Implementing Education 4.0 in Software Engineering Education: the JCC experience

Roberta Capuano, Henry Muccini
University of L'Aquila, Italy
FrAmeLab, DISIM Department
Via Vetoio, 67100 L'Aquila, Italy
{roberta.capuano, henry.muccini}@univaq.it

Zlatko Stapić
University of Zagreb Faculty of Organization and Informatics
Pavlinska 2, 42000 Varazdin, Croatia
zlatko.stapic@foi.unizg.hr

Abstract. *Education 4.0 emphasizes fostering innovation over merely imparting knowledge by seamlessly integrating advanced technologies into teaching and learning environments. This paradigm is particularly well-suited for software engineering education, aligning modern educational approaches with the evolving industry needs.*
This paper presents the Joint Creative Classroom methodology and reports the experience of the application of the Education 4.0 principle for the design, implementation and assessment of the Software Architecture Analysis and Design JCC course. This initiative aimed at fostering cross-university collaboration in higher education, involves two distinct universities, each contributing a unique course to create an innovative, interdisciplinary learning experience for students. Thus, this paper aims to provide insights into the challenges and successes of such collaborative endeavours.
The results indicate that over 90% of the students acknowledged the joint course's positive impact on enhancing their soft skills. Furthermore, the framework facilitated exposure to distributed software engineering practices.

Keywords. Education 4.0, interdisciplinary learning, industry collaboration, cross-university collaboration, global software engineering, learning design, JCC

1 Introduction

In the evolving landscape of *Global Software Engineering* (GSE), the imperative for flexible, cross-border, and *innovative teaching methods* has become increasingly evident (Iniesto et.al., 2021; Herbsleb and Moitra 2022; Beier et al., 2012). Teaching software engineering requires creating authentic contexts, fostering teamwork, handling projects with sufficient complexity, and providing ongoing student support (Angelov and de Beer, 2017). Moreover, as the *industry demands* continue to grow in complexity, educational institutions must equip their students with the skills and insights required to excel in this dynamic field (Moraes et al., 2023).

The *Education 4.0* framework is advocated as a solution to these emerging trends (Hussin, 2018; Bonfield et al., 2020; Salmon, 2019). Proposed by Harkins (2008), Education 4.0 describes innovation-producing education as opposed to knowledge-producing education. Definitions vary but usually focus on innovation, novelty, use of technology, and connections with employment and industry (Hussin, 2018; Salmon, 2019; Fisk, 2017; Mukul, 2023). In the Erasmus+ project *Accelerating the transition towards Edu 4.0 in HEIs – Teach4Edu4*, we proposed a definition of Education 4.0 that draws on ideas and descriptions in a range of past literature (Hussin, 2018; Salmon, 2019; Fisk, 2017; Jisc, 2019): Education 4.0 employs an approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context and therefore requires flexible and creative approaches to its implementation.

In the context of the *Teach4Edu* project, we proposed the concept of *Joint Creative Classroom (JCC)* to implement *Education 4.0* principles. JCC courses meet the demand for specialized education not offered by a single university, pooling resources from multiple institutions for a well-rounded learning experience. This matches the growing collaboration possibilities opened by Erasmus Mundus[1], European's Universities[2], and Erasmus plus national and European

[1] https://erasmus-plus.ec.europa.eu/opportunities/opportunities-for-individuals/students/erasmus-mundus-joint-masters

[2] https://education.ec.europa.eu/education-levels/higher-education/european-universities-initiative

programs (Czerska-Shaw and Krzaklewska, 2022; Fernández, 2009; Beier et al., 2012). In this context, *JCC courses* also prepare students for *GSE* practices, fostering collaboration, communication, and problem-solving skills in diverse, cross-cultural environments.

This paper presents the *JCC* methodology and reports the experience of implementing a *JCC* course titled *Software Architecture Analysis and Design*, jointly created by the *University of L'Aquila (UNIVAQ)* and the *University of Zagreb (UNIZG)*. This course builds upon the foundational courses already provided by the respective universities, enabling students from related courses not only to partially participate in each other's classes but also to work together in cross-disciplinary teams on a project, fostering the exchange of knowledge and comprehension of software architecture by *osmosis* (Odom and Kelly, 2022; Lago et al., 2012). The collaborative initiative spans the entire spectrum of software development, from architectural design to implementation and analysis, providing students with a comprehensive and immersive learning experience implementing the Education 4.0 principles.

The rest of the paper is structured as follows: Section 2 presents the related works on pedagogy methodologies, Education 4.0, and their application in the context of *GSE*. Section 3 introduces the *JCC* model and how it has been applied for the design and implementation of the *Software Architecture Analysis and Design JCC course*. Section 4 show how the implementation of the *JCC* matches the Education 4.0 principles, as well as the main challenges encountered. Finally, Section 5 concludes the paper.

2 State of the Art

In this Section, we analyse the state of the art in pedagogy methodologies in the context of computer science and *Global Software Engineering (GSE)* and Education 4.0. Finally, based on the provided analysis, we discuss the motivation and scope of this paper.

Computer Science, Pedagogy and Education 4.0. In the past years, different of systematic literature reviews (SLRs) on computer science teaching and innovative pedagogical methodologies have been published.

Aničić et al. (2016) conducted a meta-analysis encompassing 155 publications from 1980 to 2014, providing insights into the current research landscape regarding the education and career trajectories of graduates in the field of Information and Communication Technology (ICT). The findings underscore the necessity for curriculum design and implementation to be adapted to align with industry requirements. As noted by the authors, existing literature highlights the imperative for innovative strategies in curriculum design and delivery, including the establishment of competency-based programs that transcend traditional semester-based frameworks, facilitating a more flexible curriculum that minimizes classroom time, and offering courses focused on both innovation management and the process of innovating itself. Effective pedagogical methods that can enhance graduate employability include experiential learning, learning through error, teamwork, and collaborative education. Additionally, approaches such as job-oriented experimental courses, problem-based or project-based learning, and work-integrated learning are advocated to cultivate a comprehensive array of skills, competencies, and knowledge that align with the principles of Education 4.0.

