Bridging Academia and Industry: A Collaborative Approach to Software Development Education

Lea Masnec, Zlatko Stapić
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin
{lmasnec, zlatko.stapic}@foi.unizg.hr

Maja Benkus
SPAN d.d.
Koturaška cesta 47, 10 000 Zagreb
mbenkus@span.eu

Abstract*. This paper showcases an innovative approach to coursework by integrating practical knowledge with specific challenges posed by companies. The Joint Creative Classroom (JCC) program, involving both universities and companies, has proven effective in teaching software engineering, focusing on software analysis and development. A student survey indicates significant improvement in knowledge across topics related to software development and project management. The paper also highlights curriculum planning and the use of contemporary tools. The primary aim is to demonstrate student-company networking through teaching within the JCC program.*

Keywords: education with industry collaboration, interdisciplinary learning, joint creative classroom program, networking, software development education

1 Introduction

Today's market demands are on the rise, leading to an increased need for a broader spectrum of knowledge necessary for operating and educating in the information technology sector. To stay up to date with market trends, it is important to invest efforts in educating potential employees, namely students. In addition to college courses providing essential knowledge in industry-related areas, collaboration with companies offers a significant advantage. Through such partnerships, students and faculty can gain first hand insights into market needs and conditions, potentially specializing in specific industry branches and acquiring competitive expertise.

The demand for students with practical experience is continuously increasing. Similarly, software development and the management of digital platforms are currently pivotal concepts for innovation and success in business (Scaturro, 2023). The rationale behind this research lies in recognizing the transformative nature of software development and the imperative to align educational programs with the evolving needs of the industry (Umar & Rana, 2024). We aim to dissect the intricacies of the JCC program, examining its role in skill development and practical knowledge acquisition. Additionally, this paper seeks to present the specific benefits and challenges associated with incorporating one of the leading cloud platforms, namely Microsoft Azure (Peddireddy, 2024), into the educational landscape. This research explores the concept of JCC as an integrative approach to training students in the field of software development. We focus on the skill development, ranging from the fundamental principles of digital platforms to the implementation of applications on the Microsoft Azure cloud to combine the knowledge learned in the program with the knowledge from the business profession.

The research question to be answered in this paper is: *What are the aspects of utility in integrating the JCC program into existing courses for a better comprehension of practical knowledge?*

The structure of the paper consists of five chapters contributing to the answer to stated research question. Firstly, we present the related work, and the current state-of-the-research and practice related to the agility and collaboration between industry and academia in education. Upon aligning the need for our research, we present our research method and performed activities in the third chapter. We have used a qualitative research methods involving both company and students. The backbone of the research is related to the questionnaire survey performed within the group of students who participated in the JCC program. In the fourth chapter the course and learning design are presented along with the obtained results. This chapter includes the rationale on defined learning outcomes as well as insights in final software solutions created by teams of students. However, the most important aspect of the chapter gives the results on received students feedback. In the final chapter of this paper the conclusions on this innovative approach are given.

2 Related works

Collaboration between the academic community and industry stands as a cornerstone for shaping innovative educational programs geared towards preparing literature review that we performed shows that in the last years several authors have been investigating and contributing to this field of agile and collaborative teaching and learning.

Devedžić (2010) offers insights into agile software development education via a case study, aligning with the agile approach of JCC and emphasizing adaptability and continuous improvement in students' knowledge development. This focus on agility forms the crux of this scientific paper.

Matus et al. (2023) delve into factors influencing student satisfaction within JCC, assessing changes in individual student skills post-course completion, and the impact of student engagement and peer collaboration on satisfaction levels.

Gannod et al. (2015) and Liebenberg et al. (2015) explore the agile approach in education, with the latter underscoring the importance of tailoring educational programs to industry requirements, a core objective of JCC. Martínez Fernández et al. (2023) and Bocconi et al. (2012) contribute to the discourse by advocating project-based learning for teaching software analytics and data science best practices, enriching our understanding of collaborative education practices and fostering creativity in classrooms.

