concepts			Petra Grd	Marija Bogićević / Bojan Jovanović				
Collective Intelligence in Socio-Technical Systems						Kai Pata		
Computational Social Science			Dijana Oreški	Veljko Jeremić / Milica Maričić				
Computer Games Development			Mladen Konecki					
Computer Networks				Marija Bogićević / Bojan Jovanović			Pavel Segeč / Jana Uramová / Marek Moravčík / Jozef Papán	
Data Analytics (for software developers) /Machine Learning				Boris Delibašić	(we have a similar course)		Luboš Buzna / Michal Gregor	
Data Driven Decision Making and Change Management in Organisations			Nina Begičević Ređep	Sandro Radovanović		Sirje Virkus		
Database Systems				Srđa Bjeladinović			Karol Matiaško, Michal Kvet	
Information Retrieval and Data Mining			Jasminka Dobša	Sandro Radovanović	(we have a similar course)	Sirje Virkus		
Intelligent Systems and Robotics Laboratory					Giovanni De Gasperis			
Interaction Design Methods			Dijana Plantak Vukovac			Hans Põldoja		
Internet of Things: Embedded Software Development			Boris Tomaš		(we have a potentially complementary course)		Michal Hodoň	
Internet Security (or/and IoT security)				Bojan Jovanović / Marija Bogićević			Jana Uramová	
Mobile Software Development			Boris Tomaš / Zlatko Stapić		(we have a similar course)			
Modeling and Computer Simulation				Aleksandar Marković / Nikola Zornić			Peter Márton	
Programming Languages / Functional Programming Languages				Marko Petrović / Nina Turajlić				
Social And Behavioural Mining				Milica Maricic	Giovanni Stilo			
Software Architectures			Zlatko Stapić	Dušan Savić/Miloš Milić	Henry Muccini			
Software Architectures Project				Dušan Savić/Miloš Milić	Henry Muccini			
System Programming Laboratory					Luca Forlizzi			
Usability,user experience, user modeling			Igor Balaban / Aleksandra Sobodić					
User Eperience Evaluation			Aleksandra Sobodic / Igor Balaban			Mati Mõttus		
Virtualization and cloud technologies (+ orchestration, automatization)							Marek Moravčík, Martin Kontšek, Pavel Segeč	
Inclusive didactic pecialization and educational support					Maria Vittoria Isidori			
Introduction to Web Technologies			Dragutin Kermek				Patrik Hrkút	
Multiagent systems			Markus Schatten	Aleksandar Marković / Nikola Zornić				
Declarative programming			Markus Schatten					
Software Engineering			Vjeran Strahonja, Zlatko Stapić	Dušan Savić/Miloš Milić			Marek Tavač	
Introduction to Digital Forensics			Petra Grd					
Ethical Hacking Essentials			Igor Tomičić					
Introduction to Ethics for Engineers								Marc Alier, Maria José Casañ
History of computer Science								Marc Alier, Maria José Casañ

Figure 2.10. Spreadsheet with basic information regarding possible JCCs

3. Systematic literature review on collaborative learning/teaching methods

Joint Creative Classrooms (JCC) refers to courses jointly developed in collaboration between two or more partner universities abroad or in the same country. The cooperation follows two different strategies reflecting two points of view:

- Collaborative teaching based on teacher-centred approaches: teachers are responsible for planning and sharing the lectures. The sectioning depends on the teachers and their knowledge. In this context, teachers may cover the same or different topics of the same subject based on the same or distinct views.
- Collaborative learning based on student-centred approaches: teachers do not share the lecture with the partner universities. The teachers are accountable for planning and promoting collaboration between students of different universities.

In computer science, teachers adopt JCC in software engineering teaching. One of the goals of software engineering courses is to prepare students to be part of a team whose members: i) have different skills, ii) are geographically distributed, iii) speak different languages and iv) come from different cultural backgrounds. Thus, this teaching strategy allows teachers to expose the students to problems and challenges similar to what they will encounter working on real-world projects. Many large software companies have assigned software projects to distributed teams working in their

Co-funded by the Erasmus+ Programme of the European Union

branches in several countries to take advantage of available skills and talent and the cost of labour in various parts of the world. Researchers widely discussed and investigated software engineering education and teamwork. However, little research covers the collaborative teaching methods for joint creative classrooms.

Thus, we conducted a Systematic Literature Review (SLR) to analyse the strategies used in existing experiences of integrating two or more modules to form a joint course in computer science. Those courses combine materials from the participating universities and deliver them to their students remotely or in class. The students could work collaboratively in distributed teams on a shared project to put their theoretical knowledge into practice. We drove this SLR following the guidelines provided by Kitchenham and Charters (Lago et al., 2008). They defined three stages of a successful SLR: i) planning, ii) conducting, and iii) reporting the review. After the research questions definition, we studied the search string needed to select the primary studies on the designated data sources. We defined the selection criteria that allowed us to restrict the number of studies included in the review. Then, the analysis of the extracted data allowed for answering the specified research questions.

Related Work

In this section, we present the relevant SLRs present in the literature. The objective is to compare them to our work. The study reported in (Keele et al., 2007) aimed at identifying the challenges faced by globally distributed teams during various phases of software development. The authors suggested best practices and tools that can help alleviate these challenges. Ritu Jain and Ugrasen Suman (Nordio et al., 2011a) focused on DSD training and education to explore the challenges and proposals that might help researchers, instructors and practitioners. Their results showed a growing interest in DSD training and education, highlighting the need for practical experiences through which students can learn by doing. The study reported in (Fu et al., 2018) aims to disseminate Global Software Engineering (GSE) teaching. The authors map the courses cited in GSE papers and published since 2001, with the ones resulting from an informal search on universities' websites. The main goal of all the related studies mentioned above is to identify challenges and obstacles encountered in DSD courses and to promote the creation and maintenance of a community for DSD teaching. However, the overall objective of this conducted SLR is to investigate and detect any strategies used in the process of course integration and identify the teaching methods of the joint courses and analyse the attendance and the assessment of enrolled students.

Planning the Review

The planning phase comprehends different stages: i) research questions definition, ii) search string creation, and iii) selection criteria identification.

Research questions drive the entire literature review process (Lago et al., 2008). The well-planned selection of studies, the methodology for data extraction, and reporting should help answer the research questions. The present study focused on a methodology/experience on how to integrate two or more modules to form a joint course in computer science. To have a complete understanding of this topic, we formulated the following questions:

Co-funded by the Erasmus+ Programme of the European Union

- RQ1: How are the contents combined in JCC? This research question aims to investigate how teachers converge on the course content. In addition, the research question explores the strategy used in the integration process.

- RQ2: How is attendance managed in JCC? The goal is to configure the number of students enrolled in the JCC courses and the way of attendance.

- RQ3: How is the assessment managed in JCC? The research question aims to analyse how teachers involved in JCC courses perform the examination.

- RQ4: How to prevent dropout of courses? The objective is to detect any dropouts that occurred and report if strategies to contain them.

- RQ5: What is the teaching method? Identifies principles and approaches used by teachers to enable student learning.

The analysis of the research questions spotlighted the keywords needed to construct the following search string.

(Distributed OR global OR international OR Joint OR collaborative OR "virtual exchange" OR telecollaboration) AND (“computer science” OR Informatics OR engineering OR Software) AND (Education OR learning OR teaching OR development) AND (course* OR project*)*

Study selection criteria allow for identifying the primary studies that provide direct evidence about the research question. The inclusion criteria for the selection are listed below:

- Papers in English
- Papers published in ACM, IEEE and Wiley libraries
- The paper is about a methodology/experience on how to integrate two or more modules to form a joint course. More precisely:
    - The joint course combines the contents coming from two or more courses from the same or different university;
    - A course designed and planned by different teachers from different universities. In this scenario, the teachers work together;
    - A course attended by students enrolled and located in different universities. Thus, the students attend together;
    - A project or assessment that is completed jointly by students from different universities (students work together);
    - The papers that report the methodology adopted to design joint courses, experience reports, or pedagogical aspects.

In addition, we excluded the papers responding to the following exclusion criteria:

- Duplicated research papers;

Co-funded by the Erasmus+ Programme of the European Union

- Research papers not directly related to the objective of this research;
- Unavailable or abstract-only research papers.

Conducting the Review

The conduction phase comprehends the study selection and data extraction. We ran the search string in the above-mentioned digital libraries. The study selection went through three stages:

- **Primary selection**. In the first stage, we check for duplication in the dataset we retrieved from the digital libraries. For each duplicated paper, we included just one copy in our study.
- **Secondary selection**. In this stage, we start reading the title, abstract and conclusion of the papers and apply the study selection criteria to select the research papers that shed light on our study.
- **Final selection**. In the end, we read the full text of the selected papers to verify whether they satisfy the selection criteria.

The following table reports the number of works resulting from the research on digital libraries and the application of the three steps. After the secondary selection, we read 59 research papers. We rejected 11 research works out of 59, either due to i) repetition, ii) papers discussing the same course from a different point of view and refereeing to each other, iii) the papers are not related to computer science or software engineering, iv) have not reported any methodology/experience on how to integrate two or more modules to form a joint course.

DIGITAL LIBRARY	SEARCH RESULT	PRIMARY SELECTION	SECONDARY SELECTION	FINAL SELECTION
ACM Digital Library	286	278	44	37
IEEE Explore	165	100	15	11
Wiley Online Library	20	18	0	0
Total	471	396	59	48

The following table documents the distribution of accepted papers within digital libraries. The selected primary studies come from the ACM library (37 papers) and the Ieee Explore library (11 papers). Concerning the Wiley library, we excluded all papers due to irrelevant content to our SLR topic.

SOURCE	REFERENCE

Co-funded by the Erasmus+ Programme of the European Union

ACM Digital Library	(Su and Scharff, 2010) (Richardson et al., 2006) (Salah et al., 2018) (Azadegan and Lu, 2001) (Bosnic et al., 2011) (Bosnic and ˇCavrak, 2019) (Adya et al., 2007) (Chidanandan et al., 2010) (Macek et al., 1999) (Scharff et al., 2016) (Van Der Duim et al., 2007) (Purvis et al., 2004) (Cavrak et al., 2017) (Marutschke et al., 2020b) (Pirker et al., 2016) (Maresca et al., 2011) (Scharff et al., 2010) (Tong and Clear, 2013) (Scharff, 2011a) (Cavrak et al., 2012) (Ciccozzi and Crnkovic, 2010) (Collazos et al., 2010) (Holmes et al., 2014) (Ota and Punyabukkana, 2016) (Bosnic et al., 2019) (Jain and Suman, 2015) (Marutschke et al., 2020a) (Clarke et al., 2014) (Bothe et al., 2005) (Gotel et al., 2008b) (Gotel et al., 2008a) (Brereton et al., 1998b) (Silva et al., 2015) (Gloor et al., 2011) (Nordio et al., 2011b) (Ende et al., 2013) (Scharff, 2011b) (Fortaleza et al., 2012)
IEEE Explore	(Oda et al., 2017) (Collofello, 2000) (Brereton et al., 1998a) (Rusu et al., 2009) ([Adorjan and Nunez-Del-Prado, 2018) (Beier et al., 2012) (Stroulia et al., 2011) (Bosnic et al., 2013) (Monasor et al., 2010) (Verkamo et al., 2005)

The data extraction process aims to extract the data items needed to answer the research questions. We stored the result of the study selection phase in an excel sheet. The form contains titles, authors, and the parameters and values that will guide the reporting phase. In the following, we describe the parameters and their values used for data extraction. Note that each parameter belongs to one of the categories shown in the previous section: Content, Attendance, and Assessment.

- RQ1.P1 [Content] - Collaborative content definition description. It refers to a general description of how the teachers defined the collaborative content.
- RQ1.P2 [Content] - Collaborative content definition strategy. It describes the strategy used for content definition:
  - Complement: the two universities cover two views of the same topic.
  - Fusion: the two universities involved cover different instances on the same topic.
  - Expansion: one university expands on a topic of the other one.
  - Complement Theory/Application: both courses are on the same topic. However, one university covers its theory and the other its practice.
- RQ1.P3 [Attendance] - JCC maturity measured in terms of:
  - Iterations: it refers to the number of times the course was delivered. The iteration value is independent of academic years.
  - Years: indicates the number of academic years the university offered the course.
- RQ2.P1 [Attendance] - Number of students. It defines the total number of students involved in the JCC course during all iterations or years.
- RQ2.P2 [Attendance]: Attendance site: describe if the student involved in the JCC course attended it:
  - Home university course: students attend the JCC course at their university.
  - Foreign university course: students attend the JCC course at another university.