Garousi et al., (2019) specifically examined the alignment of software engineering education with industrial requirements. Their SLR analysed 34 articles published between 1995 and 2018 and identified eight pertinent research questions, two of which are highly relevant to our investigation: (1) *What curriculum models (bodies of knowledge) have been utilized in the design of the studies?;* and (2) *What educational recommendations are articulated in each study?* In their study, the authors revealed four thematic categories for the educational recommendations: (1) A need for greater emphasis on soft skills (20 papers), (2) An emphasis on active Infrastructure as Code (IAC) (3 papers), (3) A reduced focus on specific topics (2 papers), and (4) Additional recommendations (7 papers). To foster the development of soft skills, the authors advocated for incorporating real-world projects, establishing industry-academia collaborations in educational design, and anticipating future trends in student preparation.

Medeiros et al. (2018) reviewed 89 papers addressing introductory programming in higher education from 2010 to 2016, investigating how prior skills impact programming learning and the challenges students and educators encounter. The authors identified a significant gap in understanding problem-solving and prior knowledge, alongside the need for improved tools and methodologies for problem formulation and solution representation. In a broader review examining Artificial Intelligence (AI) and education, encompassing 425 papers published between 2000 and 2019, Guan et al. (2020) illustrated the diverse approaches employed to integrate technology into higher education; however, only a subset of these approaches has proven sustainable over time.

Overview of practices for integrating software testing within programming education based on an analysis of 195 empirical papers was performed by Scatalon et al. (2020). The study revealed varying levels of student engagement with testing practices in programming assignments, including analysis of test results from submission tools, interaction with instructor-provided tests, utilization of support mechanisms for test design (e.g., plugins for input and expected output), and instances of students crafting their tests. Nonetheless, few studies addressed the mechanisms through which students acquired testing concepts in programming courses.

While these SLRs offer valuable insights into the intersection of computer science, programming education, and AI, none specifically address or incorporate the principles of Education 4.0. In addition, the COVID-19 pandemic, and the swift transition to online or hybrid education have significantly impacted education, necessitating a contemporary reassessment of how computer science educators embrace innovative pedagogies and Education 4.0 frameworks. This context is crucial for understanding the current state of the field and the need for innovative approaches.

Consequently, two systematic literature reviews were undertaken within the context of our *TEACH4EDU4* to investigate how computer science educators design and implement innovative methodologies and technologies consistent with Education 4.0 principles. In an initial exploratory study involving 20 studies across Europe, Rienties et al. (2021) found that most European educators employed only one or two of nine Education 4.0 elements presented by Hussin (2018).

A subsequent study (Rienties et al., 2023) examined how current pedagogical approaches in Computer Science align with the principles of Education 4.0, reviewing 66 articles incorporating an average of several key characteristics from Education 4.0. The study confirmed the existence of three distinct clusters in how computer science educators design innovative courses: 1) EDU 4.0 light, 2) project-based/hands-on learning, and 3) full EDU 4.0. In the EDU 4.0 light category, the focus was primarily on fostering independent learning, offering flexibility for learning anytime and anywhere, personalized learning, and allowing students choice in how to learn.

However, there was minimal emphasis on hands-on learning and no project-based learning. The project-based/hands-on learning cluster prioritized project-based and hands-on activities, but placed less emphasis on personalized learning, flexibility in how to learn, and learning anytime and anywhere. The third cluster, labelled full EDU 4.0, emphasized hands-on learning, independent learning, personalized learning, flexibility in learning anytime and anywhere, and student choice in how to learn.

Education in the context of Global Software Engineering. The dynamic landscape of education has witnessed a surge in initiatives aimed at fostering synergistic learning experiences, often facilitated through collaborative methodologies.

Bosnić et al. (2019) investigated the effects of Distributed Software Development Course (DSD) on the early career stages of software engineers. Results proved the beneficial effects of this initiative on professional careers. as almost 70% of the students engaged in distributed collaborations.

Report on experiences using Project-Based Learning to teach virtual, cooperative courses in global software engineering was given by Fu et al. (2018). The study provides a comparative analysis of bachelor's and master's level courses, shedding light on the challenges faced by students at various stages of their academic journey.

Marutschke et al. (2020) share experiences teaching global software engineering through distributed, virtual courses. They evaluate the effectiveness of project-based learning methods and collaborative platforms, providing best practices for similar courses. This work emphasizes the importance of effective coordination and cultural sensitivity in global software engineering education.

Bosnić and Čavrak (2019) investigate how distributed student teams in three European universities organize their project work on software products using the Scrum framework. Findings suggest that architectural choices positively impact work division, while younger students tend to create mixed-location sub-teams, leading to improved project performance.

Lago et al. (2008) outline a method for creating a collaborative course in global software development, leveraging diverse background knowledge. The authors introduce an orientation map to guide the development of a joint course, emphasizing cultural diversity.

Finally, the REXNet project in (Salah et al., 2018) involves universities in Kurdistan Region-Iraq and Oklahoma State University. Supported by IREX, it establishes distributed laboratories for collaborative experiments, modernizing education methods and promoting technological collaboration in the MENA region, particularly in Iraq.

Paper Motivation and Scope. Despite the analysis of the state of the art revealed that researchers put a lot of effort in studying the pedagogical aspects in computer science teaching and on how to apply them to promote GSE, for the best of our knowledge the adoption of the Education 4.0 framework for the design and implementation of software engineering related courses is still not investigated. Thus, the goal of this paper is to examine how the implementation of the *JCC* class aligns with the core principles of Education 4.0. Through a detailed analysis of the case study, we aim to identify which elements of Education 4.0 were addressed, how they were applied in practice, and their impact on student learning outcomes. We also seek to draw lessons from this case study to inform future educational practices within the framework of Education 4.0.