Easterling (2022) delves into collaboration dynamics among university faculties, pertinent to the theme of connecting with industry partners. Guo & Yang (2012), Layman (2006), and Nwodo (2023) further this understanding by providing insights into integrating education and industry in software development contexts, with a focus on project-based learning, changes in students' perceptions, and practical tools like Azure DevOps for collaborative software development.

Abegglen et al. (2021) stress the significance of partnerships in the educational ecosystem, echoing the ethos of JCC, while Malik & Zhu (2023) explore the impact of project-based learning on student success, reinforcing the practical dimension of JCC's objectives. Oda et al. (2017) assess engineering students' global competence through project-based learning, embracing the multidisciplinary approach of JCC.

Scarani et al. (2020) and Zantedeschi et al. (2020) offer insights into technological tools supporting collaborative learning and decentralized collaborative learning, respectively, enriching our understanding of collaborative software development aligned with JCC principles.

3 Method

This research relies on the three-phase methodological approach to explore the effectiveness of integrating the JCC program into existing courses in software development. This study has several limitations that should be considered when interpreting the results. The research was conducted at a single institution, which may limit the generalizability of the findings to a broader population. The sample consisted of a relatively small group of only 18 students, which could affect the statistical power and reliability of the quantitative analyses. In terms of data protection, special attention was given to the privacy and anonymity of the participants. The names of the students were not collected or mentioned in the study, and all survey responses were anonymous. These steps were taken in accordance with ethical guidelines for research to ensure the protection of personal data and the confidentiality of participant information.

3.1 Learning design

In the initial stage of the research, special attention was devoted to planning and developing the learning design for the JCC program. This process involved a comprehensive analysis of industry needs in software development to identify key themes and program objectives. By collaborating with industry representatives, we selected teaching and evaluation methods that best aligned with both student needs and industry expectations. 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 in university education.

3.2 Questionnaire survey

Quantitative evaluation of the program's effectiveness was conducted through a survey among students who took the program. The survey questionnaire was designed to quantitatively assess students' progress in understanding concepts of software development before and after participating in the JCC program. The questions in the survey were focused on various aspects of software development, and the responses used a scaled metric for quantitative analysis.

3.3 Qualitative evaluation by experts

In addition to quantitative evaluation, the qualitative aspect of the program's effectiveness was obtained through evaluation by experts participating in the JCC program. Company representatives assessed the practical skills and competencies students acquired through the program, providing deeper insights into the actual benefits of the program for students. Qualitative evaluation facilitated the assessment of students' readiness for the job market and identification of areas where the program could be further improved to better meet industry needs.

4 Program implementation and obtained results

Throughout the program, students successfully mastered key concepts and principles of digital platforms. They developed a profound understanding of the role of digital platforms in modern business. Agile software development became a daily practice, and students acquired the ability to craft high-quality user stories. Through active participation in projects on the Azure DevOps platform, they applied their acquired knowledge in real-world scenarios.

However, the overall process of planning and designing this program started a months before, incorporating the several learning design activities and phases. Thus, in this chapter we present both program planning and program execution aspects, including the results obtained with questionnaires given to the students before and after the program.

4.1 Learning design and program implementation

A previously mentioned comprehensive method of learning design was performed to ensure the best alignment of all teaching and learning activities and students' coursework with desired learning outcomes and overall workload of the students.

The JCC aspect was incorporated in a graduate level program *Analysis and program development* and was called *Agile Development of Digital Platforms*. Program syllabus covered various topics relevant to modern software engineering. Those topics are all covered in a way that only latest trends in software tools and methodologies were followed and considered. However, to build up on this university version of the program we wanted to replace the activities related to the students' project with the innovative version of the curriculum supported by industry experts from partnering company.