Co-funded by the
Erasmus+ Programme
of the European Union

  - Partial attendance: students attend the JCC course in both universities based on program content.
- RQ2.P3 [Attendance] - Type of attendance. It defines how students attend the course:
  - Remote: students did not physically attend the lectures in the foreign university.
  - In class: students physically attend classes in a foreign university.
- RQ3.P1 [Assessment]- Type of Assessment Implementation. It describes how the examination is carried out (methods).
  - Oral Examination/open-ended
  - Questions/Multiple Choice
  - Project
  - Presentations
- RQ3.P2 [Assessment] - Type of Assessment Implementation. It explains how the students' evaluation is carried out (methods).
- RQ3.P3 [Assessment] - Location of examination. It defines where the assessment took place:
  - Remote: students did not physically participate in the questioning.
  - In class: students physically participate in the evaluation.
- RQ3.P4 [Assessment] - Number of members per group. If the students' evaluation provides group assessment, this parameter describes the number of students involved in the team.
- RQ3.P5 [Assessment] - Assessment person. It denotes the person responsible for the student's evaluation:
  - Home professor: the home professor evaluates its students.
  - Foreign professor: the foreign professor assesses all the students.
  - Both: home and foreign professors are involved in the evaluation of the students.
- RQ4.P1 [Attendance] - Dropout Prevention of Courses Strategy. It describes the strategies adopted to prevent the dropout of courses.
- RQ5.P1 [Other] - Teaching/Learning method. It refers to the principles and techniques used by teachers to enable student learning:
  - Teacher-Centred: the teacher functions in the familiar role of classroom lecturer, presenting information to the students that passively receive the knowledge from their teachers.
  - Student-Centred: the teacher is still the classroom authority figure. However, the teacher covers the role of a coach or facilitator. Thus the students embrace a more active and collaborative role in their learning by working within groups on specific projects.
  - Mixed: Teacher-Centred and Student-Centred applied together.

Reporting the Review

The last phase of the SLR is reporting the results. Thus, the extracted data are analysed to answer the research questions.

RQ1: How are the contents combined in JCC?

This research question investigates i) how teachers plan and share the content of the JCC and ii) the strategy used to integrate the courses. To reply to this research question, we extracted and studied the three parameters in the following table.

Co-funded by the Erasmus+ Programme of the European Union

PRAMETER ID	PARAMETER DESCRIPTION	ASSOCIATED VALUES
RQ1.P1	Collaborative content description	Open
RQ1.P2	Collaborative content definition strategy	Fusion, Expansion, Complement, Complement Theory/Application
RQ1.P3	JCC maturity	Number of iterations/Years

The investigation of the parameter RQ1.P1 shows that most studies discuss Distributed Software Development and Global Software Engineering. In this context, the student teams distributed across several physical sites work collaboratively on planning, designing, building, testing, and managing software. Thus, teachers pay attention to the course project instead of planning the content combination. Their role is to facilitate communication between students.

Parameter RQ1.P2 reflects this tendency. We can associate one of the values reported in Table 3 just on 8/48 research papers included in this SLR. The fusion strategy applies when the combined courses cover different instances on the same topic. From our analysis, the fusion strategy has been adopted only in (Bosnic et al., 2015)(Marasovic and Lutz, 2015)(Zhang et al., 2020). In (Bosnic et al., 2015), each participating university has its course with its course material. However, they covered similar topics. The course material used by one university is shared, and the lectures are recorded and available for all students. The authors in (Marasovic and Lutz, 2015) report two courses having the same content and syllabus. However, they do not provide information about the teacher's content responsibility. The research paper (Zhang et al., 2020) reports how instructors combined the course program in remote sessions, face-to-face meetings, and site visits as an effective practice and pedagogy to develop student's skills to work in a multicultural setting. The expansion strategy occurs when a course expands on a topic of another one. The studies (Ota and Punyabukkana, 2016) and ([Adorjan and Nunez-Del-Prado, 2018) adopt this approach. In particular, the authors in (Ota and Punyabukkana, 2016) adopted the strategy in the context of a project collaboration between computer science students and law, biology, and design. The goal was to manage the development process of a videogame from design to testing. In ([Adorjan and Nunez-Del-Prado, 2018), business engineering students shared knowledge with Software Engineering on the same project related to family trees. The complement strategy is applied when the combined courses cover different views of the same topic. Based on our results, the study (Macek et al., 1999) uses the complement strategy to combine the joint two classes for global software development teaching. One university draws attention to the use of one Web design method. The other one focused on the motivated selection of one among multiple approaches (specialised topics) together with documentation of design rationale. The study (Adya et al., 2007) discusses a joint project developed by two teams of students enrolled in two courses that cover different views of the same topic. While complement strategy refers to distinct views of the same topic, complement theory/application strategy is applied when the combining courses are on the same topic, but one focuses on the theoretical part of the topic and the other on its practical aspects. The study (Van Der Duim et al., 2007) is an example of the application of this strategy. In fact, in a pilot project of two universities, a cross-cultural student team

Co-funded by the
Erasmus+ Programme
of the European Union

gathered experience by developing a software product in a distributed project. The first team had a strong background in algorithm design needed in the analysis part of the project. The other one was more familiar with the tools required for implementing the project. A similar scenario applies to students involved in the collaboration project discussed in (Maresca et al., 2011).

Parameter RQ1.P3 classifies the maturity of the studies into two categories: iterations and years. The authors in (Ciccozzi and Crnkovic, 2010) give an example of iteration classification. This work describes a collaborative course experience between students from two universities carried out just one time. The study (Bosnic et al., 2019) presents a project-based Software Engineering course carried out in a distributed manner between two universities. They have been offering the class for 12 years, each with a different group of students. Thus, they offered the course for 12 iterations. In total, we have 26 studies that reported the maturity of the JCC in terms of iterations. Several studies report the maturity indicating how long the JCC has been holding without pointing out the number of iterations during these years. The number of years does not necessarily mean the number of iterations. Sometimes the iteration could last for six months and happens twice a year. For example, the study (Jain and Suman, 2015) provides the lessons learned from Managing Distributed Software Engineering Courses through the Undergraduate Capstone Open-Source Projects (UCOSP) program. This course has been running twice a year for five years, each year with a different group of students. Thus, the number of iterations of this course is 10. The following table reports the paper distribution among the two categories.

MATURITY CATEGORY	REFERENCE
Iteration	(Su and Scharff, 2010) (Richardson et al., 2006) (Adya et al., 2007) (Van Der Duim et al., 2007) (Purvis et al., 2004) (Cavrak et al., 2017) (Pirker et al., 2016) (Maresca et al., 2011) (Scharff et al., 2010) (Scharff, 2011a) (Ciccozzi and Crnkovic, 2010) (Ota and Punyabukkana, 2016) (Brereton et al., 1998b) (Scharff, 2011b) (Fortaleza et al., 2012) (Stroulia et al., 2011) (Monasor et al., 2010) (Verkamo et al., 2005) (Bosnic et al., 2019) (Azadegan and Lu, 2001) (Gloor et al., 2011) (Collofello, 2000) (Scharff et al., 2016) (Salah et al., 2018) (Jain and Suman, 2015) (Bosnic and ˇCavrak, 2019)
Year	(Bosnic et al., 2013) (Cavrak et al., 2012) (Bosnic et al., 2011) (Marutschke et al., 2020b) (Tong and Clear, 2013) (Ende et al., 2013) (Oda et al., 2017) (Gotel et al., 2008b) (Brereton et al., 1998a) (Holmes et al., 2014) (Marutschke et al., 2020a) (Silva et al., 2015)

Co-funded by the Erasmus+ Programme of the European Union

	(Nordio et al., 2011b) (Bothe et al., 2005) (Collazos et al., 2010) (Gotel et al., 2008a) (Clarke et al., 2014) (Chidanandan et al., 2010)

The following figure shows the number of studies for each subset of the iterations. Considering all the studies reporting maturity in terms of iterations 69% of them held the course for 1 iteration. Contrariwise, the other left-view studies reported a different number of iterations, where the maximum number of iterations is 12 iterations.

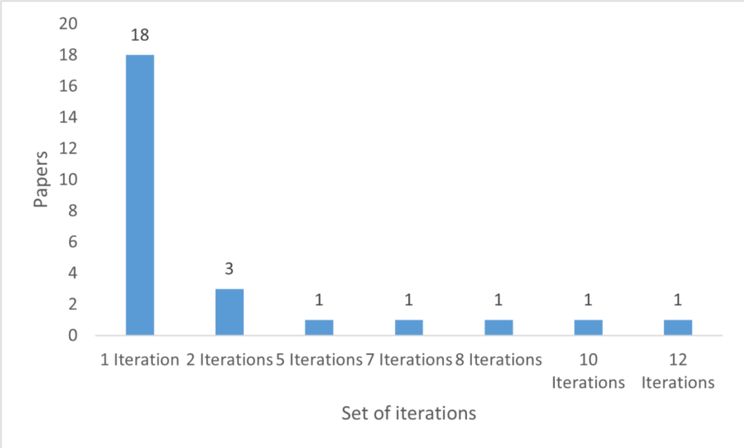


Figure 3.1. Number of studies for each subset of the iterations

The following figure shows the values analysis concerning the studies reporting the JCC maturity in terms of years and their wide variation (from 2 to 15 years).

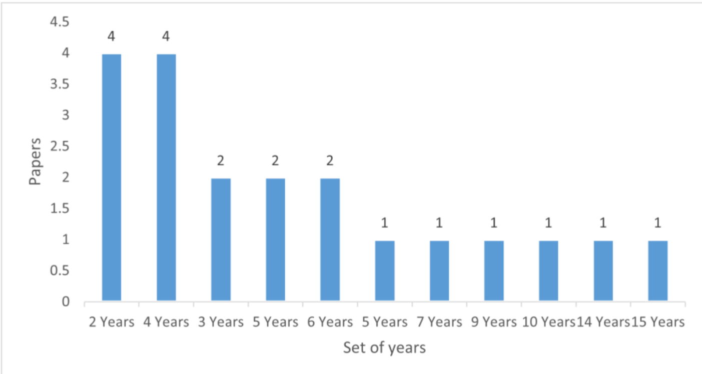


Figure 3.2. Values concerning the studies reporting the JCC maturity

RQ2: How is attendance managed in JCC?

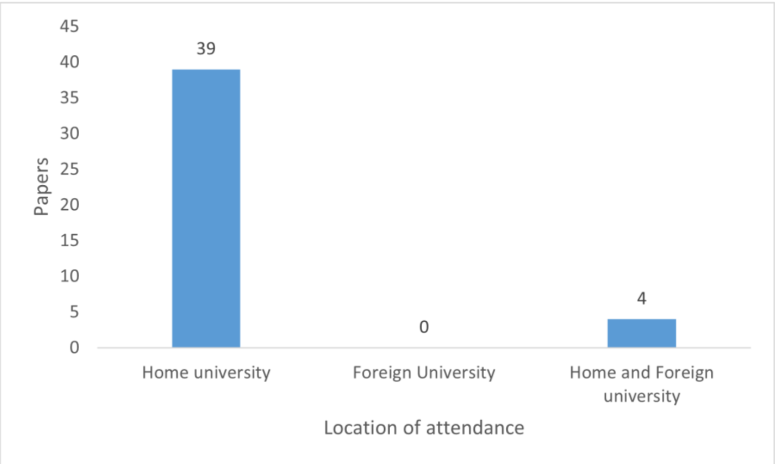
This research question aims to configure the number of students enrolled in the JCC courses and the way of attendance. The following table shows the parameters used to answer the research question.

Co-funded by the Erasmus+ Programme of the European Union

PARAMETER ID	PARAMETER DESCRIPTION	POSSIBLE VALUES
RQ2.P1	Number of students	Open
RQ2.P2	Attendance site	Home university course, Foreign university course, Both
RQ2.P3	Type of attendance	Remote, In Class

The parameter RQ2.P1 aims to report the number of students involved in the JCC course presented in the research paper included in this SLR. There is a strong relationship between this parameter and the JCC maturity. This connection justifies why the number of students reported goes from 5 to 482. On one hand, the paper (Scharff, 2011b) documents one iteration of the JCC experience shared between three universities involving just 5 students. On the other hand, the research paper (Cavrak et al., 2012) discusses the joint Distributed Software Development course between three universities involving 482 students over 14 years. Thus, 27 is the average number of students enrolled in a JCC for iterations and years.

Our study selection criteria aimed to identify all the research works discussing a joint course created by combining knowledge and contents of two or more classes. In this context, the goal of parameter RQ2.P2 is to investigate the location where the students attend the class. The analysis of the extracted data shows that all the selected primary papers examine a joint course between universities of different countries. The results show that 81% of the students attend a JCC from their home university using online communication tools. Four of the included research studies report the experience of students attending the course in both universities. The paper (Ciccozzi and Crnkovic, 2010) describes a collaborative course experience between students from two universities. Student teams worked together on a software engineering project for a non-profit organisation. The student teams from one university flew to the other for face-to-face work. Thus, they attended the course in two different locations. The following Figure 3.3 shows the distribution of the included studies concerning the three provided categories of attendance location.



Co-funded by the Erasmus+ Programme of the European Union

Figure 3.3. Distribution concerning attendance location

The aim of parameter RQ2.P3 is to underlying the overall type of attendance. In this scenario, we identified two categories of attendance i) remotely and ii) in class. In the first case, students use communication tools to attend the course. In the latter, the students are physically present in the classroom. Note that the RQ2.P3 parameter differs from the RQ2.P2 because of the students considered. More precisely, RQ2.P2 refers to the students of a single university whereas RQ2.P3 considers all the students of the JCC. Concerning RQ2.P3, the results show that most of the included studies did not give evidence of whether the student attended the course in class or remotely. Few studies explicitly reported whether the students attend the course remotely (Rusu et al., 2009) (Verkamo et al., 2005) or in class (Ciccozzi and Crnkovic, 2010) (Brereton et al., 1998b).

RQ3: How is the assessment managed in JCC?

A university course comprises three phases: i) planning, ii) lecturing, and iii) assessing students. This research question aims to study how teachers perform students' assessments in the context of the JCC. The following table shows the parameters used to answer this research question.