3 JCC Course Design and Implementation

This section discusses the implementation and design of the *Joint Creative Classroom* (*JCC*) course titled *Software Architecture Analysis and Design*. We show

Table 1. Education 4.0 Characteristics by Hussin

ID	Description	Addressed in our Experiment
P1	Learning can be taken place anytime anywhere.	Yes
P2	Learning will be personalized to individual student.	No
P3	Students have a choice in determining how they want to learn.	Yes
P4	Students will be exposed to more project-based learning.	Yes
P5	Students will be exposed to more hands-on learning through field experience.	Yes
P6	Students are required to apply their theoretical knowledge to numbers and use their reasoning skills to make inferences based on logic and trends from given sets of data.	No
P7	Students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient.	Yes
P8	Students' opinion will be considered in designing and updating the curriculum.	Yes
P9	Students will become more independent in their own learning.	Yes

how the course addressed nine key Education 4.0 principles created by Hussin (2018) and reported in Table 1 by fostering global collaboration (P1) through cross-border teamwork and enabling self-paced learning (P3) via flexible structures and self-assessment tools. It emphasized project-based and hands-on learning (P4 and P5), allowing students to connect theory with practical, real-world challenges while developing critical soft skills like communication and teamwork. In addition, technology-enhanced learning environments and modern assessment methods (P7 and P8) promoted innovation and placed student feedback at the core of the educational experience. Finally, we discuss the effectiveness of the course reporting the results of pre- and post-surveys administered to participant students.

3.1 Course design

In today's increasingly globalized and technology-driven world, the *JCC* joint classes methodology offers an innovative approach to teaching software architecture. Developed within the framework of *Education 4.0*, this model emphasizes cross-university collaboration to prepare students for the complexities of modern, global, software engineering. The *JCC* course, titled *Software Architecture Analysis and Design*, was conceived as an interdisciplinary educational experience designed to merge theory and practice, as well as bridge gaps between institutions and academic disciplines. By bringing together students from two universities and engaging them in a unified learning experience, this course provides a comprehensive, immersive environment in which students develop both technical expertise and essential soft skills, such as communication, teamwork, and problem-solving, all within the context of global software engineering.

The ***Software Architecture course*** offered by *University of L'Aquila (UNIVAQ)* provides a deep dive into the foundational and advanced aspects of modern software architecture[3]. The course covers a wide range of topics central to the design and management of complex software systems. Students are introduced to *Components and Connectors*, which form the building blocks of software architecture, and learn to make critical *Architecture Design Decisions* through effective *Group Decision Making (GDM)* processes. The course explores various *Architectural Styles*, including *Cloud Architectures* and *Self-Adaptive Architectures*, equipping students to design distributed and scalable systems. Additionally, the course addresses the concept of *Technical Debt*, teaching students to recognize and manage trade-offs in system design to ensure long-term sustainability. Finally, *DevOps practices* are emphasized, highlighting collaboration between development and operations teams and providing students with hands-on experience using tools such as *Kubernetes* for container orchestration and *Kafka* for messaging systems. This course offers students the theoretical and practical skills required to design, assess, and manage complex software architectures.

The ***Software Analysis and Development course*** from *University of Zagreb (UNIZG)* focuses on the entire software development lifecycle, giving students a solid foundation in both conceptual and practical aspects of software development[4]. The course begins with an exploration of *Trends and Market Analysis*, providing insights into industry dynamics and emerging technologies. Students then learn a range of *Software Development Methodologies*, with an emphasis on Agile project management frameworks, which prepare them for managing iterative development processes. A key component of the course is its focus on *Mobile Development*, where students apply *SOLID Design Principles* to build

[3] *https://univaq.coursecatalogue.cineca.it/insegnamenti/2024/35828-1/2017/9/10013*

[4] *https://nastava.foi.hr/course/93066*

maintainable, scalable software systems with mobile, smart, wearable or IoT components. The course also covers *Non-Functional Requirements*, stressing the importance of performance, security, and usability in software design. Additionally, students learn about *Testing Automation* and *Continuous Integration*, ensuring that they develop skills in validating and integrating code efficiently. By the course's conclusion, students are well-versed in both technical design and business considerations, equipping them to face modern software development challenges.

Although the individual courses provide comprehensive coverage of their respective domains, neither course offers a holistic view of developing complex software products, which inherently involve methodological, architectural and implementation demands. Thus, the objective of the ***Software Architecture Analysis and Design,*** *the JCC course* we designed, was to integrate the distinct yet complementary content of the two described courses and to create a learning experience that blends theoretical knowledge with practical application in both domains allowing students from both universities to work together and contribute to the same project from their different perspectives[5] (P4, P5). Using *learning design tools* and relying on the *complement model*, the course was structured to ensure a cohesive educational experience despite the geographical separation of the students and the differences in the institutions' curricula. This integration also allowed students to benefit from the strengths of both institutions, providing a holistic view of software architecture and development, but also aligns with Education 4.0 characteristic of enabling students learning anytime and anywhere (P1), even with students from different country.

Special attention was devoted to planning and developing the *learning design (LD)* for the *JCC course*. Learning objectives were clearly defined to align the content from both universities. This process involved a comprehensive analysis of learning outcomes but also themes and topics that will contribute to their implementation. The modes of delivery of the course content as well as the activities that were to be performed by students or by teams were carefully designed. By collaborating with industry representatives, we selected teaching and evaluation methods that best aligned with both student needs and industry expectations (P7 and P8). This curriculum design effectively integrated practical skills with theoretical concepts, creating a well-rounded educational program that prepares students for the job market while ensuring compliance with the official accreditation standards required by both universities. In this way students will become more independent in their own learning as expected in Education 4.0 (P3).

The tool used for learning design is *Balanced Design Planning Tool* (*BDP Tool*)[6] which is research-based, constructive-alignment oriented and modular solution based on *learning outcomes* (*LOs*) and learner workload as foundations of learner-centred learning approach and principles (Divjak et al., 2022). An excerpt from the tool showing the detailed plan of one teaching topic with several teaching units and different teaching and learning activities is presented in Fig. 1, while the excerpt from the learning design analysis given by the tool is presented in Fig. 2.