The tool used for learning design is Balanced Design Planning (BDP) Tool (Divjak et al., 2023) which is research-based, innovative, constructive-alignment oriented and modular solution based on learning outcomes (LOs) and learner workload as (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 the Fig. 1.

The tool enabled us to define and connect all the activities with the learning outcomes and to explore the overall coverage of the learning outcomes with both teacher present and non-teacher present activities.
All topics covered are listed in section 4.2. in Table 1. Each topic was covered by lectures held by thirteen company experts. In classic academic courses all lectures are usually held by one or two course professors, so this was a great opportunity for students to have each topic covered by industry experts specialized for each topic. Also, they had an opportunity to experience different teaching styles.

The JCC program attended 18 students. Since one of the program goals was teamwork skills development, students were divided into four teams each containing four or five team members. Each team could choose one of three different project assignments. Team's progress was assessed in three project submissions. Evaluation criteria were clearly defined and available to students on program's *e-learning* page. Evaluation criteria followed topics covered in lectures which were held by the time assessment is done. Beside scores, in each project assessment students were provided with feedback about what was done well and what needs to be improved for the next phase. Project assessments were done by industry experts, so students had an opportunity to gain valuable feedback from industry standpoint. Since this program was project-oriented, scores gained through project submissions had a major weight in overall student score. Students were also assessed in three knowledge quizzes held during

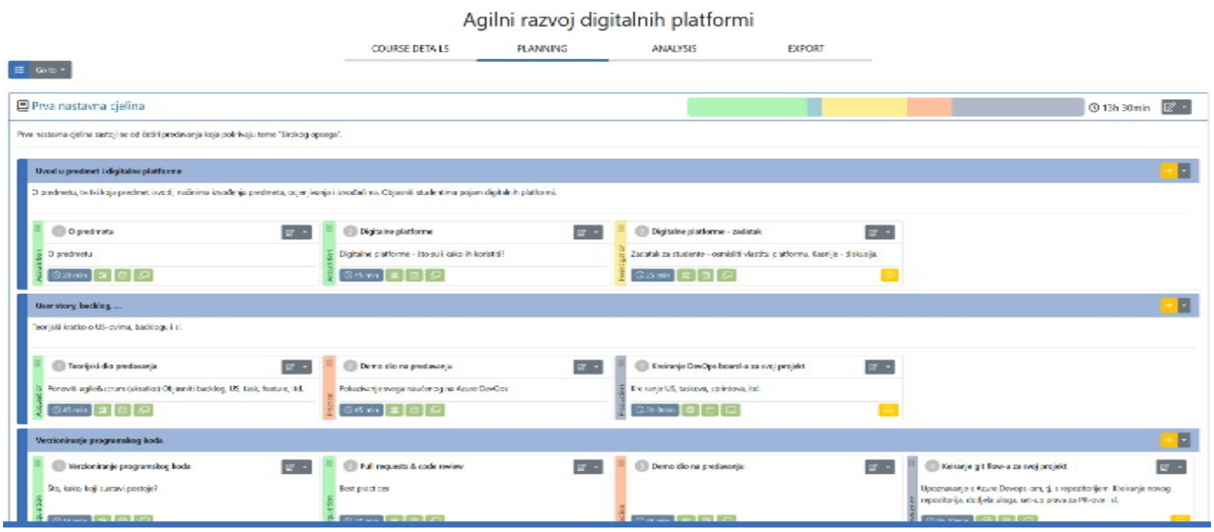

Figure 1. Excerpt from the learning design tool

program duration. In those quizzes theoretical knowledge is tested from the topics covered in lectures.

The results from the JCC aspect of the overall program were taken into account and joined with the results in other evaluation elements which were performed in the main program (such as assessment of theoretical knowledge) in order to define the final grade of the students.