PARAMETER ID	PARAMETER DESCRIPTION	POSSIBLE VALUES
RQ3.P1	Type of assessment	Oral Examination, Open-ended Questions, Multiple Choice Questions, project presentation
RQ3.P2	Assessment implementation	Open
RQ3.P3	Location of examination	Remote, in class
RQ3.P4	Number of members per team	Open
RQ3.P5	Assessment person	Home professor, foreign professor, both

The parameters RQ3.P1 and RQ3.P2 aim to investigate the methods used to assess student knowledge. As already mentioned, most of the selected studies describe distributed software development courses. In this context, the students work as a part of a dispersed team to develop a software project. The goal is to make them work together in teams to put theoretical knowledge into practice.

The analysis of the parameter RQ3.P3 allows us to examine whether the students are in the presence or remotely during the assessment phase. Similar to RQ2.P3, there is no evidence that the student attended the examination in class or remotely.

Co-funded by the
Erasmus+ Programme
of the European Union

Finally, the parameter RQ3.P5 aims to describe the teacher responsible for student evaluation in the JCC context. The selected possible values are i) home professor, ii) foreign professor, and iii) both. Even if the studies illustrate distributed software development courses with project assignments, no one describes who is accountable for the evaluation.

RQ4: How to prevent dropout of courses?

This research question aims to discover the technique used to discourage students from dropout JCC. The following table shows the parameters and the associated possible values that allow us to answer this research question.

PARAMETER ID	PARAMETER DESCRIPTION	POSSIBLE VALUES
RQ4.P1	Dropout prevention of courses strategy	Open

Despite the authors of the included studies discussing the experience in planning and running collaborative courses in software engineering, few works investigate the dropout of courses. In (Gotel et al., 2008a), the authors presented a six-year experience in offering a global software engineering course between two to four universities. During one of the considered courses, one instructor did not travel to a one-course location for the introductory session. The authors report that students from this location never were fully integrated. Thus, only seven over twelve students continued attending the course and only one fully completed all parts. This result shows that the face-to-face introductory part is essential for course completion. In addition, the authors considered that having teams with at least two members from each site participating in the project is useful for dropout prevention. The authors in (Nordio et al., 2011b) share the lesson learned in a four-year delivery of a course on distributed software development. In 2007 and 2008, the teachers encountered difficulties with whole teams leaving the course project due to a lack of motivation or incapacity to manage the language barrier. On one hand, the teachers applied some strategies to prevent the side effects on the project derived from the students' or the entire team's escape:

- Set up the team size to three or more students. Thus, if a student quits during the course, the remaining member can continue on a reduced part of the project.
- If the whole team leaves the course, they proceed with group rearrangement. Alternatively, they drop the related subsystem of the project.

On the other hand, the teachers decided to offer the course only to a restricted number of students based on the quality of their application. The results show that no student left the project. In addition, the applied strategy improved the quality of the requirements documents and the implementation.

RQ5: What is the teaching method?

The aim of this research question is to collect teaching methods referred to principles used by teachers to enable student learning. The following table shows the parameters used to answer this research question.

Co-funded by the
Erasmus+ Programme
of the European Union

PARAMETER ID	PARAMETER DESCRIPTION	POSSIBLE VALUES
RQ5.P1	Teaching/Learning method	Student-centred, teacher-centred

There are two main categories in teaching methods, i) teacher-centred approach and ii) student-centred approach. In the first case, the teacher functions in the familiar role of classroom lecturer, presenting information to the students that passively receive knowledge from their teachers. In the latter, the teacher is still the classroom authority figure. However, it functions as a coach or facilitator as students embrace a more active and collaborative role in their learning by working within groups on specific projects.

The analysis of the parameter RQ5.P1 shows that 73% of the courses involved student-centred methods in their class instruction. In this context, students from different universities work together on the same project led by instructors. Teaching-centred approaches for collaborative courses are presented in 6% of the included papers (Bosnic et al., 2011)(Brereton et al., 1998a)(Salah et al., 2018). In addition, 18% of the included studies applied both methods.

Conclusion

In this section we performed a SLR to analyse the literature related to collaborative teaching methods in computer science to create joint creative classrooms. The final outcome of the study showed that universities across the globe are more interested in creating collaborative courses in DSD, which encourage students to work together and simulate real-world distributed software projects. However, they are less inclined to combine existing theoretical course material, and as a result, they do not require the implementation of course combination strategies. Instead, they tend to adopt student-centred teaching methods, where students learn by doing with minimal teacher involvement. Furthermore, the study found that JCC or DSD courses have matured significantly, with courses often running for a single iteration due to various challenges such as skill differences between students, cultural and time-zone differences, and language barriers. Students enrolled in JCC that developed between two or more universities that are geographically distributed attend classes online or collaborate from their respective universities.

The results of the systematic literature review presented in the previous section highlight the need to pay close attention to several parameters when designing collaborative courses. These parameters include differences in time zones, language barriers, students' skill levels, available hours for collaboration, and academic curricula. It is essential to take these factors into consideration to ensure that the collaborative courses are effective and beneficial for all students involved. By addressing these challenges, educators can create an environment that fosters collaboration and provides students with the tools and resources necessary to succeed in a globalised world. Therefore, it is important for educators to carefully plan and execute collaborative courses, keeping in mind the diverse needs of their students and the challenges that come with cross-cultural collaboration. Ultimately, by focusing on these parameters, educators can create an environment that promotes successful collaborative learning experiences for students.

Co-funded by the Erasmus+ Programme of the European Union

4. JCC Design

As briefly introduced above, the creation of a JCC requires keeping into account a number of conditions, among them, the language of instruction, the period of teaching, the type of course, the number of teaching hours, and so on. In order to take into account all these tasks, we design a JCC Design process that is graphically summarized in Figure 4.1 below:

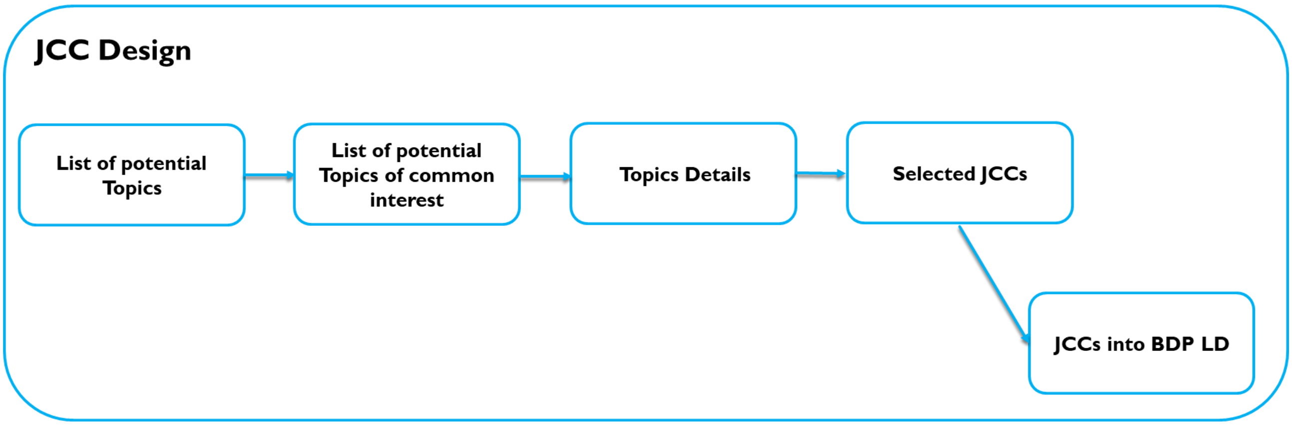


Figure 4.1. JCC design

List of Potential Topics

First of all, we had to identify the topics offered by the five universities involved in the JCCs: the University of Zagreb, University of Belgrade, University of L'Aquila, Tallinn University, and University of Žilina.

Instead of listing and focussing on "courses" offered by the partner universities, we wanted to focus on "topics" representing an abstraction and facilitating the identification of common teaching interests by the partner universities.

For this purpose, a shared online spreadsheet has been created, and filled in by the JCC partners Universities, including (a subset of) the topics though at the five Universities that may become part of a JCC. The topics have been selected by focusing on Computer Science courses, and after having received the lecturer's interest to participate in the JCC program. At this stage, we only listed the topics names and their respective teachers.

We ended up with 28 potential topics that could be offered as JCCs, as reported in Table 4.1 below:

List of potential topics	University of Zagreb	University of Belgrade	University of L'Aquila	Tallinn University	University of Žilina
(Modern) Web Application Development	Dragutin Kermek				Patrik Hrkút
Accessible Computing				Vladimir Tomberg	

Co-funded by the Erasmus+ Programme of the European Union

Advanced Database Systems	Markus Schatten				Karol Matiasko, Michal Kvet
Agile Software Development		Marko Petrović / Nina Turajlić			Marek Tavač
Artificial Intelligence: Key Ingredients (for software developers)	Markus Schatten				Michal Gregor
Big, Open and Linked Data	Jasminka Dobsa? Sandra Lovrencic?			Sigrid Mandre	
Biometrics: Fundamental Concepts	Petra Grd				
Collective Intelligence in Socio-Technical Systems				Kai Pata	
Computational Social Science		Veljko Jeremić / Milica Maričić			
Computer Games Development	Mladen Konecki				
Computer Networks	Nikola Ivković				Pavel Segeč / Jana Uramová / Marek Moravcik / Jozef Papan
Data Analytics (for software developers) /Machine Learning	Marcel Maretić / Damir Horvat	Boris Delibašić / Sandro Radovanović			Lukáš Falát / Luboš Buzna / Michal Gregor
Data Driven Decision Making and Change Management in Organisations	Nina Begicevic Redjep?			Sirje Virkus	
Database Systems					Karol Matiasko, Michal Kvet
Geographic Information Systems					Peter Márton
Information Retrieval and Data Mining	Dijana Oreški?			Sirje Virkus	
Intelligent Systems and Robotics Laboratory			Giovanni De Gasperis		
Interaction Design Methods	Dijana PV			Hans Põldoja	
Internet of Things: Embedded Software Development	Boris Tomaš				Michal Hodoň
Internet Security (or/and	Petra Grd				Jana Uramová

Co-funded by the Erasmus+ Programme of the European Union

IoT security)					
Mobile Software Development	Boris Tomaš / Zlatko Stapić				
Modeling and Computer Simulation		Aleksandar Marković/Nikola Zornić			Peter Márton
Programming Languages / Functional Programming Languages		Marko Petrović / Nina Turajlić			
Social And Behavioural Mining			Giovanni Stilo		
Software Architectures Project			Henry Muccini		
Software Architectures			Henry Muccini		
Usability,user experience, user modelling	Igor Balaban / Aleksandra Sobodić				
User Eperience Evaluation	Aleksandra Sobodic / Igor Balaban			Mati Mõttus	

Table 4.1: lis to potential topics

List of potential topics of common interest

Once the list of potential topics has been finalized, we asked the JCC partner Universities their interest in actively considering contributing to a JCC involving the selected courses. As a step ahead with respect to the first analysis, we asked the JCC partner universities if they might be interested in activating one of the topics offered by other universities. The overall idea of JCC is, in fact, not only to jointly provide existing courses, but also to activate (in collaboration with other partner universities) new topics not currently available.

The results have been collected in a new online spreadsheet, reported in Table 4.2 below (in orange, , topics not yet offered, but of interest):

ID	topics	University of Zagreb	University of Belgrade	University of L'Aquila	Tallinn University	University of Žilina
1	(Modern) Web Application Development					
2	Accessible Computing					
3	Advanced Database Systems					
4	Agile Software Development					
5	Artificial Intelligence: Key					

Co-funded by the Erasmus+ Programme of the European Union

	Ingredients (for software developers)					
6	Big, Open and Linked Data					
7	Biometrics: Fundamental Concepts					
9	Computational Social Science					
11	Computer Networks					
12	Data Analytics (for software developers) /Machine Learning					
13	Data Driven Decision Making and Change Management in Organisations					
14	Database Systems					
15	Information Retrieval and Data Mining					
16	Intelligent Systems and Robotics Laboratory					
17	Interaction Design Methods					
18	Internet of Things: Embedded Software Development					
19	Internet Security (or/and IoT security)					
20	Mobile Software Development					
21	Modeling and Computer Simulation					
21 b	Programming Languages / Functional Programming Languages					
22	Social And Behavioural Mining					
23	Software Architectures					
24	Software Architectures Project					
27	Virtualization and cloud					

Co-funded by the
Erasmus+ Programme
of the European Union

	technologies (+ orchestration, automation)					
29	Introduction to Web Technologies					
30	Multiagent systems					
32	Software Engineering					
37	Trust in Computing					
38	Development of Interactive Systems					
39	Prototyping					
40	Software Quality Engineering					
41	Programming Languages / Functional Programming Languages					

Table 4.2: list of potential topics of common interest

Topics Details

For each topic of interest, we started collecting a number of information, necessary for the concrete realization of a JCC, and mainly:

- Language of instruction
- If known:Start date of the course (month/day)
- If known: End date of the course (month/day)
- Number of ECTS credits
- Professors
- Link to Syllabi
- # of teaching hours (lect.-sem.-pract.)
- # of students
- Type of course: Theory, Lab, Theory and Lab
- Part of the Study Program or ad hoc
- Facilitators names

We ended up with six different languages of instruction (Croatian, Serbian, Slovak, English, Italian, Estonian), start and end dates of existing courses that may span towards different semesters or universities implementing different trimester, quadrimester, semester systems. Number of ECTS

Co-funded by the
Erasmus+ Programme
of the European Union

credits, on the same topic, may also vary from two to six, with a different number of teaching hours dedicated to lectures, seminars or practice. Number of students, on the same topics, could also vary from 80 to 220 (Software Engineering topic).