Course schedules were synchronized to provide a logical progression of topics, enabling students to seamlessly move from one area of study to another. Collaborative platforms were utilized for sharing course materials and resources, while communication tools like video conferencing and email facilitated interactions between students and instructors across institutions. Weekly or bi-weekly meetings with instructors ensured that students' progress was

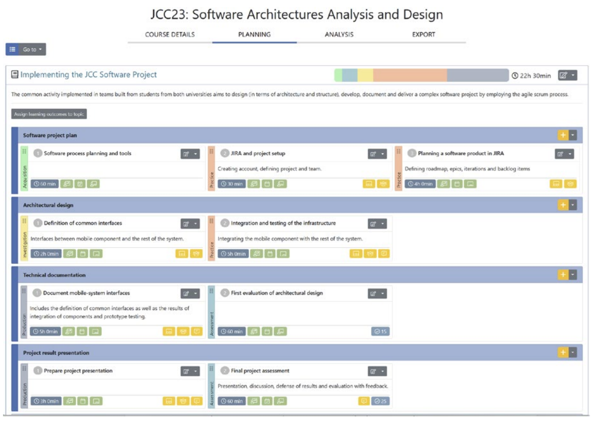

Figure 1. Excerpt from the learning design topic definition

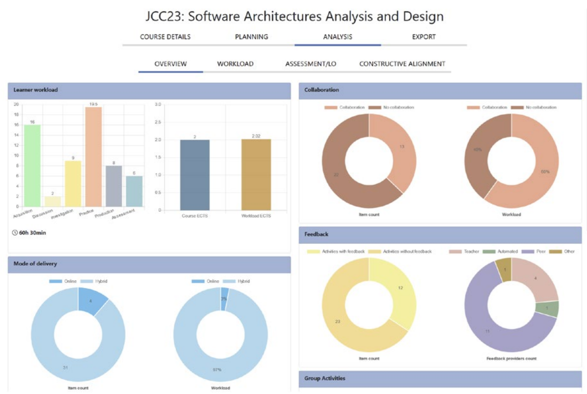

Figure 2. Excerpt from the learning design analysis

continuously monitored, allowing for real-time feedback and adjustments (P8).

To elucidate the design choices, we will highlight which collaboration models were adopted for each dimension of the course design, namely content, attendance, and assessment, shedding light on the

[5] *https://learning-design.eu/en/preview/81f9a72eb213cf3e3d1b2ea2/details*

[6] *https://learning-design.eu/*

Table 2. Adopted Models for JCC Course Design

	Content	**Attendance**	**Assessment**
Collaboration Model	Complement	Fusion	Complement Theory/Practice
Description	Introduction to Software Architecture, Components and Connectors, and Technical Debt from UNIVAQ. Trends and Market Analysis, SCRUM, and SOLID design principles, provided by UNIZG.	Students from UNIVAQ attended part of the classes provided by UNIZG. Conversely, students from UNIZG attended part of the classes from UNIVAQ.	Each student evaluated through a self-assessment approach. Mixed teams, comprising students from both universities, undertook distinct software architecture and analysis and design projects. These projects were evaluated collectively, recognizing individual contributions while celebrating the collective achievements of teams.

comprehensive framework that underpins our *JCC* course. Table 2 presents the models selected for each dimension providing a brief description of how they have been adopted in the design of the *Software Architecture Analysis and Design* course.

3.2 Course implementation

The implementation of the *JCC* course took place over one (winter) semester, but with staggered start dates between the universities. This required careful coordination to ensure that students from both institutions could engage in the joint components of the course without disruption. The course engaged three primary roles: students, instructors, and practitioners each playing integral parts in the collaborative learning experience.

Students. The course drew participation from a diverse cohort of students, representing both *UNIVAQ* and *UNIZG*. Notably, the *Software Architecture* course at *UNIVAQ* saw an approximate enrolment of 40 students, while the *Software Analysis and Design* course at *UNIZG* garnered approximately 90 students. At the outset, both groups were introduced separately to the *JCC* course, allowing them to familiarize themselves with the content and expectations. Subsequently, students were given the opportunity to express their interest in joining the *JCC* course. Six out of the 40 students from *UNIVAQ* and 18 out of the 90 students from *UNIZG* opted to embark on this cross-university educational journey. Students were organized into mixed teams, with each team comprising members from both universities.

Instructors. The instructional team consisted of two educators, one from each participating university, who played crucial roles in managing the *JCC* course. Through a mix of weekly synchronous and asynchronous meetings, they delivered lectures and exercises, monitored the course's progress, and assessed how students responded to the collaborative approach. This constant interaction allowed for real-time adjustments, ensuring the course adapted to the students' evolving needs (P8). To accommodate differences in the start dates of the courses, the instructor from *UNIVAQ* replicated key lectures for *UNIZG* students, and vice-versa. Where possible, both educators worked together to deliver joint sessions on essential topics, ensuring a comprehensive learning experience for all students, enabling smooth integration of both student group into the *JCC* course.

Practitioners. The role of the industry mentors was not only to provide the students with suitable industry-related project assignments, but also to provide real-world insights, expertise, and guidance to students, helping bridge the gap between academic knowledge and practical industry experience (P5). Their role involved mentoring students, offering industry-specific feedback, and helping to align the project with current industry standards and best practices. Most of the company's activities were conducted online, but after the completion of the projects, students from *UNIZG* had the opportunity to present their solutions to the management and other employees on-site at the company.

Attendance. In terms of attendance, the course followed a *fusion* model. Students attended their respective courses at their home universities but also participated in selected sessions from the partner institution. For example, students from *UNIVAQ* attended modules on SCRUM and SOLID design principles delivered at *UNIZG*, while students from *UNIZG* attended sessions on Components and Connectors and Technical Debt delivered at *UNIVAQ*. This cross-institutional exposure allowed students to experience diverse teaching styles and content, enriching their subject matter understanding. It is important to note that, as the *Software Architecture* and *Software Analysis and Development* courses were designed to provide 6 *ECTS* credits[7] to students, students were awarded an additional 2 *ECTS* credits for their participation in the *JCC* course.

Projects. To give the students the possibility to work on an industry related and real-life projects, the implementation of the course included the cooperation with industry partner – *Cetitec Ltd.* It is a company with a strong focus on developing an automotive communication embedded software. The company already left a deep impact on the automotive industry,

[7] ECTS - European Credit Transfer and Accumulation System

with many vehicles on the road using the company's products. As the future sets in, modern technologies and agendas require the whole industry to even enlarge the presence of various software solutions inside the vehicles, with specific requirements targeting a communication part.