4.2 Learning outcomes

Table 1 summarizes the achieved training outcomes of the JCC program, highlighting their innovation in applying tools like Azure DevOps and their practical utility in the real software development. The table shows the total of seven topics that were part of the JCC program and contribute to the practival knowledge of the students necessary to implement their project assignments. All other topics were part of the main university course and are not listed in this table.

Table 1. Achieved training outcomes

Topic	Learning Outcome	Innovation on the Program	Practical Utility
Digital platforms and agile development	Understanding key concepts of digital platforms and applying agile principles in practice.	Implementation of Azure DevOps as part of agile development.	Improved team collaboration, faster development cycles.
User story & backlog	Ability to define, manage, and prioritize user stories and product backlog.	Use of Azure DevOps for backlog management and user story creation.	Enhanced clarity in project requirements and prioritization.
Code versioning and best practices	Ability to manage code versions and maintain code quality through best practices.	Active collaboration within the team through code evaluation.	Maintenance of clean and quality code.
Architecture (monolithic vs. microservices)	Understanding the differences between monolithic and microservice architectures and their applications.	Application of best practices in designing both types of architectures.	Development of scalable and sustainable applications.
Data layer and entity framework	Designing and implementing data layers and using the Entity Framework for database management.	Use of Appropriate data management techniques and tools.	Efficient data handling and management in applications.
Business logic and API layer	Developing robust business logic and effectively creating API layers.	Implementation of strategic API endpoints and business rules.	Creation of essential application functionalities.
Authentication, validation, and logging	Mastering security practices including authentication, validation, and logging.	Application of security measures and logging in real projects.	Development of secure, reliable applications with comprehensive logging.
Debugging and exception handling	Proficiency in debugging techniques and handling exceptions effectively.	Use of advanced debugging tools and methodologies.	Improved application stability and reliability.
Clean code and SOLID (principles of object oriented design)	Applying clean code principles and SOLID design patterns in software development.	Application of clean coding practices and SOLID principles in projects.	Development of maintainable and scalable software.
Microsoft Azure	Understanding and using Microsoft Azure for application deployment and development.	Utilization of Azure Cloud services for real-world projects.	Gained experience with cloud-based application development.
Deploy and CI/CD on Azure	Configuring deployment pipelines and	Implementation of CI/CD using	Faster and more reliable software

	automating CI/CD processes on Azure.	Azure DevOps for streamlined workflows.	deployment processes.

4.3 Finalized student solutions

After working throughout the entire semester on the comprehensive solution, encompassing frontend, backend, Azure management, and documentation guided by agile development, all four student teams were required to showcase their final solutions to the company through a presentation and project defense. While students had already defended their projects in front of the faculty professors, the additional review within the company provided them with a unique opportunity to present their final product before industry experts. Notably, the company representatives who attended the presentations had also served as mentors to the teams throughout the semester. Consequently, students had the invaluable chance to receive feedback directly from experts in the field on which they had been working.
Fig. 2 illustrates the students' presentation within the company.

Figure 2. Presentation of student work in *the SPAN d.d.*

This culmination event served as a platform for the teams to demonstrate their achievements and innovations to professionals within the industry, creating a bridge between academic learning and real-world application. It further reinforced the collaborative nature of the educational program, emphasizing the practical relevance and impact of the students' efforts on the industry landscape.

Following the questionnaire-based assessment, the utilization of qualitative evaluation by companies becomes evident.

The qualitative evaluation of the JCC program was conducted in three phases, where participants projects were systematically assessed based on predefined criteria. Each evaluation phase involved a thorough review of the projects, reassessment of identified issues, and the provision of detailed feedback to the teams, which facilitated continuous improvement and a significant increase in the quality of the projects. The evaluation criteria included project management using Azure DevOps, technical implementation according to best practices (such as layered architecture, clean code, and SOLID principles), functionality coverage and accuracy, successful deployment on Azure Cloud, and the quality of the final project presentation.