At this stage, we also identified one responsible for each topic offered in each University. We call them facilitators, and they are responsible to monitor the progress on eventually selected JCCs.

Selected JCCs

Based on the topics details, we selected 13 topics that may become JCCs. For each of them, the assigned facilitators interacted with the teachers from their own universities, and prepared an implementation plan. At this stage, facilitators and teachers have been asked to share the local courses sillabi (when the course was already locally offered) or the expected sillabi (for those courses to be activated). Each JCC has been asked to fill in a document including the following information:

- Teacher(s) name and email
- Q: What is the starting date of the JCC? Duration?
- Q: What is the number of students who will participate in the JCC per partner institution?
- Course content/syllabus (a link to a doc will work as well)
- University interest in joint activities (i.e., which benefits you expect to get from this collaboration?)
- Any Open questions or Challenges to be solved (i.e., do you see any difficulty to make this cooperation working?)
- Joint Creative Classroom Content model (i.e., are you planning to create an ad hoc fully joint course, or to share some of the content of your course with other courses, or any other content sharing model, or what?)
- Q: Do you have a Learning Management System (e.g., Moodle, Schoology) or related system in mind that you use to give access to JCC partner students?
- Joint Attendance Model. Question: are students jointly attending shared lectures?
  Please describe the joint attendance model (i.e., are students from both courses attending all lectures, or are they attending only some lectures from both courses, or are they only attending their original course, or what?)
- Joint Assessment Model. Question: Are students jointly working on a shared project? Please explain the joint assessment Model (i.e., are students from both universities working on a shared project, or are them completing separate parts of a common assignment, or what?)
- JCC Study Plan: Which topics will be covered & by which University?

An example is reported below.

Software Architectures Analysis and Design

Co-funded by the Erasmus+ Programme of the European Union

academic year: 2022-2023, semester I		
	University of Zagreb	**University of L'Aquila**
BDP tool link: (58 hours planned)	https://learning-design.eu/en/preview/81f9a72eb213cf3e3d1b2ea2/details	
Teacher(s) name and email	Zlatko Stapic zlatko.stapic@foi.hr	Henry Muccini henry.muccini@univaq.it
Q: What is the starting date of the JCC? Duration?	Sept-Dec 2022	
Q: What is the number of students who will participate in the JCC per partner institution?	about 20 students, 2ECTS	
Course content/syllabus (a link to a doc will work as well)	course name: **Software Analysis and Design** Agile development & SCRUM Sw development trends (sota and motivational) Specifics of development (mobile, smart, etc) Solid low level design of software products (architecture, combination of patterns) NFR DevOps, testing and monetization more info at https://docs.google.com/spreadsheets/d/1oXWHXO8GKW86ECU-r60FXqKZ3bf69UPLmczs_69XH6c/edit?usp=sharing	course name: **Software Architectures** Components and Connectors Architecture Design Decisions Software Architecture Styles Views and Viewpoints Cloud Architectures Self-Adaptive Architectures Technologies for Architectural design and implementation more info at https://docs.google.com/spreadsheets/d/1oXWHXO8GKW86ECU-r60FXqKZ3bf69UPLmczs_69XH6c/edit?usp=sharing
University interest in joint activities (i.e., which benefits you expect to get from this collaboration?)	I can complement the content of my Software Analysis and Design course with those coming from the Software Architectures one	I can complement the content of my Software Architectures course with those coming from the Software Analysis and Design one

Co-funded by the Erasmus+ Programme of the European Union

Any Open questions or Challenges to be solved (i.e., do you see any difficulty to make this cooperation working?)	all issues have been solved during the February meeting	all issues have been solved during the February meeting
Joint Creative Classroom Content model (i.e., are you planning to create an ad hoc fully joint course, or to share some of the content of your course with other courses, or any other content sharing model, or what?)	We are essentially SHARING some of the lectures from one course into the other. More specifically: UnivAQ Software Architecture course is receiving Agile Development and SCRUM lectures (2 lectures). FOI Software Analysis and Design course is receiving some introduction to Software Architecture design (2 lectures)	
Q: Do you have a Learning Management System (e.g., Moodle, Schoology) or related system in mind that you use to give access to JCC partner students?	Moodle	Teams and Schoology
Joint Attendance Model **Question: are students jointly attending shared lectures?** **Please describe the joint attendance model (i.e., are students from both courses attending all lectures, or are they attending only some lectures from both**	Answer: they are jointly attending a few shared lectures UnivAQ's students are attending SCRUM lectures from Prof. Stapic FOI's students are attending Introduction to Software Architecture lectures from Prof. Muccini Other than this, UnivAQ's students are attending the full Software Architectures course. FOI's students are attending the Software Analysis and Design course. They both are then knowledgeable about basics of SCRUM and Software Architectures, while FOI's students are expert on Software Analysis and Design and UnivAQ's students are expert of Software Architectures.	

Co-funded by the Erasmus+ Programme of the European Union

<table>
<tr><td>courses, or are they only attending their original course, or what?)</td><td colspan="2"></td></tr>
<tr><td>Joint Assessment Model.
Question: Are students jointly working on a shared project?
Please explain the joint assessment Model (i.e., are students from both universities working on a shared project, or are them completing separate parts of a common assignment, or what?)</td><td colspan="2">Answer: six teams are joint, the other are disjoint.

We are selecting six students @UnivAQ, and 18 students @FOI.
We are creating six joint teams with members from UnivAQ and FOI.
They will run a specific project: a software architecture project that uses SCRUM, and an Analysis and Design project that uses Software Architecture design.</td></tr>
<tr><td>JCC Study Plan: Which topics will be covered & by which University?</td><td>course name: Software Analysis and Design (with Software Architecture)

Agile development & SCRUM
Sw development trends (sota and motivational)
Specifics of development (mobile, smart, etc)
Software Architecture Design (lectures provided by UnivAQ)
Solid low level design of software products (architecture, combination of patterns)
NFR
DevOps, testing and monetization</td><td>course name: Software Architectures (with SCRUM)

Components and Connectors
Agile and SCRUM (lectures provided by FOI)
Architecture Design Decisions
Software Architecture Styles
Views and Viewpoints
Cloud Architectures
Self-Adaptive Architectures
Technologies for Architectural design and implementation</td></tr>
</table>

Table 4.3: Software Architecture Analysis and Design JCC

Co-funded by the Erasmus+ Programme of the European Union

Moreover, for each selected JCC, we asked the respective teachers to fill in the information requested in the BDP LD tool, presented in the IO2 deliverable. An example, follows.

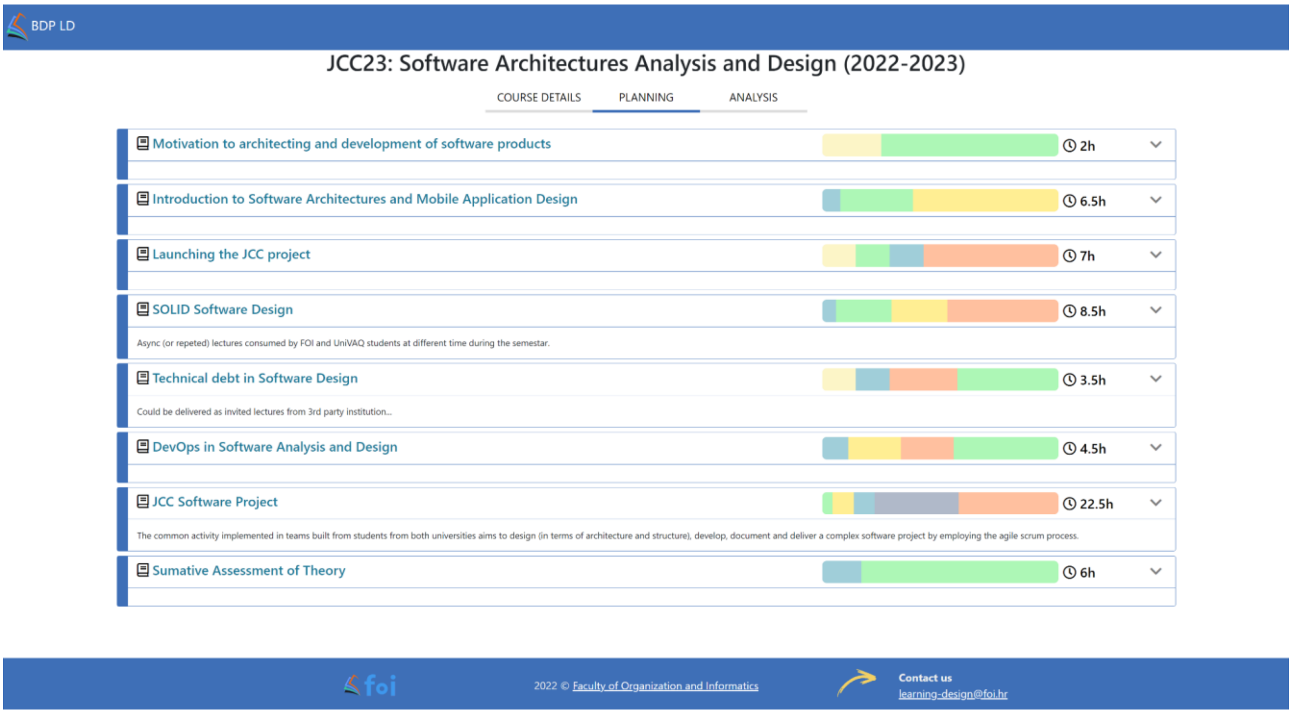


Figure 4.2. Information about JCC requested in a BDP tool

5. JCC Implementation

After further considerations and analysis, TEACH4EDU4 ended up with eight implemented JCCs, listed below:

- Advanced Database Systems,
- Information Retrieval and Data Mining,
- Internet of Things: Embedded Software Development,
- Internet Security and Trustworthiness,
- Machine Learning School,
- Modeling and Computer Simulation,
- Software Architectures Analysis and Design,
- Robotics: Embedded Software Development.

Advanced Database Systems (ADS)

Co-funded by the Erasmus+ Programme of the European Union

Overview:
Number of ECTS: 3
Synchronous workload: 35
Asynchronous workload: 51
Dates: 10/2022-01/2023
Language: English
Number of participants: 86

Goal of JCC:
The aim of this course is to acquaint students with the theory and technology needed to implement advanced database systems including relational, temporal, deductive, object-oriented, active and graph databases. In addition, various indexing, partitioning, optimization, and denormalization techniques will be covered. Students will learn about new trends and open questions in the field of database theory.

Learning outcomes:

- conduct database analytics through practical work in SQL,
- use basic concepts of data flow systems,
- implement various database optimization techniques,
- implement database indexing and partitioning techniques,
- apply different optimization techniques in complex database applications,
- manage temporal data,
- analyze conflicts in active databases,
- implement a data flow application using the default data flow system,
- analyze databases using a specific data model,
- implement an active database using the default database management system,
- design and implement a complex database application.

Learning outcomes assessment:
Forming teams for the implementation of the student's projects dealing with advanced database systems. Students will need to implement a complex example database, generate adequate data, implement a contextual application, present their work and write a report.

Prerequisites for the JCC:
Students should have completed the elementary database course

Number and type of assessment:
Students will work on students' projects and take the final exam (test)

Teachers:
University of Zagreb: Markus Schatten
University of Belgrade: Srđa Bjeladinović, Milica Škembarević, Slađan Babarogić
University of Žilina: Michal Kvet

Co-funded by the Erasmus+ Programme of the European Union

Information Retrieval and Data Mining (IRDM)

Overview:
Number of ECTS: 2
Synchronous workload: 30
Asynchronous workload: 30
Dates: 10/2022-01/2023
Language: English
Number of participants: 17

Goal of JCC:
The aim of the program is to introduce the discipline of data mining and information retrieval, and to enable students to apply methods of i data mining with an emphasis on textual data. The course is project-oriented and it is planned to solve different tasks on the same data sets for teams from the universities involved.

Learning outcomes:

- identify appropriate visualization methods and apply them to a given data set,
- explain the principles of clustering and data classification algorithms and apply them to given data sets using appropriate software,
- explain the principles of optimization of hyperparameters and selection of attributes for a given problem of grouping or classification of data and apply them in solving the problem of data analysis using the selected software,
- explain the principles of inverted indexes and compression for the problem of information retrieval,
- explain the principles of textual data representation using the bag of words model,
- apply pre-processing methods on a given set of text data and represent them in a form that will enable the application of basic tasks of text mining (visualization, grouping, classification).

Learning outcomes assessment:
Checking theoretical knowledge through quizzes, evaluating solutions to problem tasks of data mining for given data sets.