In this context, three highly demanding projects were given to the students to develop viable solutions:

- Prototype of remote control of the car and its components via mobile application.
- Prototype of monitoring and configuration of car-to-car communication via mobile application.
- Prototype of autonomous driving information and configuration via mobile application.

The primary objective of the projects assigned to the students was to facilitate the development of a demonstration highlighting the seamless incorporation of contemporary technologies commonly employed in conventional automobiles into remotely operated vehicles. This integration was complemented by the introduction of a user interface and control mechanism via a mobile application. These solutions were intended not only as showcases but also as experimental platforms, fostering the exploration of novel concepts and innovations, which could subsequently undergo further refinement and integration into actual automotive systems.

It is noteworthy to underscore that the projects were designed to be autonomous, yet capable of harmonious integration, and each team had the discretion to opt for hardware simulation or physical prototyping. Moreover, the projects required students to integrate learnings from both courses, applying SCRUM methodologies to real-world challenges such as designing software solutions for the automotive industry. The industry relevance of these projects ensured that students were not only engaging with theoretical concepts but also learning to apply them in practical, tangible ways. Such setup exposed students to more hands-on learning through field experience which also contributed to the alignment of our course with Education 4.0 requirements (P5).

Teams. A total of 24 students were organized into six teams, with each project topic being concurrently assigned to two distinct teams. These teams were encouraged to engage in a collaborative exchange of knowledge and ideas, while concurrently maintaining the expectation of delivering a unique, independently authored solution.

Assessment model. The course was assessed using a *complement theory / practice model*, blending individual evaluations with collaborative project-based assessments. Students were assessed on their theoretical knowledge through quizzes and self-assessment tools specific to their home institution's course. More importantly, the assessment also included a collaborative element, where mixed teams of students from both universities worked on joint projects. The acquisition of the knowledge was assessed continuously during the semester. The process involved the usual assessment activities performed at both universities as well as the assessment of the joint elements of the course through self-assessment and project assessment (P7).

Self-assessment. Self-assessment was executed through a sequence of concise examinations intricately synchronized with the instructional content delivered by the educators. Access to these self-assessment activities was conditional upon students' prior completion of prerequisite tasks associated with knowledge acquisition, discussion, or investigation within the relevant domain. Using the tools from the learning management system, teachers prepared the pool of questions (approximately 20 questions per lesson) and designed the assessment elements (number of questions, duration, time frame for approaching the assessment, question and answer randomization options, feedback, the use of Safe exam browser etc.). During the assessment, each student was presented with unique 5-questions quiz which students had to answer in the timeframe of no more than 10 minutes.

The findings presented in Table 3 revealed a notably elevated level of intrinsic motivation among participants engaged in knowledge acquisition through the prescribed activities. All enrolled students diligently completed the self-assessment exercises. It is noteworthy that these results exhibit a considerable enhancement when compared to the performance of traditional course participants at their respective home universities.

Table 3. Summary of Assessment Results

Assessment	Pass Rate (%)	Average Score (%)
SOLID design principles	95.83	87.0
Technical debt	91.67	83.20
Software architecture	87.50	83.6
SCRUM agile framework	83.33	87.6

Assessment of projects. The assessment framework delineated three separate evaluations: the initial assessment conducted at the commencement of the semester, subsequent to the establishment of the agile project; mentoring and the intermediate evaluation (see Fig. 3), positioned at the midpoint of the semester following the conclusion of the research and architectural design phase by the teams; and the concluding assessment, which took the form of a public project solution defence held at the end of the semester. The novel approach in the assessment which did not rely on the conventional assessment methods emphasizes the assessment model as perceived by Education 4.0 (P7).

Considering the demanding nature of the projects, characterized by their multifaceted requirements encompassing intricate architectural design, the necessity for simulation or hardware prototyping, and the imperative involvement of embedded, backend,

Figure 3. Presentation of the solution mockups during the mentoring session

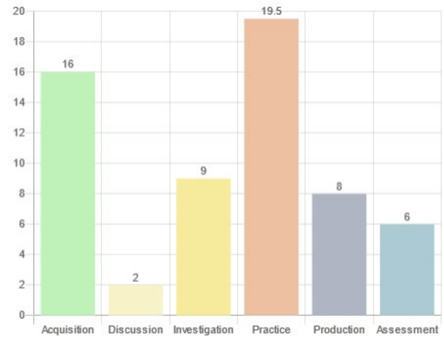

Figure 4. Distribution of student's workload in hours across different activity learning types

and mobile development, coupled with the demand for effective agile teamwork and extensive documentation, it became evident that a total of four out of the six teams succeeded in producing comprehensive solutions within the duration of the semester. Regrettably, the remaining two teams faced initial setbacks, failing to meet the requirements during the first examination period, and consequently, they were required to continue their work after the *JCC* course had concluded.

Alignment with Full Edu 4.0 learning design model. The distribution of student workload across different learning activities is shown in Fig. 4. Our course places significant emphasis on acquisition (27,67 hrs), practice (23.5 hrs), and investigation (11.17 hrs), in relation to three types of how pedagogical approaches in Computer Science align with the principles of Education 4.0 as detailed in (Rienties et al., 2023).

This model emphasizes experiential learning, self-directed study, personalized education, flexibility to learn anytime and anywhere, and providing students with the autonomy to choose their learning methods. Further information on how the *JCC* course matches with the Education 4.0 principles are provided in Section 4.

3.3 The effectiveness of the course

The effectiveness of the *JCC* course was assessed through pre- and post-surveys administered to participating students. The surveys aimed to gauge their perceptions and experiences in various aspects of the course focusing on *real-word skills development*, *soft skills development*, and the evaluation of the *assessment process*. The following subsections present a comparative analysis of the pre- and post-survey responses.