The evaluation process involved experts from various fields: Software Developers and Business Analysts assessed business analysis, git flow, and architecture, while two Software Developers evaluated the programming aspect of the projects. The final project defense was assessed by considering the feedback from all evaluators, including Team Leads, Software Architects, and QA specialists. During the evaluation process, each evaluator independently reviewed and tested the projects and assigned scores based on the defined criteria. After individual assessments, the evaluators convened to discuss their scores and reach a consensus on the final ratings, ensuring fairness and objectivity.

The evaluation results were documented through evaluators notes, feedback provided to the teams, and Teams received personalized comments with specific recommendations for improvement, such as code optimization and better project management. Throughout the iterative evaluation process, there was a notable improvement in the quality of the solutions, underscoring the importance of continuous support and feedback in the development of participants skills.

4.4 Analysis of student feedback

For the purpose of this scientific study, a survey questionnaire was conducted to evaluate the program's impact on students' knowledge across various topics related to software development and project management. Out of the 18 students enrolled in the program, only 11 chose to complete the survey. This response rate reflects the voluntary nature of survey participation, where not all students opted to provide their feedback. It is important to note that the decision to participate in the survey was left entirely up to the students, which may have influenced the number of responses received. The arithmetic mean of their scores for each topic before and after the program was presented in Fig. 3. Students assessed their knowledge on 11 topics using a scale from 1 to 5, where 1 indicates "no knowledge" and 5 indicates "complete knowledge". The topics included digital platforms, user stories, code versioning and best practices, software architecture, data layers, business logic, security, debugging, clean code, Microsoft Azure, and CI/CD. The top line with the topic title represents the knowledge before, while the bottom line represents the knowledge after.

The survey results show a significant increase in average scores across all topics after completing the program (Carver et al., 2010). The most substantial improvements were observed in *User Story & Backlog* from 2.73 to 4.55 and *Architecture (Monolithic vs. Microservices)* from 2.36 to 4.09. These topics are crucial for successful project management and software development, making the knowledge gains in these areas particularly noteworthy.

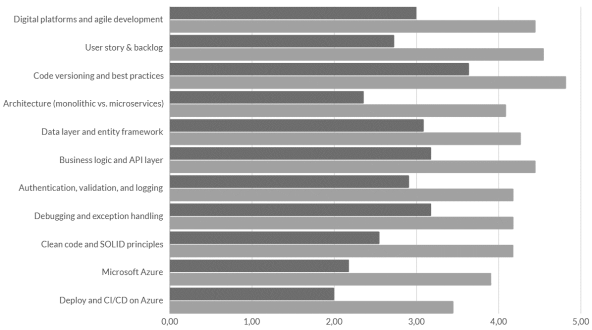

Figure 3. Analysing student proficiency in program topics

The average score for *Code versioning and Best Practices* also saw a significant rise, from 3.64 to 4.82, indicating successful adoption of best practices in code management. This is important because proper version control reduces the risk of errors and enhances team collaboration.

Security-related topics, such as *Authentication, Validation, and Logging*, increased from 2.91 to 4.18, while *Debugging and Exception Handling* rose from 3.18 to 4.18. These results demonstrate that students have gained a deeper understanding and the skills necessary to develop secure and reliable applications.

The smallest but still significant increases were observed in topics related to Microsoft Azure. *Deploy and CI/CD on Azure* increased from 2.00 to 3.45, while *Microsoft Azure* rose from 2.18 to 3.91. Although these results show significant progress, the relatively lower initial scores indicate that these areas were less familiar to students before the program. Nonetheless, the increase reflects successful transfer of fundamental knowledge and skills for using cloud platforms.

5 Conclusion

In the conclusion of this research, we emphasize the importance of collaboration between the academic community and industry in the development of innovative educational programs. Through the analysis of various approaches to agile software development, research on collaborative initiatives such as the JCC, and the application of project-based learning, key guidelines for successful integration of education and industry have been identified. Case studies of four student teams, who developed comprehensive solutions throughout the semester, underscore the significance of practical application of acquired knowledge. Presentations before industry experts allowed students not only to convey their creative ideas but also to receive constructive feedback directly from industrial professionals who served as mentors during the development process.