Prerequisites for the JCC:
Participants should have passed exams of mathematical courses including area of mathematical analysis and linear algebra and introductory course of probability and statistics

Number and type of assessment:
2 assessments of practical work (10 points), 3 quizzes (30 points), 2 project tasks (2*90 min joint students work, 60 points)

Teachers:
University of Belgrade: Sandro Radovanović
University of Zagreb: Jasminka Dobša

Co-funded by the Erasmus+ Programme of the European Union

Internet of Things: Embedded Software Development (ITESD)

Overview:
Number of ECTS: 2
Synchronous workload: 20
Asynchronous workload: 40
Dates: 5-6/2022
Language: English
Number of participants: 46

Goal of JCC:
IOT ESD is a short 3-week training in which students are taught to develop microprocessor-based hardware solutions. As part of the education, students are introduced to the basics of electronic circuits and the physical characteristics of various components. Solution development includes hardwaredesign and application background development in C / C ++. The microprocessor used and the basic hardware are based on the Photon Particle component. In addition to application development, students are introduced to different types of sensors and actuators that they can use when designing their projects. Thanks to the IOT features of the Photon Particle component, the training includes connecting and managing devices over the Internet.

Learning outcomes:
- interpret and use ESD development documentation,
- develop a prototype IOT application,
- program an ARM based micro-controller using C / C ++ development tools,
- develop an IoT solution and connect it to existing business systems.

Learning outcomes assessment:
Simple independent tasks that will test the mastery of basic knowledge and skills relevant to the project, work on the project within the team - design the development of a new IoT device that will send simple data to the cloud infrastructure (cloud). (e.g. create a circuit that will send the cloud storage temperature in real time), they need to develop a prototype IOT solution and application. (e.g. create a prototype system that will record cloud temperatures in real time with an average display on mobile devices), the student team should develop a part of the IOT solution within the project and connect it with the existing business systems. (e.g. create a product counter in the production business system and integrate it with the existing delivery note system).

Prerequisites for the JCC:
Undergraduate or graduate students with basic C/C++ or higher language skills

Number and type of assessment:
One written exam, one team project presentation and evaluation

Co-funded by the Erasmus+ Programme of the European Union

Teachers:
University of Zagreb: Boris Tomas
University of Žilina: Michael Hodon

Internet Security and Trustworthiness (IST)

Overview:
Number of ECTS: 3
Synchronous workload: 35
Asynchronous workload: 55
Dates: 9/2022 – 01/2023
Language: English
Number of participants: 25

Goal of JCC:
The program has guided students toward the safe use of web resources through an in-depth understanding of core technologies and security issues from three key perspectives - human, computer networks, and web technologies.

Learning outcomes:

- summarize concepts, theories and research within the domain of trust in technology,
- explain the types of data used in network security monitoring, the alert assessment process and the classification of intrusion events with a diamond model,
- use the knowledge gained to solve the Capture-The-Flag (CTF) challenge,
- apply NIST incident management procedures in response to an incident,
- analyze common vulnerabilities in network and web technologies, attacks, and data for the purpose of identifying warning sources,
- assess vulnerability assessment tools,
- create a virtual laboratory environment for conducting simulated penetration testing.

Learning outcomes assessment:
Self-assessments, automated checks, monitoring through CTF activities.

Prerequisites for the JCC:
Basics of computer networks

Number and type of assessment:
Self-assessments, automated checks, monitoring through CTF activities

Teachers:
University of Zagreb: Igor Tomičić
Tallinn University: Sonia Sousa
University of L'Aquila: Walter Tiberti
University of Žilina: Jana Uramova

Co-funded by the Erasmus+ Programme of the European Union

Machine Learning School (MLS)

Overview:
Number of ECTS: 6
Synchronous workload: 132
Asynchronous workload: 48
Dates: 9/2022 – 1/2023
Language: English
Number of participants: 20

Goal of JCC:
A course in modern machine learning methods that covers theoretical and practical aspects.

Learning outcomes:
- explain basic concepts in the field of machine learning such as: machine learning, implicit and explicit representation of knowledge, local and global generalization, overfitting, bias, variance, regularization and others,
- explain the principle of basic machine learning methods,
- assess where and how machine learning methods can be applied,
- apply machine learning methods and approaches,
- identify machine learning problems and look for appropriate modern methods.

Learning outcomes assessment:
Evaluation of mandatory tasks and final exam.

Prerequisites for the JCC:
Participants should have basic programming skills, e.g. Python; a background in basic algebra, analysis - for the majority of modules, they are not strictly required.

Number and type of assessment:
Student assessment is going to be three-fold: based on select activities in each module (such as Google Colab notebooks and quizzes), the team project and the final exam

Teachers:
University of Belgrade: Sandro Radovanović
University of Žilina: Milan Straka, Michal Gregor

Modeling and Computer Simulation (MCS)

Overview:
Number of ECTS: 2
Synchronous workload: 35
Asynchronous workload: 25
Dates: 4/2022 - 6/2022
Language: English
Number of participants: 32

Co-funded by the Erasmus+ Programme of the European Union

Goal of JCC:
The goal of the JCC is to work together and exchange knowledge of teachers and students on modeling and computer simulations.

Learning outcomes:
- compare simulation experiments and conduct sensitivity analysis,
- identify system problems that can be solved using modeling and computer simulation,
- identify the input parameters of the model from the system observation,
- evaluate the results of the simulation in order to improve the system,
- formulate simulation experiments.

Learning outcomes assessment:
Self-assessments and monitoring through collaborative activities.

Prerequisites for the JCC:
There are none

Number and type of assesment:
3 quizes and 3 projects

Teachers:
University of Belgrade: Aleksandar Marković and Nikola Zornić
University of L'Aquila: Vittorio Cortellessa
University of Žilina: Peter Márton

Software Architectures Analysis and Design (SAAD)

Overview:
Number of ECTS: 2
Synchronous workload: 17
Asynchronous workload: 43
Dates: 10/2022 - 2/2023
Language: English
Number of participants: 24

Goal of JCC:
The aim of this program is to expand the knowledge and skills of students from partner universities who receive in the home courses Analysis and Development of Program and Software Architecture. These courses are disjoint, but complementary and allow students from both universities to expand their knowledge at the theoretical and practical level in areas that complement each other.

Learning outcomes:
- model the entire software architecture of the system and design a mobile application,

Co-funded by the Erasmus+ Programme of the European Union

- apply practices and principles of agile development process according to the development methodology "SCRUM",
- develop mobile application and background services,
- prepare documentation for architectural design and prototyping product,
- explore current trends related to architecture and software product development,
- explore DevOps tools and practices,
- determine the appropriate SOLID software design concepts to be used in the project,
- assess the technical debt of the implemented software project.

Learning outcomes assessment:
Verification of learning outcomes would be conducted through formative and summative tests of knowledge. Formative checks would be: self-assessment of knowledge in the form of automated tests/quizzes and direct feedback, self-assessment of knowledge through the application and implementation of the activities covered in the practical project. Summative knowledge tests would include: evaluation of the project application with teacher feedback, evaluation of architecture design with teacher feedback, examination of theoretical knowledge in the form of a classical exam, evaluation of the finished project in the form of presentation and defense of the project.

Prerequisites for the JCC:
Students' need to have attended a basic Software Engineering course. Better if they also have good programming skills in Java, Kotlin or C#

Number and type of assessment:
Students will have to work together on a shared project. UnivAQ students will focus more on the architectural aspects of the shared project. FOI's students will focus more on the mobile app implementation of the selected joint project.

Teachers:
University of Zagreb: Zlatko Stapić
University of L'Aquila: Henry Muccini

Robotics: Embedded Software Development (RESD)

Overview:
Number of ECTS: 3
Synchronous workload: 30
Asynchronous workload: 59
Dates: 04/2022 -04/2022
Language: English
Number of participants: 25

Goal of JCC:
In the course, students should get the basic knowledge about the embedded systems development. They should get in touch with the C/C++ programming of special embedded systems with

Co-funded by the Erasmus+ Programme of the European Union

ATmega328 MCU used as the control unit. IDE Atmel Studio is used for this purpose. The students will program the robot controlled through Bluetooth interface via the smartphone application.

Learning outcomes:

- acquiring the basic knowledge about embedded systems development,
- learning how to assemble control boards driven by AtMega328,
- how to program control boards driven by AtMega328,
- how to integrate control boards driven by AtMega328, into the mobile robotics.

Learning outcomes assessment:
The students will present the functionality of the assembled robot by driving its movement through the mobile-phone application.

Prerequisites for the JCC:
Basics of physics, basics of electrotechnics, basics of programming

Number and type of assessment:
Final project - student work, on their own project, which will be evaluated

Teachers:
University of Zagreb: Boris Tomaš
University of Žilina: Michal Hodon

6. JCC lived experiences and evaluation

This section consists of two distinct parts. 6.1 will focus on the lived experiences of the 275 students who participated in the 8 JCCs, of whom 123 (48%) students completed a JCC successfully. 6.2 will focus on the reflections from 16 teachers who were involved in the eight JCCs.

6.1 Lived experiences by students

Participants

275 students were enrolled in the eight JCCs. In total 162 participants completed the pre-survey, while 108 participants completed the post-survey, which is a response rate of 59% and 39% respectively, and in total 222 responses of participants were recorded (81%). 33% of participants identified as female, and the average age of participants was 24.09 (SD = 2.44, range 21-37). 124 (56%) students were from Croatia (University of Zagreb), followed by 38 (17%) of students from Slovakia (University of Zilina), 34 (15%) studying at Serbia (University of Belgrade), 11 (5%) from Italy (University of L'Aquila), 10 (5%) from Estonia (Tallinn University), and five preferred not to say.

Co-funded by the Erasmus+ Programme of the European Union

Instruments

Pre-survey

Using principles of Kirkpatrick's four phases evaluation model students at the start of the JCC completed a pre-survey of 18 items (phase 2 learning). Twelve items were related to students' expectations of the JCC, and how these were related to the nine core principles of Education 4.0 (e.g., "I would like to learn where and when I choose", "It would be great if the course is personalised to my learning", "I would like to work on projects with other students", and "I would like to learn how to interpret and reason with data"). A Likert-Response scale of 1-4 (1 = Totally Disagree, 2 = Disagree, 3 = Agree, 4 = Totally agree) was used. A three factor structure was identified from the survey related to hands-on experience, flexible learning, and collaborative activities, with cronbach alphas ranging between .615 - .710, indicating reasonable reliability.

Post-survey

The same twelve items were asked at the post-test, with a notable difference that "would" was replaced with "could", as well as whether or not specific learning outcomes were achieved according to the students. Furthermore, each JCC had specific learning outcomes related to their specific focus (between 2-8 items). For comparability purposes, in this report we will only report on the Education 4.0 items. The cronbach alphas were .605-.796, indicating reasonable reliability. For those respondents who completed the pre- and post-test and who could be matched in terms of coded anonymity (i.e., n = 48), we computed subsequent learning gain scores. In the first wave of data collection (April - October 2022) participants were able to complete the survey completely anonymous, which prevented us from linking individual responses together. Therefore, in the second wave of data collection (June 2022- February 2023) we asked participants to indicate the first letter of their first and last name, the number of the place they live, and the first two letters of the town/city they live. This allowed us to link pre- and post-test responses together without compromising participants identity (as the research team did not have access to names of participants and their home addresses).

Findings

In terms of the pre-survey, as indicated in Figure 6.1 on the twelve items of the pre-test 95% of participants agreed that they wanted to have these "innovative" Education 4.0 elements in their learning design (pre-average). In terms of the twelve items, in order of preference all but two students (i.e., 99%) wanted to develop their skills to work effectively in a modern workplace, 98% wanted to gain hands-on, authentic experiences and real-world skills, 98% wanted to learn how to interpret and reason with data, 96% wanted learning and teaching to emphasise skills and competences in a modern workplace, 96% wanted the assessments to be innovative, 93% wanted to learn where and when they chose, 93% wanted to develop their softer skills, 92% wanted to become more independent, 90% wanted the course to be personalised to their learning, 85% wanted to work on projects with other students, while only 70% wanted to contribute to the design and implementation of the JCC, and only 69% wanted to study how they would like to study (rather than a predetermined structure by their teacher). In other words, students had high expectations at the beginning of the JCCs in terms of effective and real-world skills development and flexibility in terms of how to study.