Real-world Skills Development. The results of the pre and post surveys on the course demonstrate an interesting shift in students' perspectives regarding gaining hands-on, authentic experiences, and real-world skills as depicted in Fig. 5 (P5). Before the course, a considerable number of students expressed a neutral opinion, indicating that they were unsure about their ability to gain such experiences. However, post-course surveys show that a substantial number of students have shifted their perspective positively. A considerable percentage of students have moved from the "Neutral" category to the "Agree" and "Strongly Agree" categories, reflecting a tangible increase in their confidence that they have indeed gained hands-on, authentic experiences, and real-world skills through the course.

The data strongly supports the efficacy of the course in providing students with the hands-on experiences and practical skills sought after in the field of software engineering. The shift from a more neutral stance in the pre-survey to a highly positive one in the post-survey underscores the course's ability to meet the students' expectations and aspirations for authentic learning experiences.

Soft Skill Development. The survey results demonstrate a positive shift in students' perceptions regarding the development of their soft skills, specifically in areas such as team-working and creativity as depicted in Fig. 6 (P4). Prior to the course, most students (13) expressed agreement with the statement, indicating an initial interest in improving these skills. Following the course, this sentiment was further reinforced, with an even larger number (22) of students indicating that they were able to strengthen their soft skills. Notably, a sizeable portion of students (11) were initially neutral on the matter, suggesting a potential for growth in this area.

The absence of responses in the "Strongly Disagree" and "Strongly Agree" categories both before and after the course indicates a consistent middle ground. This suggests that, while some students may not have felt strongly about their soft skill development prior to the course, the majority did see positive progress in this regard by the course's conclusion.

Innovative Assessment. The data from the pre- and post-surveys regarding the assessment methods provides interesting insights into the students'

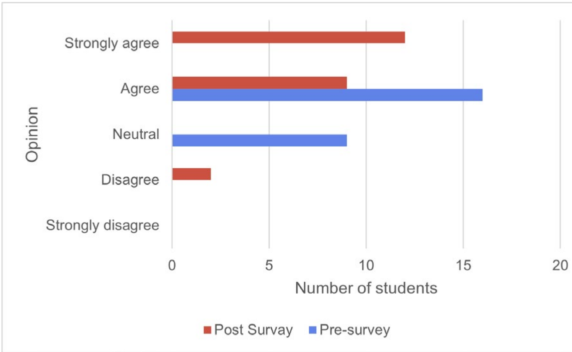

Figure 5. Hands-on Experiences and Real-world Skills

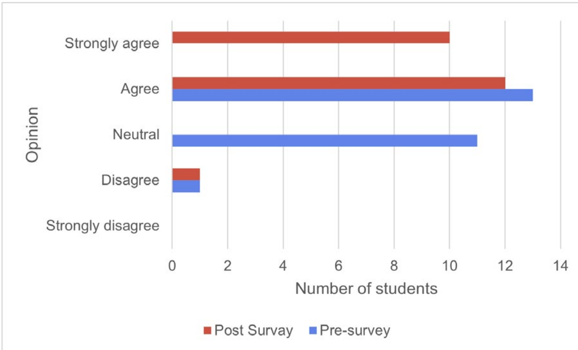

Figure 6. Development of soft skills

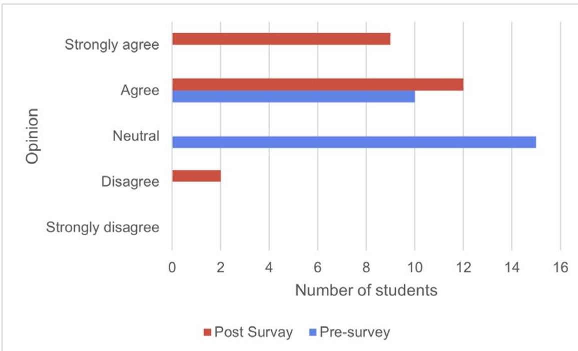

Figure 7. Innovative assessment

perspectives on innovative evaluation techniques as reported in Fig. 7 (P7). Before the course, most students were in a neutral position, with 15 expressing neither agreement nor disagreement with the statement. This suggests a certain level of uncertainty or a lack of prior exposure to innovative assessment methods.

After the course, we observe a notable shift in opinions. A considerable number of students (12) now agree that they have participated in new ways of assessment, indicating a positive change in their perception. Additionally, nine students strongly agree, demonstrating a ringing endorsement of the innovative assessment methods introduced during the course.

This data suggests that the course successfully introduced students to innovative assessment techniques, leading to a favourable shift in their perspectives.

Students Level of Satisfaction. The results of the post-survey indicate an elevated level of satisfaction and positive feedback from the students. A significant majority, 15 out of 21, expressed satisfaction with the JCC course, emphasizing its effectiveness in meeting their expectations. Even more encouraging, 16 out of 21 students expressed a willingness to recommend the course to their peers, underscoring the value they saw in the collaborative learning experience. Additionally, the unanimous agreement among all students regarding the support provided by the teacher, who acted as a facilitator, highlights the effectiveness of this teaching approach in fostering a conducive learning environment. These results affirm the success of the *JCC* course in delivering a meaningful and impactful educational experience.

Other Considerations. The continuous assessment model included mentoring by company experts and university instructors as part of the project work, as well as theoretical knowledge assessments through online tests created on the e-learning platform.

Considering *UNIZG* students, the average final grade of JCC students is 64,4% and pass rate is 100% while the average final grade of non-JCC students is 64,2% and pass rate is only 82,42%, while 17,58% failed the course and had to enrol the course next academic year.

Concerning *UNIVAQ* students, the average final grade of JCC students is 88,3%. Five of six students enrolled in the JCC course pass the exam in the first session (83%). On the other hand, non JCC students scored, in average, 93%, but only the 79% of the students completed the exam by the first session. It is important to notice that *UNIVAQ* JCC students scored worst due to synchronization problem with students from *UNIZG*.

4 Discussion

The *JCC* course, designed collaboratively between two European universities, provides an interesting case study of the practical application of Education 4.0 principles in higher education. The course emphasized interdisciplinary learning, cross-border collaboration, and the integration of advanced technologies to prepare students for the dynamic and global landscape of software engineering. In this section, we explicit how the design and implementation of the course matches the Education 4.0 core principles, as well as the encountered main challenges.