Our study confirms that collaboration with the industry provides students with useful insights, encourages a practical approach to learning, and enables the development of skills necessary for successful functioning in the job market. It is noteworthy to highlight that the agile development approach, supported by mentorship from industry experts, has empowered students not only to develop software solutions but also to gain relevant experience and skills essential for future professional endeavours.

The survey results indicate that the JCC program has significantly improved students' knowledge across all analysed topics. The greatest progress was achieved in project management and software architecture, while knowledge in Microsoft Azure and CI/CD processes also showed substantial improvement. These results confirm the program's effectiveness in preparing students for the challenges of modern software development and IT project management.

The research question we addressed in the research was: *What are the aspects of utility in integrating the JCC program into existing programs for a better comprehension of practical knowledge?* With the execution of all activities, starting from the research design throughout the implementation and feedback analysis we conclude that the integration of JCC program into existing programs emerges as a transformative approach that significantly enhances students' comprehension of practical knowledge in the realm of software development. This conclusion is rooted in several key aspects that underscore the utility of JCC integration:

- *Real-World application as a learning cornerstone*: JCC's emphasis on real-world application serves as a cornerstone for practical knowledge acquisition. By engaging students in hands-on projects mirroring industry scenarios, the program bridges the gap between theoretical understanding and pragmatic implementation, ensuring a more profound comprehension of software development concepts.
- *Industry mentorship as a guiding light*: The integration of industry experts as mentors throughout the JCC program brings a unique utility by providing students with direct guidance from professionals actively involved in the field. This mentorship model fosters a deeper understanding of industry best practices, offering students insights that extend beyond theoretical frameworks.
- *Technological relevance for current industry standards*: The utility of JCC lies in its commitment to technological relevance. Alignment with current technologies equips students with skills directly applicable to the ever-evolving landscape of software development.

Continuous feedback loop for iterative improvement: Integral to the JCC integration is the establishment of a continuous feedback loop. Regular interactions with industry mentors and presentations to experts create a platform for iterative improvement. This ongoing feedback mechanism refines students' practical skills, encouraging constant learning and adaptation.

The suggestion for further research is to investigate how continuous feedback and iterative improvements affect long-term retention and application of software development skills among students.

In essence, the integration of JCC into existing programs not only enriches theoretical understanding but propels students towards a practical comprehension of developing digital platforms. By addressing the dynamics of real-world application, mentorship, collaboration, technological relevance, continuous improvement, and job market preparedness, JCC emerges as a powerful catalyst in shaping the next generation of skilled and industry-ready software developers.

References

Abegglen, S., Burns, T., & Sinfield, S. (2021). Collaboration in higher education: Partnering with students, colleagues and external stakeholders. Journal of University Teaching & Learning Practice, 18(7), 1-6. doi:10.53761/1.18.7.01

Bocconi, S., Kampylis, P., & Punie, Y. (2012). Innovating teaching and learning practices: Key elements for developing creative classrooms in Europe. eLearning papers, 30, 1-13.

Carver, J. C., Jaccheri, L., Morasca, S., & Shull, F. (2010). A checklist for integrating student empirical studies with research and teaching goals. Empirical Software Engineering, 15, 35-59. doi:10.1007/s10664-009-9109-9

Devedžić, V. (2010). Teaching agile software development: A case study. IEEE transactions on Education, 54(2), 273-278. doi:10.1109/TE.2010.2052104

Divjak, B., Bađari, J., Grabar, D., Horvat, D., Vondra, P., & Rienties, B. (2023). Enhancing Learning Design through User Experience Research: Insights from a Survey in Four European Countries. In *Central European Conference on Information and Intelligent Systems* (pp. 213-221). Faculty of Organization and Informatics Varaždin.