Co-funded by the
Erasmus+ Programme
of the European Union

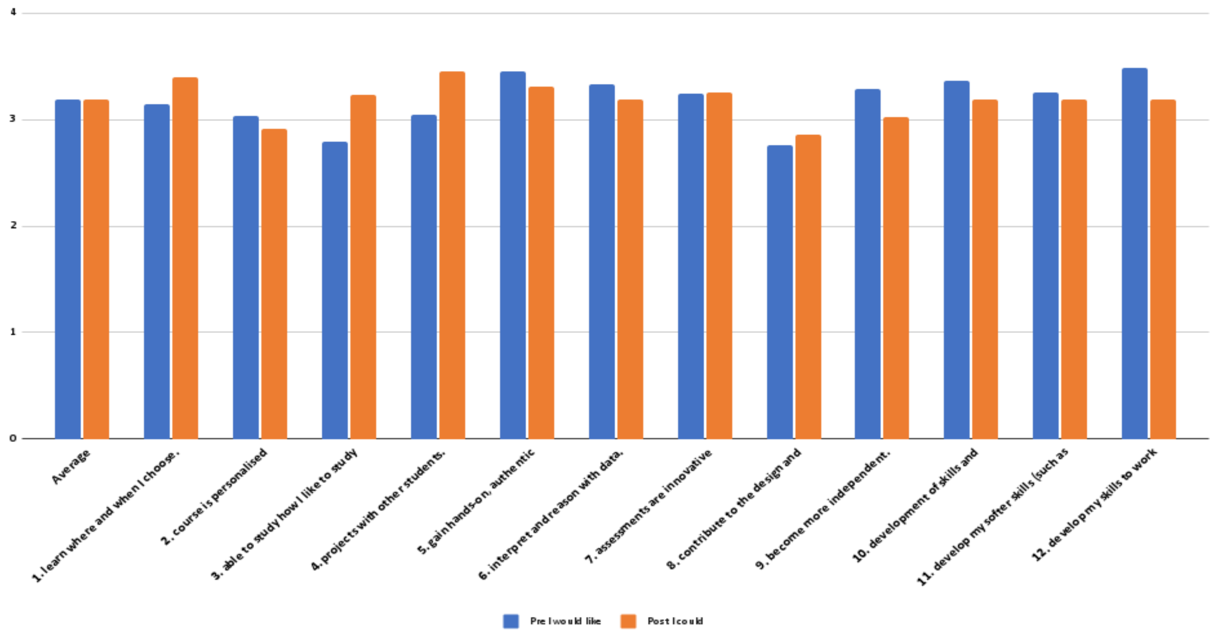


Figure 6.1 Pre- and Post-test results of Education 4.0 (npre = 162, npost = 108)

As indicated in Figure 6.2, subtle differences were present between the JCCs in terms of expectations. In five Education 4.0 items (i.e., project work, interpret data, innovative assessments, contribute to design, skills modern workplace) these differences were statistically different using ANOVAs. Subsequent analyses per institute (not illustrated) indicated no significant differences in expectations with the exception of innovative assessments and skills in the modern workplace.

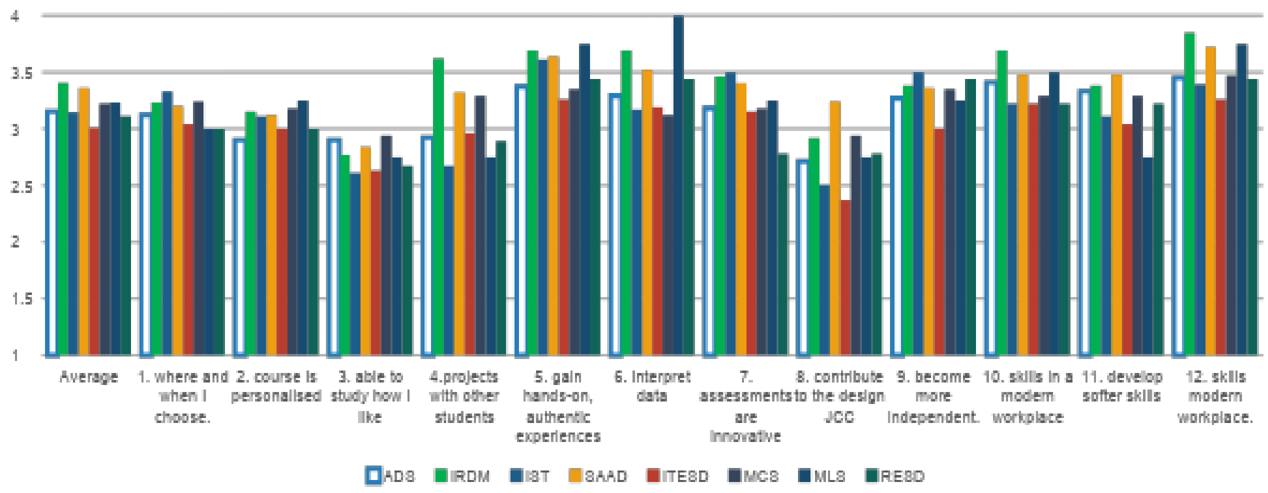


Figure 6.2 Pre-test expectations of Education 4.0 design elements per JCC.
The JCCs acronyms are defined in Section 5.

Co-funded by the
Erasmus+ Programme
of the European Union

In terms of the post-survey, 108 participants completed the survey, of which 48 could be linked to the pre-test results based upon returned results from 4 JCCs. As indicated in Figure 6.1 on the twelve items of the post-test 90% (-5%) of participants agreed that they received these “innovative” Education 4.0 elements in their learning design (post-average). In terms of the 12 items, in order of preference, 96% (=) received learning and teaching that emphasised skills and competences in a modern workplace, 95% (+2%) indicated that they could learn where and when they chose, 95% (+26%) could study how they would like to study, 92% (+7%) worked on projects with other students, 93% (-5%) learned how to interpret and reason with data, 91% (-7%) gained hands-on, authentic experiences and real-world skills, 93% (-3%) worked with the assessments to were innovative, 90% (-3%) were able to develop their softer skills, 90% (-9%) of students wanted to develop their skills to work effectively in a modern workplace 84% (-8%) have become more independent in how they study, 80% (-10%) indicated that the JCC was personalised to their learning,, while 70% (=) indicated that their voice was used to contribute to the design and implementation of the JCC.

As mentioned previously, as students had extremely high expectations at the beginning of their respective JCC it is noteworthy to report that on most elements of Education 4.0 the JCCs actually delivered, or perhaps even exceeded, students’ expectations. In particular it is recommendable that students felt that the JCCs provided skills and competences of the modern CS workplace, that there was flexibility in terms of how they could study and work together. At the same time, probably because of the relatively new innovation fewer students felt part of the (co-)creation and implementation of the JCCs, which given the context in which teachers from different institutions had to collaborate on a relatively short notice together is not entirely surprising.

As indicated in Figure 6.3, on average the high pre-expectations of students were met with the four JCCs in terms of the post-survey. While on average the JCCs were slightly less hands-on relative to initial expectations, both the learning gains in terms of flexibility to study and learn substantially improved as well as the opportunity to collaboratively work with peer students.

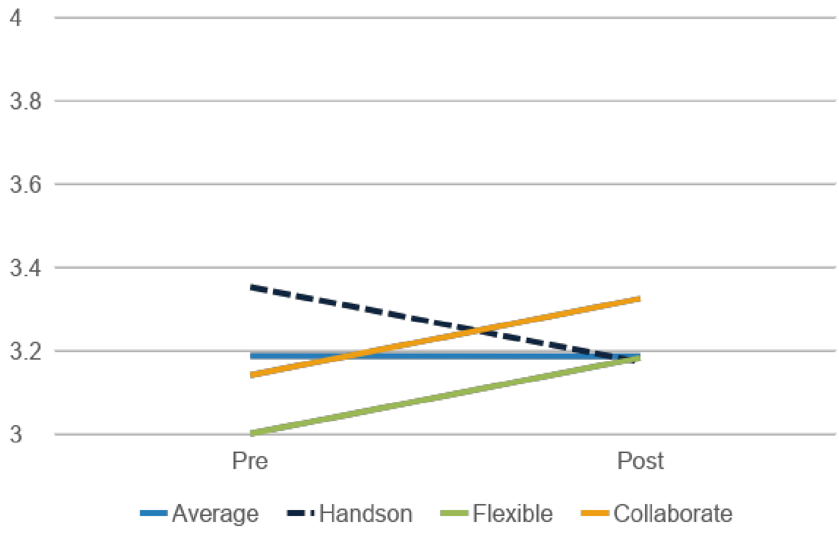


Figure 6.3 Average Learning gains in the four JCCs.

Co-funded by the
Erasmus+ Programme
of the European Union

In terms of differences between the four JCCs, there were only significant differences in terms of project work and softer skills. Specifically, ADS participants were relatively more positive about the lived experiences at the end of the JCC relative to their initial expectations, while for the other three JCCs expectations were not entirely realised.

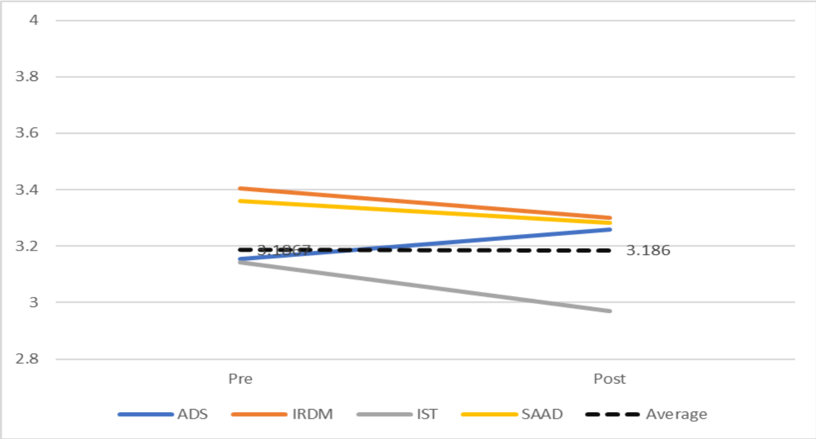


Figure 6.4 Average learning gains in four JCCs

On average, around a third of participants who completed both the pre- and post-survey had higher post-test results relative to pre-test results (i.e., a learning gain). As indicated in Figure 6.5 some participants (e.g., ADS participant on top left) made substantial learning gains during the JCC, while several other participants did not make these learning gains (e.g., participants on the bottom right of Figure 6.5). Obviously there could be a Dunning-Kruger effect as some participants might have had unrealistically high expectations at the beginning, which may or may not have materialised at the end of the JCC.

Co-funded by the
Erasmus+ Programme
of the European Union

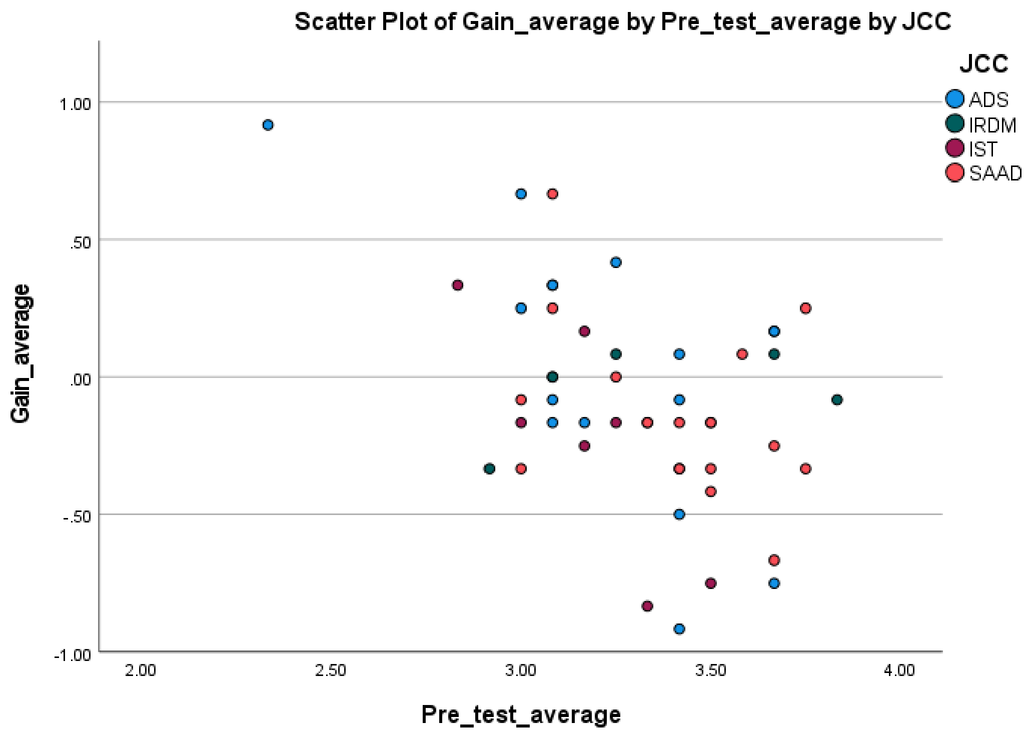


Figure 6.5 Learning gains per student

6.2 Lived experiences by teachers

As indicated by Table 6.1, in total 16 Computer Science (CS) teachers as part of eight JCCs were interviewed using a semi-structured interview approach in January-March 2023. These teachers were selected based upon suggestions from the lead contacts from the participating institutions, and were in general intensively involved in the design, implementation, and/or evaluation of the respective JCCs. The interviews were conducted either in English or in the respective local language, and interviewers of respective institutions (see Table 6.2) received a detailed interview protocol. All interviews were recorded and automatically transcribed, translated in English (if needed), and afterwards checked and coded by one of two human coders from the Open University. The transcripts and recordings were subsequently shared with the OU, and two researchers who were not involved in the JCCs independently analysed the data of the 16 transcripts.

In total 10 open questions were included in the interview, including the lived experiences of the overall experiences of the JCC (e.g., “What did you like the most about your JCC?”, “What was the least rewarding aspect of your JCC?, “If there is anything you would want to change in terms of the learning design if you would implement your JCC again?”), using the Balanced Design Planning (BDP) tool (e.g., “Did you find the learning design dashboard useful?), and future intentions (e.g., “How do you intend to use JCC in the future?).