4.1 The *JCC* Course and Education 4.0 Principles

Principle P1: Cross-Institutional Collaboration. The course leveraged the fusion model, where students from two different universities participated in joint classes and projects. This experience exposed students to diverse viewpoints, promoting the exchange of knowledge across disciplines and borders. This aligns with the idea that learning can occur anywhere (P1), as it breaks down physical barriers and prepares students for the globalized, interconnected nature of modern work environments. In addition, the cross-university collaboration not only enhanced students' domain-specific and industry-ready knowledge but also equipped them with essential soft skills such as teamwork, communication, and problem-solving within a global software engineering context. Indeed, this collaborative approach also mirrored the needs of the global software engineering (GSE) industry, where professionals often work in geographically distributed teams. It enabled students to simulate real-world collaborative environments, further aligning their skills with industry expectations.

Central to the success of the joint course was the implementation of technology-enhanced learning. Various digital platforms were employed to facilitate seamless communication, coordination, and learning. The use of video conferencing, shared platforms for course materials, and real-time project management tools enabled efficient collaboration despite geographical distances. This aligns with the Education 4.0 principle of promoting learning anywhere (P1), thus breaking down traditional barriers to education and fostering flexibility.

Principles P4 and P5: Project-Based and Hands-On Learning. A standout feature of the course was its focus on project-based learning (P4). Students engaged in real-world projects in collaboration with industry partners, which simulated the complexities of modern software development. This approach provided them with hands-on experience, allowing them to apply theoretical knowledge in practical settings (P5). Industry-driven challenges, provided by an external partner, further enhanced the learning experience, giving students insight into the expectations and realities of the industry. Project-based learning aligns with the Education 4.0 framework, which emphasizes the need for learners to develop skills through experiential learning. By working on real-world projects, students were able to develop critical thinking, creativity, and innovation, all of which are essential skills for the future workforce.

One of the most significant educational strategies employed in the course was the *complementary theory/practice* model. The course seamlessly integrated theoretical lectures with practical applications, particularly using SCRUM methodology in software development projects. This approach allowed students to see the direct application of theoretical concepts to real-world scenarios, which is a fundamental tenet of Education 4.0 (P5). The fusion of theory and practice not only deepened the students' understanding of complex software architecture concepts but also prepared them for the agile work environments that are prevalent in the industry. This experiential learning approach is essential for ensuring that students can transition smoothly from academic settings to professional roles.

In addition, over 90% of the students involved in the course reported a significant improvement in their soft skills. This is a critical achievement in the context of Education 4.0, which places a strong emphasis on the development of interpersonal skills alongside technical proficiency through hands-on learning (P5). The collaborative nature of the course fostered communication, teamwork, and leadership skills, as students worked in mixed teams, combining diverse skill sets from the two institutions. Finally, problem-solving was a core skill developed through this course. By addressing real-world challenges and working in interdisciplinary teams, students were able to enhance their ability to analyse complex issues, propose solutions, and adapt to changing project requirements – skills that are indispensable in the ever-evolving technology industry.

Principles P3, P7, and P8: Self-Paced Learning, Assessment and Curriculum Design. The flexible structure of the course gave students autonomy over how they wanted to engage with the material, giving them choices on pacing and learning pathways. This is a direct reflection of the principle that students should have control over how they learn, promoting more self-directed learning experiences (P3).

The use of self-assessment tools marks a shift from traditional assessment methods, providing a more dynamic way for students to evaluate their progress. This aligns with modern approaches where conventional assessment platforms become less relevant, and student feedback becomes key in shaping future learning experiences (P7).

Finally, the usage of tools for course design based on learning design and constructive alignment methods, the self-assessment tools and the pre- and post-surveys allowed the instructor to design and update the curriculum based on the student's knowledge and learning outcomes (P8).

4.2 Challenges

Despite the overall success of the course, several challenges and lessons were identified that highlight areas for improvement in future implementations of the joint course model.

Missed Deadlines and Project Overload. A significant challenge observed was that not all teams were able to meet the project deadlines. Specifically, two out of the six teams failed to deliver their project solutions by the end of the semester, necessitating continued work beyond the duration of the course. The complex and demanding nature of the assigned projects, which included intricate architectural designs, embedded systems, and extensive documentation requirements, may have contributed to this outcome. The scope and workload should be carefully managed to ensure that all students can complete the tasks within the given timeframe.

Synchronization of Course Schedules. Another challenge faced was the difficulty in synchronizing the course schedules between the two participating universities. The universities had different start dates for the semester, which caused disruptions in the delivery of some joint lectures and delayed project kick-offs. This issue highlights the importance of international alignment of academic calendars or the incorporation of flexibility into the course structure to accommodate such discrepancies.

Effort to Bridge Intercultural Differences. Given the international nature of the course, students from diverse cultural backgrounds participated. Some intercultural communication challenges emerged, necessitating extra effort from both students and instructors to overcome differences in work habits, communication styles, and academic expectations. While this provided valuable learning experiences, it also required additional support mechanisms, such as increased mentoring and frequent meetings, to ensure smooth collaboration.

Coordination and Assessment Complexity. The joint course model involved two universities and an industry partner, which required clear coordination of roles in the assessment process. However, students from different courses possessed varying levels of technical expertise, leading to difficulties in establishing uniform assessment criteria. Balancing the evaluation of theoretical and practical skills across diverse student backgrounds remains a challenge that needs to be addressed more effectively in future courses.

5 Conclusion

The implementation of the *JCC* joint course represents a successful application of Education 4.0 principles in software engineering education. By fostering interdisciplinary, cross-university collaboration, and integrating technology-enhanced and project-based learning, the course provided students with a holistic and practical experience that mirrors the demands of the global software engineering industry.

An important aspect of the course's success was the careful use of *learning design* methodology and tools. By using structured learning design methodology, the course effectively integrated diverse content and learning objectives from two universities into a cohesive educational experience. This planning ensured that students engaged in a balanced mix of theory and practice, with clear learning outcomes aligned with industry needs. The thoughtful design of activities, assignments, and assessments played a crucial role in facilitating personalized learning experiences and optimizing student engagement across institutions.