Divjak, B., Grabar, D., Svetec, B. & Vondra, P. (2022). Balanced Learning Design Planning: Concept and Tool. Journal of information and organizational sciences, 46 (2022), 2; 361-375 doi:10.31341/jios.46.2.6

Easterling, L. A. (2022). Collaborative Faculty Development Activities in Higher Education Institutions: An Activity System Analysis. Indiana University.

Gannod, G. C., Troy, D. A., Luczaj, J. E., & Rover, D. T. (2015). Agile way of educating. In 2015 IEEE Frontiers in Education Conference (FIE) (pp. 1-3). IEEE. doi:10.1109/FIE.2015.7344019

Guo, S., & Yang, Y. (2012). Project-based learning: An effective approach to link teacher professional development and students learning. Journal of Educational Technology Development and Exchange (JETDE), 5(2), 5. doi: 10.18785/jetde.0502.04

Layman, L. (2006). Changing students' perceptions: an analysis of the supplementary benefits of collaborative software development. In 19th Conference on Software Engineering Education & Training (CSEET'06) (pp. 159-166). IEEE. doi:10.1109/CSEET.2006.10

Liebenberg, J., Huisman, M., & Mentz, E. (2015). The relevance of software development education for students. IEEE Transactions on Education, 58(4), 242-248. doi:10.1109/TE.2014.2381599

Malik, K. M., & Zhu, M. (2023). Do project-based learning, hands-on activities, and flipped teaching enhance student's learning of introductory theoretical computing classes?. Education and Information Technologies, 28(3), 3581-3604. doi:10.1007/s10639-022-11350-8

Martínez Fernández, S. J., Gómez Seoane, C., & Lenarduzzi, V. (2023). Applying project-based learning to teach software analytics and best practices in data science. International journal of engineering education, 39(2), 476-487.

Matus, M., Hajdin, G., & Balaban, I. (2023). Joint Creative Classroom as a Response to Industry 4.0 Demands. *In Central European Conference on Information and Intelligent Systems* (pp. 199-204). Faculty of Organization and Informatics Varaždin.

Nwodo, A. (2023). Beginning Azure Devops: Understanding and Using Azure Developer Services to Plan Work, Collaborate on Code Development, and Build and Deploy Applications. Wiley. isbn:978-1-394-16588-9

Oda, S., Inoue, M., & Yamazaki, A. K. (2017, November). Assessment of global competency for engineering students in a multicultural and multidisciplinary project based learning course. In *2017 7th World Engineering Education Forum (WEEF)* (pp. 439-443). IEEE. doi:10.1109/WEEF.2017.8467071

Peddireddy, K. (2024). Cloud Computing Platforms. In Emerging Trends in Cloud Computing Analytics, Scalability, and Service Models (pp. 186-199). IGI Global.

Scarani, S., Muñoz, A., Serquera, J., Sastre, J., & Dannenberg, R. B. (2020). Software for interactive and collaborative creation in the classroom and beyond: An overview of the soundcool software. Computer Music Journal, 43(4), 12-24.

Scaturro Heil, M. R. (2023). The Power of Digital Platforms: Facilitating the Collaboration of Undergraduate Computer Science Project Teams. International Association for Development of the Information Society. *Proceedings of the 20th International Conference on Cognition and Exploratory Learning in the Digital Age*, Madeira Island, Portugal.

Umar, U. S., Rana, M. E. (2024, January). Cloud Revolution in Manufacturing: Exploring Benefits, Applications, and Challenges in the Era of Digital Transformation. In 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS) (pp. 1890-1897). IEEE. doi:10.1109/ICETSIS61505.2024.10459473

Zantedeschi, V., Bellet, A., & Tommasi, M. (2020, June). Fully decentralized joint learning of personalized models and collaboration graphs. In International Conference on Artificial Intelligence and Statistics (pp. 864-874). PMLR.