Co-funded by the
Erasmus+ Programme
of the European Union

JCC code	JCC title	Interviews
33	Information Retrieval and Data Mining	3
12	Machine Learning School	1
21	Modeling and Computer Simulation	2
18	Internet of Things: Embedded Software Development	1
60	Robotics: Embedded Software Development	2
19	Internet Security and Trustworthiness	4
3	Advanced Database Systems	3
23	Software Architectures Analysis and Design	1
	Total	16

Table 6.1 Interviews with teachers and their respective JCCs

Universities	**Interviews**
FRI UNIZA	4
TU	1
UNIVAQ	1

Co-funded by the Erasmus+ Programme of the European Union

UNIZG FOI	6
UOB	4
Total	**16**

Table 6.2 Interviewees per participating institutions

Thematic analysis

A thematic analysis (Braun & Clarke, 2006) was conducted on the data from the 16 interviews. As indicated by Figure 6.6, two main themes (i.e., learning design, JCC experience) were identified in terms of how participating teachers reflected on their lived experiences of their JCC. In terms of learning design, five (sub)themes were identified, including design process and reflection, sharing the design amongst institutions, tool feedback and improvement, JCC model, and the EDU 4.0 template. In terms of the second main theme of JCC experience, both positive and negative aspects were thematically analysed, as well as teachers' future intentions using Nvivo 12.

Nodes

Name	Files	References
1 Learning Design	0	0
1 Design process and reflection	15	22
2 Sharing the Design among institutions	12	12
3 Tool feedback and improvement	16	31
4 JCC model designed (quantification)	10	11
5 EDU 4.0 Template (quantification)	10	10
2 JCC experience	0	0
1 Positive aspects	16	16
2 Negative aspects	16	16
3 Future use	10	11

Figure 6.6 Thematic analysis results

Co-funded by the Erasmus+ Programme of the European Union

Learning Design

In this section we specifically highlight the lived (qualitative) experiences from teachers using the BDP tool[1]. In general the overall experiences and perspectives of the BDP tool were positive by most of the 16 teachers as the BDP tool allowed participating teachers to share and discuss their initial learning design ideas with their JCC colleagues in other institutions. Furthermore, by discussing and sharing these initial learning design ideas the tool provided a platform for discussions of how to co-design the JCC, and ensure that teachers had an agreement in terms of both process, content, and learning outcomes.

In terms of **"Design process and reflection"** teachers reflect of the type of activities considering having a correct balance among them, not including too many "acquisition" type learning activities:

> *I feel like we do have quite a lot of activities on acquisition which I don't think is good. We used only four types of activities – acquisition, practice, investigation and assessment in our learning design and the number in the chart there are six. (…). I don't think we had enough on investigation. We had some content but I think we couldn't use probably more*. (12-1)[2]

Or having problems to implement what was planned:

> *Therefore, in my part of the course, they had more practical things. Due to the fact that we were physically distant, the students mostly got a second teacher from me in our part of the assignment. (…) But to be honest, I think that when it came to the implementation of our subject, I don't know if we really implemented our plan completely and faithfully. Learning design served to create some idea, but in the end the reality was a little different.* (19-1)

Given the nature of JCC, activities were online or blended with combination of synchronous sessions:

> *Content of the JCC was provided fully online. It was mostly synchronous sessions. We had special afternoon sessions scheduled for JCC, so they did not merge into regular classes. All lectures had short knowledge tests. As part of the class, they also organised teams, used programs that are normally used in classes for software development (GitHub, etc.). We had two invited lectures.* (23-1)

> *Both teachers have existing courses, so the JCC was based on them as a hybrid execution that was rather challenging. (…) Such a subject should not be performed partially online.* (60-2)

And usually including a combination of more than one teacher:

[1] Note that a detailed description of how the teachers used the BDP tool in their respective JCC has already been described in IO2.
[2] (12-1) stays for JCC 12 (so as indicated in Table 6.1), 1st teacher.

Co-funded by the Erasmus+ Programme of the European Union

> *Learning design has two main parts for our JCC. There is not one section made for each teacher, but it is divided according to the structure of the subject. When there were meetings (consultations) for semester projects, there were always at least two, but mostly three teachers. When there were situations where it was not possible, there were two of us, but we always tried to be all of us.* (3-1)

"Sharing the design among institutions" was proven to be a good experience where teachers from different institutions was positive:

> *The process went smoothly. The focus was a little different between the institutions, but the existing subjects were very complimentary and the way of organisation was quickly agreed; collaboration with teacher from [...] was much easier due to the absence of a language barrier.* (3-2)

> *I mean it really gave us a view of what we were assembling, and it forced us to think about the different kinds of content, the different kinds of activities. So, I think that's good. I would actually advise all of my colleagues to use this for their courses at the university. Because really, I don't think we might think about this enough. You know, people are a little bit resistance to new ideas and of course it's going to take time if you decide based on this to change some activities in your course.* (12-1)

> *Cooperation was possible, even if at a distance and in addition to the duties of the current semester. Since there were four of us, one did something for the other, if possible.* (19-1)

In terms of the LD "**Tool feedback and improvement**" feedback is good in terms of easiness although feedback should be included once the course has been run:

> *It is very easy to create a learning design there. But I think, it would be important to evaluate what the reality was like during the lesson and to get, for example, feedback from the students, whether what was planned was learned. (…) The teacher should also be able to make a note during the semester so that he does not forget for the following academic year what he plans to spend more or less time on.* (19-1)

While visualisation of the learning activities was considered useful it could be easier to be shared with students and embed in the LMS

> *Yes, it allowed me to visualize the planned workload but could include less detail, that way will be easier to share the plan with the students. (…) Could be integrated with the virtual learning environment so students have access to it as well.* (19-2)

> *Based on the graphical display of activities and different knowledge delivery options, this tool gives a very good insight into the designed course and allows easy modification of components in order to improve the course.* (21-1)

Co-funded by the
Erasmus+ Programme
of the European Union

> *Analytics inside the tool is handy. It provides insightful analysis after the teachers enter the activities. That way, teachers are relieved from calculating the ratio of different types of activities manually.* (3-3)

More detail for planning can be handy for some teachers and assigning tasks to particular teachers:

> *We mostly focused on the students workload dashboard and the balance of planned and real workload. It would be good to add a course performance card. In addition, distinguish 60 min and school hours (45 min). It would be good if there was a higher level of planning - how the subject fits into the entire study program. There is a lack of case implementation - who is doing what part of the plan*. (23-1)

The inclusion of potential was also suggested:

> *The tool had everything that was needed. In my opinion, there is nothing missing. But yes, for example, if I have a practical subject, for 2 credits, if the tool suggested how much time should be in which component of the learning design (based on a selected template), that would be a big help. I would only add this functionality to the tool.* (60-1)

JCC model used (quantitative) for those who identified was:

- fusion model (5)
- complement model (3)
- expansion (3)

As indicated in Figure 6.7, the 16 teachers were given a visual artefact about three common learning designs used in Computer Science, based on a systematic literature review of 66 innovative practices in CS (Rienties et al., 2023). As indicated from this review, several innovative practices in CS could be categorised as Education 4.0 light, whereby some innovative elements are introduced alongside more common practices in CS. In Project-based/hands on JCCs teachers are mainly focussed on giving their learners opportunities to work together in smaller groups and/or projects, while the third approach identified is the full EDU4 JCC approach. In this approach, teachers focus on both flexible provision (i.e., anytime, anywhere), hands-on experiences, authentic assessments, and/or involvement of students in learning design activities. As identified by the 16 teachers, six indicated that their learning activities resembled the full EDU4 model most, while four indicated that their design resembled the project-based approach.

Co-funded by the Erasmus+ Programme of the European Union

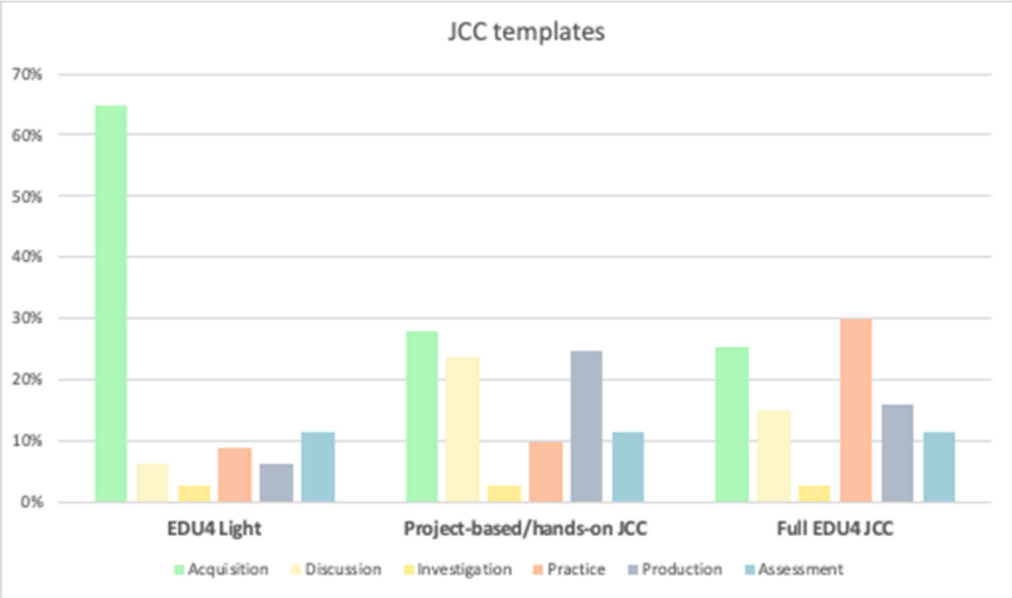


Figure 6.7 Three templates for JCC design

EDU 4.0 (quantitative) for those who identified was:

- Project based (4)
- Full EDU 4.0 (6)
- Theoretical (1)

JCC Experience

Some of the "**positive aspects**" from the JCC experience including the use of different language in international processes:

> *Collaboration with the foreign professor and presenting in a different language. It was a different experience.* (33-2)
>
> *Collaboration with the teacher from [XXX] and establishment of contact both for teaching and for future scientific cooperation. I also enjoyed gaining experience of teaching in a foreign language.* (33-1)
>
> *I think establishing contacts with colleagues from other institutions and placing students in teams with students from other countries provides an international context for learning*. (3-2)

Co-funded by the
Erasmus+ Programme
of the European Union

Collaboration between institutions and students:

> *Collaboration was that it wasn't just teaching at one university. We know our local system, and thanks to the preparation of the JCC, one can also see the experiences of foreign colleagues, how they see the world, but also that they always come with some other knowledge. (..) The students, when solving the semester projects, each had a different view. Maybe it works a little differently for them than it does for teachers, so they have to communicate with each other as a team.* (3-1)

And motivation of students:

> *The students were very interested. (…) I liked that the students were motivated and liked to work on the lessons. When the students are motivated, the teacher also learns well and is motivated.* (60-1)

While ”**negative aspects**” include the bureaucracy and a lot of time invested:

> *Honestly, I think it was the bureaucracy, having to deal with paperwork. Because we had a lot of work with this stuff, creating the content and then we were running actually parts from this project. We were involved in several other educational projects.* (12-1)

> *The least awarding aspects were an apparent effort and a lot of time needed to reduce the gaps in the time schedules of different classes on particular faculties. Besides that, managing all activities with seven teams of students on practical projects was also very demanding, mainly because of the different levels of shown engagement among students.* (3-3)

> *Additional burden on the existing workload. The JCC required additional six hours per week for almost entire semester due to preparation of lectures, conducting classes, additional office hours, etc.* (33-2)

Collaboration did not work well in all JCCs because they were busy or communication:

> *For students, although the original intention that they would work in international groups didn't quite work out in the end, because everyone was focused on their semester at their home university, and they weren't as agile at the JCC.* (19-1)

> *Communication problems and students who had difficulty cooperating in the international teams. The challenge was to motivate students to do such work; they didn't want to participate in the projects, so they received extra points within the subjects they were enrolled in at their faculties to further motivate them for the JCC*. (3-2)

Roles of teachers was not always clear:

Co-funded by the Erasmus+ Programme of the European Union

> *Some misunderstandings regarding what the role of a distance teachers was, especially in the assessment.* (19-2)

> *There were challenges in the initial contacts because the teachers did not know each other; it would have been much easier if they had physically connected and met during the face-to-face meetings.* (19-4)

Finally, **"future use"** includes using the content in "regular" classes:

> *I'm certainly thinking of integrating some of these concepts into regular classes. (…) For example this flipped classroom. (…). I tried this activity in my class. I gave them the material before. (…) I think that's very useful. I can really help people learn to read these scientific papers and textbooks, really learn to work with text.* (12-1)

In particular for master or PhD students:

> *This way of working would be suitable for students at higher levels of study (for example master's studies), where a significantly different level of prior knowledge is noticeable. As with groups of students who speak different languages*. (21-1)

And those groups with motivated students:

> *I would prefer this type of teaching for smaller elective courses with highly motivated students. It is hard to organise additional course during the semester and have the students attend it and complete all the obligations in a timely manner.* (21-2)

References

[Adorjan and Nunez-Del-Prado, 2018] Adorjan, A. and Nunez-Del-Prado, M. (2018). Fostering 21 century learning and innovation competencies through students' online collaborative activities in software engineering courses. In 2018 IEEE World Engineering Education Conference (EDUNINE), pages 1–4. IEEE.

[Adya et al., 2007] Adya, M., Nath, D., Malik, A., and Sridhar, V. (2007). Bringing global sourcing into the classroom: experiential learning via software development project. In Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, pages 20–27.

[Azadegan and Lu, 2001] Azadegan, S. and Lu, C. (2001). An international common project: implementation phase. In Proceedings of the 6th Annual Conference on innovation and Technology in Computer Science Education, pages 125–128.