Key lessons learned from the course highlight both the strengths and challenges of this innovative educational approach. The development of students' technical and soft skills, especially in real-world problem-solving, communication, and teamwork, was a notable success. Additionally, the use of advanced technologies facilitated flexible, cross-border learning, aligning with the core tenets of Education 4.0.

However, challenges such as schedule synchronization between universities, managing intercultural differences, and ensuring timely project completion underscore the need for careful planning and coordination in future implementations. Addressing these challenges will further improve the effectiveness of such courses and strengthen the alignment with Education 4.0 objectives.

In conclusion, the *JCC* joint course provides valuable insights into how software engineering education can evolve to meet the needs of the modern workforce. By embracing the principles of Education 4.0, it creates a model for future educational initiatives that can better prepare students for the complexities of the global and digital economy.

References

Angelov, S., & de Beer, P. (2017). Designing and applying an approach to software architecting in agile projects in education. *Journal of Systems and Software, 127,* 78–90.

Aničić, K. P., Divjak, B., & Arbanas, K. (2016). Preparing ICT graduates for real-world challenges: Results of a meta-analysis. *IEEE Transactions on Education, 60*(3), 191–197.

Beier, S., Bickel, M., Brockmann, P., & Choinzon, M. (2012). It takes a global village to teach global software engineering: A Mongolian-German team-teaching project. *In 2012 International Conference on E-Learning and E-Technologies in Education (ICEEE)* (pp. 152–157). IEEE.

Bonfield, C. A., Salter, M., Longmuir, A., Benson, M., & Adachi, C. (2020). Transformation or evolution? Education 4.0, teaching and learning in the digital age. *Higher Education Pedagogies, 5*(1), 223–246.

Bosnić, I., & Čavrak, 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)* (pp. 162–171).

Bosnić, I., Čavrak, I., & Žagar, 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.

Czerska-Shaw, K., & Krzaklewska, E. (2022). Uneasy belonging in the mobility capsule: Erasmus Mundus students in the European Higher Education Area. *Mobilities, 17*(3), 432–445.

Divjak, B., Grabar, D., Svetec, B., & Vondra, P. (2022). Balanced learning design planning: Concept and tool. *Journal of Information and Organizational Sciences, 46*(2), 361–375.

Fernández, O. B. (2009). A computer sciences teaching experience in the Erasmus Mundus Program. In *INTED2009 Proceedings* (pp. 2566–2570). IATED.

Fisk, P. (2017). Education 4.0 … the future of learning will be dramatically different, in school and throughout life. *Retrieved from https://www.peterfisk.com/2017/01/future-education-young-everyone-taught-together/*

Fu, Y., Reina, L. P., & 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* (pp. 34–38). ACM.

Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M. (2019). Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software, 156,* 65–83.

Guan, C., Mou, J., & Jiang, Z. (2020). Artificial intelligence innovation in education: A twenty-year data-driven historical analysis. *International Journal of Innovation Studies, 4*(4), 134–147.

Harkins, A. M. (2008). Leapfrog principles and practices: Core components of education 3.0 and 4.0. *Futures Research Quarterly, 24*(1), 19–31.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software, 18*(2), 16–20.

Hussin, A. A. (2018). Education 4.0 made simple: Ideas for teaching. *International Journal of Education and Literacy Studies, 6*(3), 92–98.

Iniesto, F., Sargent, J., Rienties, B., … Muccini, H. (2021). When industry meets Education 4.0: What do computer science companies need from higher education? In *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)* (pp. 367–372). ACM.

Jisc. (2019). Education 4.0 – transforming the future of education (through advanced technology). [Video]. YouTube. https://www.youtube.com/watch?v=aVWHp8FsV1w

Lago, P., Muccini, H., & Babar, M. A. (2008). Developing a course on designing software in globally distributed teams. In *2008 IEEE International Conference on Global Software Engineering* (pp. 249–253). IEEE.

Lago, P., Muccini, H., & Babar, M. A. (2012). An empirical study of learning by osmosis in global software engineering. *Journal of Software: Evolution and Process, 24*(6), 693–706.

Marutschke, D. M., Kryssanov, V. V., & Brockmann, P. (2020, January). 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* (pp. 256–260). ACM.

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77–90.

Moraes, E. B., Kipper, L. M., Hackenhaar Kellermann, A. C., Austria, L., Leivas, P., Moraes, J. A. R., & Witczak, M. (2023). Integration of Industry 4.0 technologies with Education 4.0: Advantages for improvements in learning. *Interactive Technology and Smart Education, 20*(2), 271–287.

Mukul, E., & Büyüközkan, G. (2023). Digital transformation in education: A systematic review of education 4.0. *Technological Forecasting and Social Change, 194,* 122664.

Odom, A. L., & Kelly, P. V. (2001). Integrating concept mapping and the learning cycle to teach diffusion and osmosis concepts to high school biology students. *Science Education, 85*(6), 615–635.

Rienties, B., Balaban, I., Ferguson, R., Herodotou, C., Iniesto, F., Muccini, H., … Virkus, S. (2021). Education 4.0 and computer science: A European perspective. In *Proceedings of the Central European Conference on Information and Intelligent Systems* CECIIS (pp. 139–146)

Rienties, B., Ferguson, R., Gonda, D., Hajdin, G., Herodotou, C., Iniesto, F., … Isidori, M. V. (2023). Education 4.0 in higher education and computer science: A systematic review. *Computer Applications in Engineering Education, 31*(5), 1339–1357.

Salmon, G. (2019). May the Fourth be with you: Creating Education 4.0. *Journal of Learning for Development, 6*(2).

Scatalon, L. P., Garcia, R. E., & Barbosa, E. F. (2020). Teaching practices of software testing in programming education. In *2020 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Salah, R. M., Cecil, J., & Atrushi, D. (2018, October). Collaborative remote laboratories for serving sciences and engineering education in Iraq: REXNET project. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 134–139). IEEE.