[Beier et al., 2012] Beier, S., Bickel, M., Brockmann, P., and Choinzon, M. (2012). It takes a global village to teach global software engineering: A mongolian-german team-teaching project. In 2012

Co-funded by the
Erasmus+ Programme
of the European Union

International Conference on E-Learning and E-Technologies in Education (ICEEE), pages 152–157. IEEE.

[Bosnic and ˇCavrak, 2019] Bosnic, I. and ˇCavrak, I. (2019). Project work division in agile distributed student teams-who develops what? In 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), pages 162–171. IEEE.

[Bosnic et al., 2011] Bosnic, I., ˇCavrak, I., Orli´c, M., ˇZagar, M., and Crnkovi´c, I. (2011). Avoiding scylla and charybdis in distributed software development course. In Proceedings of the 2011 community building workshop on Collaborative teaching of globally distributed software development, pages 26–30.

[Bosnic et al., 2019] Bosnic, I., ˇCavrak, I., and ˇZagar, M. (2019). Assessing the impact of the distributed software development course on the careers of young software engineers. ACM Transactions on Computing Education (TOCE), 19(2):1–27.

[Bosnic et al., 2015] Bosnic, I., Ciccozzi, F., ˇCavrak, I., Di Nitto, E., Feljan, J., and Mirandola, R. (2015). Introducing scrum into a distributed software development course. In Proceedings of the 2015 European Conference on Software Architecture Workshops, pages 1–8.

[Bosnic et al., 2013] Bosni´c, I., Ciccozzit, F., ˇCavrak, I., Mirandola, R., and Orli´c, M. (2013). Multi-dimensional assessment of risks in a distributed software development course. In 2013 3rd International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD), pages 6–10. IEEE.

[Bothe et al., 2005] Bothe, K., Schutzler, K., Budimac, Z., and Zdravkova, K. (2005). Collaborative development of a joint web-based software engineering course across countries. In Proceedings Frontiers in Education 35th Annual Conference, pages S3H–7. IEEE.

[Brereton et al., 1998a] Brereton, P., Gumbley, M., and Lees, S. (1998a). Distributed student projects in software engineering. In Proceedings 11th Conference on Software Engineering Education, pages 4–15. IEEE.

[Brereton et al., 1998b] Brereton, P., Gumbley, M., and Lees, S. (1998b). Distributed student projects in software engineering. In Proceedings 11th Conference on Software Engineering Education, pages 4–15. IEEE.

[Cavrak et al., 2017] ˇCavrak, I., Bosni´c, I., Zagar, M., Ciccozzi, F., Di Nitto, E., Mirandola, R., and Crnkovi´c, I. (2017). Hall of fame nomination paper: Distributed software development course. In 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T), pages 7–8. IEEE.

[Cavrak et al., 2012] ˇCavrak, I., Orli´c, M., and Crnkovi´c, I. (2012). Collaboration patterns in distributed software development projects. In 2012 34th International Conference on Software Engineering (ICSE), pages 1235–1244. IEEE.

[Chidanandan et al., 2010] Chidanandan, A., Russell-Dag, L., Laxer, C., and Ayfer, R. (2010). In their words: student feedback on an international project collaboration. In Proceedings of the 41st ACM technical symposium on Computer science education, pages 534–538.

Co-funded by the
Erasmus+ Programme
of the European Union

[Ciccozzi and Crnkovic, 2010] Ciccozzi, F. and Crnkovi´c, I. (2010). Performing a project in a distributed software development course: lessons learned. In 2010 5th IEEE International Conference on Global Software Engineering, pages 187–191. IEEE.

[Clarke et al., 2014] Clarke, P. J., Davis, D., King, T. M., Pava, J., and Jones, E. L. (2014). Integrating testing into software engineering courses supported by a collaborative learning environment. ACM Transactions on Computing Education (TOCE), 14(3):1–33.

[Collazos et al., 2010] Collazos, C. A., Ochoa, S. F., Zapata, S., Giraldo, F. D., Lund, M. I., Aballay, L., and de Clunie, G. T. (2010). Codila: A collaborative and distributed learning activity applied to software engineering courses in latin american universities. In 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010), pages 1–9. IEEE.

[Collofello, 2000] Collofello, J. S. (2000). University/industry collaboration in developing a simulation-based software project management training course. IEEE Transactions on Education, 43(4):389–393.

[Ende et al., 2013] Ende, M., L¨ammermann, R., Brockmann, P., and Ayurzana, G. (2013). A virtual, global classroom to teach global software engineering: A mongolian-german team-teaching project. In 2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE), pages 229–233. IEEE.

[Fortaleza et al., 2012] Fortaleza, L. L., Conte, T., Marczak, S., and Prikladnicki, R. (2012). Towards a gse international teaching network: Mapping global software engineering courses. In 2012 Second International Workshop on Collaborative Teaching of Globally Distributed Software Development (CT-GDSD), pages 1–5. IEEE.

[Fu et al., 2018] Fu, Y., Reina, L. P., and Brockmann, P. (2018). Teaching global software engineering: Experience report comparing distributed, virtual collaborative courses at the bachelor's and master's degree levels. In Proceedings of the 3rd European Conference of Software Engineering Education, pages 34–38.

[Gloor et al., 2011] Gloor, P., Paasivaara, M., Lassenius, C., Schoder, D., Fischbach, K., and Miller, C. (2011). Teaching a global project course: experiences and lessons learned. In Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, pages 1–5.

[Gotel et al., 2008a] Gotel, O., Kulkarni, V., Say, M., Scharff, C., Sunetnanta, T., Touch, S., and Des, P. (2008a). Quality-driven competition: Uniting undergraduates, graduates and professionals on global software development projects. In 2008 21st IEEE-CS Conference on Software Engineering Education and Training Workshop, pages 19–21. IEEE.

[Gotel et al., 2008b] Gotel, O., Kulkarni, V., Scharff, C., and Neak, L. (2008b). Integration starts on day one in global software development projects. In 2008 IEEE International Conference on Global Software Engineering, pages 244–248. IEEE.

[Holmes et al., 2014] Holmes, R., Craig, M., Reid, K., and Stroulia, E. (2014). Lessons learned managing distributed software engineering courses. In Companion Proceedings of the 36th International Conference on Software Engineering, pages 321–324.

Co-funded by the
Erasmus+ Programme
of the European Union

[Jain and Suman, 2015] Jain, R. and Suman, U. (2015). A systematic literature review on global software development life cycle. ACM SIGSOFT Software Engineering Notes, 40(2):1–14.

[Keele et al., 2007] Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering.

[Lago et al., 2008] Lago, P., Muccini, H., and Babar, M. A. (2008). Developing a course on designing software in globally distributed teams. In 2008 IEEE International Conference on Global Software Engineering, pages 249–253. IEEE.

[Macek et al., 1999] Macek, T., Mannova, B., Kolar, J., and Williams, B. (1999). Global cooperation project in computer programming course. In The proceedings of the thirtieth SIGCSE technical symposium on Computer science education, pages 208–211.

[Marasovic and Lutz, 2015] Marasovic, K. and Lutz, M. (2015). Remote development and distance delivery of innovative courses: challenges and opportunities. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 2, pages 299–302. IEEE.

[Maresca et al., 2011] Maresca, P., Stanganelli, L., and Coccoli, M. (2011). Managing a software project leveraging students' cooperation: on the road to eclipse (otre) experience. In Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, pages 96–100.

[Marutschke et al., 2020a] Marutschke, D. M., Kryssanov, V. V., and Brockmann, P. (2020a). Distributed virtual courses to teach global software engineering: Lessons learned and best practices. In Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, pages 256–260.

[Marutschke et al., 2020b] Marutschke, D. M., Kryssanov, V. V., and Brockmann, P. (2020b). Teaching distributed requirements engineering: simulation of an offshoring project with geographically separated teams. In 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), pages 1–5. IEEE.

[Monasor et al., 2010] Monasor, M. J., Vizca´ıno, A., Piattini, M., and Caballero, I. (2010). Preparing students and engineers for global software development: a systematic review. In 2010 5th IEEE International Conference on Global Software Engineering, pages 177–186. IEEE.

[Nordio et al., 2011a] Nordio, M., Ghezzi, C., Meyer, B., Di Nitto, E., Tamburrelli, G., Tschannen, J., Aguirre, N., and Kulkarni, V. (2011a). Teaching software engineering using globally distributed projects: the dose course. In Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, pages 36–40.

[Nordio et al., 2011b] Nordio, M., Ghezzi, C., Meyer, B., Di Nitto, E., Tamburrelli, G., Tschannen, J., Aguirre, N., and Kulkarni, V. (2011b). Teaching software engineering using globally distributed projects: the dose course. In Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, pages 36–40.

[Oda et al., 2017] Oda, S., Inoue, M., and Yamazaki, A. K. (2017). Assessment of global competency for engineering students in a multicultural and multi-disciplinary project based learning course. In 2017 7th World Engineering Education Forum (WEEF), pages 439–443. IEEE.

Co-funded by the
Erasmus+ Programme
of the European Union

[Ota and Punyabukkana, 2016] Ota, E. and Punyabukkana, P. (2016). Effects of bilateral problem-based learning program for engineering students: Case of a joint course with japan and thailand. In 2016 IEEE Frontiers in Education Conference (FIE), pages 1–9. IEEE.

[Pirker et al., 2016] Pirker, J., Economou, D., and G ̈utl, C. (2016). Interdisciplinary and international game projects for creative learning. In Proceedings of the 2016 ACM conference on innovation and technology in computer science education, pages 29–34.

[Purvis et al., 2004] Purvis, M., Purvis, M., and Cranefield, S. (2004). Educational experiences from a global software engineering (gse) project. In Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30, pages 269–275.

[Richardson et al., 2006] Richardson, I., Milewski, A. E., Mullick, N., and Keil, P. (2006). Distributed development: an education perspective on the global studio project. In Proceedings of the 28th international Conference on Software Engineering, pages 679–684.

[Rusu et al., 2009] Rusu, A., Rusu, A., Docimo, R., Santiago, C., and Paglione, M. (2009). Academia-academia-industry collaborations on software engineering projects using local-remote teams. In Proceedings of the 40th ACM technical symposium on Computer science education, pages 301–305.

[Salah et al., 2018] Salah, R. M., Cecil, J., and Atrushi, D. (2018). Collaborative remote laboratories for serving sciences and engineering education in iraq: rexnet project. In 2018 International Conference on Advanced Science and Engineering (ICOASE), pages 134–139. IEEE.

[Scharff, 2011a] Scharff, C. (2011a). An evolving collaborative model of working in students' global software development projects. In Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, pages 11–15.

[Scharff, 2011b] Scharff, C. (2011b). Guiding global software development projects using scrum and agile with quality assurance. In 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), pages 274–283. IEEE.

[Scharff et al., 2010] Scharff, C., Gotel, O., and Kulkarni, V. (2010). Transitioning to distributed development in students' global software development projects: The role of agile methodologies and end-to-end tooling. In 2010 Fifth International Conference on Software Engineering Advances, pages 388–394. IEEE.

[Scharff et al., 2016] Scharff, C., Sheikh, Y. H., and Rai, I. A. (2016). Global software development projects as a framework for capacity building in developing countries. In Proceedings of the 7th Annual Symposium on Computing for Development, pages 1–4.

[Silva et al., 2015] Silva, M. F., Malheiro, B., Guedes, P., Ferreira, P., Ribeiro, C., Ferreira, F., and Duarte, A. J. (2015). Development of biomimetic robots in the eps engineering programme capstone project. In Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, pages 227–234.

[Stroulia et al., 2011] Stroulia, E., Bauer, K., Craig, M., Reid, K., and Wilson, G. (2011). Teaching distributed software engineering with ucosp: The undergraduate capstone open-source project. In

Co-funded by the
Erasmus+ Programme
of the European Union

Proceedings of the 2011 community building workshop on Collaborative teaching of globally distributed software development, pages 20–25.

[Su and Scharff, 2010] Su, S. H. and Scharff, C. (2010). Know yourself and beyond: A students' global software development project experience with agile methodology. In 2010 5th International Conference on Computer Science & Education, pages 412–417. IEEE.

[Tong and Clear, 2013] Tong, M. J. and Clear, T. (2013). A reflection on the first run of the runestone project at tongji university: Observations on cross cultural distributed teams vs face to face teams. In 2013 Learning and Teaching in Computing and Engineering, pages 148–153. IEEE.

[Van Der Duim et al., 2007] Van Der Duim, L., Andersson, J., and Sinnema, M. (2007). Good practices for educational software engineering projects. In 29th International Conference on Software Engineering (ICSE'07), pages 698–707. IEEE.

[Verkamo et al., 2005] Verkamo, A. I., Taina, J., Tuohiniemi, T., Bogoyavlenskiy, Y., and Korzun, D. (2005). Distributed cross-cultural student software project: A case study. In 18th Conference on Software Engineering Education & Training (CSEET'05), pages 207–214. IEEE.

[Zhang et al., 2020] Zhang, R., Mestre, P., Serodio, C., Prada, M., and Wenjuan, G. (2020). Web-based teamwork: Distributed software development course under covid-19. In 2020 15th International Conference on Computer Science & Education (ICCSE), pages 45–50. IEEE.

Co-funded by the
Erasmus+ Programme
of the European Union