On the Relation Between Open Project-Based Learning in Undergraduate Computer Science Education and Contemporary Technological Trends

Ruben Tous[a], Felix Freitag[b] and Josep Lluis Berral[c]
Universitat Politecnica de Catalunya, Barcelona, Spain

Keywords: Project Based Learning, Information Technology, Professional Skills, Curriculum Adaptation.

Abstract: Amidst rapid and constant technological change, keeping IT higher education curricula up to date is becoming increasingly challenging. For a decade, the course ”Project on Information Technologies” within the undergraduate computer science studies offered by a major technical university has been pursuing an effort of continuous curriculum adaptation and active learning practices based on an open project-based learning (PBL) methodology. This article researches the implications of employing an open-ended PBL approach, with a specific focus on the alignment of acquired skills with the trends in the professional landscape. Our analysis has identified a strong correlation between the technologies utilized in the projects and the contemporary technological trends in the areas of global technological focus, programming languages, server-side technologies, database management systems, and DevOps-related tools. For the study, we analyzed empirical data gathered from over 100 projects involving more than 400 students who enrolled in this reference course, belonging to the last year of the IT higher education programme in the last 10 years. The results suggest the open project-based design as a teaching means in the student's study plan for fostering the student the learning of prevailing current practical technologies.

1 INTRODUCTION

The skills and competencies essential for future professionals are a pivotal consideration in shaping any educational strategy. A significant challenge within the university system is showcasing its ability to adapt to the rapid pace of change in contemporary society. This challenge is particularly difficult in the IT domain, where the pace of technological change has been immensely accelerated.

For the past ten years, the course ”Project on Information Technologies” (PIT) within the undergraduate computer science studies offered by a major technical university has pursued continuous curriculum adaptation through the utilization of an open project-based learning methodology. Today, project-based learning (PBL) plays a fundamental role in the curricula of students pursuing bachelor's and master's degrees in Information Technology (IT) and related disciplines (Giannakos et al., 2017; Sindre et al., 2018; Fioravanti et al., 2018; Hulls et al., 2015). The degree of openness in the project statement and solution stands out as one of the most relevant parameters in the design of a course based on PBL (Hulls et al., 2015).

Different from a traditional PBL approach, where instructors define specific project goals and restrict potential solutions and which fosters a consistent skill development (Vasilevskaya et al., 2015), allowing students to determine project objectives and technology choices offers the advantage of naturally aligning the specific skills they desire to acquire through the course with trends in key IT technological domains (Sindre et al., 2018; Suseno et al., 2023). The course under examination employs this latter open-statement and open-solution approach in which, with the feedback from the instructor, the students are encouraged to drive the project definition.

To delve deeper into the implications of the open project choice, we examined the historical evolution of the technologies addressed in the projects developed in the course. For this, we analyzed empirical data drawn from over 100 projects undertaken by more than 400 students in the course during ten

[a] https://orcid.org/0000-0002-1409-5843
[b] https://orcid.org/0000-0001-5438-479X
[c] https://orcid.org/0000-0003-3037-3580

Tous, R., Freitag, F. and Berral, J.
On the Relation Between Open Project-Based Learning in Undergraduate Computer Science Education and Contemporary Technological Trends.
DOI: 10.5220/0012650600003693

In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024) - Volume 2*, pages 397-404
ISBN: 978-989-758-697-2; ISSN: 2184-5026

academic years, from 2012/2013 to 2022/2023. The resulting findings and conclusions, particularly those related to the alignment of acquired skills with contemporary trends in IT's technological domains, are presented in this study.

2 BACKGROUND

2.1 Course Description

The research undertaken in this paper was carried out within the framework of the semester-long course "Project on Information Technologies", of 6 ECTS credits (around 150 hours), and from the last year of the Bachelor of Computer Science offered by the School of Informatics at a major technical university. The study comprises ten academic years, from 2012/2013 to 2022/2023.

The course curriculum encompasses technical skills pertinent to information technologies, computer networks, and distributed applications. Additionally, it covers non-technical cross-disciplinary competencies such as teamwork, project conceptualization, and administration, as well as verbal and written communication. The primary aim of the course revolves around the collaborative development of a project by groups of typically 4 students. The project spans 15 weeks, equivalent to half an academic year.

2.2 Related Work

Numerous studies have advocated for project-based learning as a suitable methodology to attain effective competency-based education (Chinowsky et al., 2006; Gijselaers, 1996; A. Johnson, 1999; Gijselaers, 1996; Padmanabhan and Katti, 2002; Veselov et al., 2019; Anggraeni et al., 2023; Suseno et al., 2023). Several works have analyzed its effectiveness in higher education, particularly focusing on engineering disciplines (De los Ríos et al., 2010; Ruikar and Demian, 2013; Stewart, 2007; Gibbes and Carson, 2014; Requies et al., 2018). In the study of De los Ríos et al. (De los Ríos et al., 2010), the authors chronicle two decades of employing project-based learning within higher education engineering in the context of the final years of the undergraduate programme of the Technical University of Madrid, Spain. One of the findings drawn from this study is that project-based learning significantly enhances the connection between university education and the practical professional sphere. The work of Ruikar et al. (Ruikar and Demian, 2013) delves into the associations between project-based learning and engagement with the industry. The study of Stewart (Stewart, 2007) puts the focus towards the correlation between self-directed learning readiness and the results of project-based learning. Gibbes and Carson(Gibbes and Carson, 2014) applied activity theory analysis to investigate project-based learning. The authors document mixed results in learning outcomes due to contradictions identified within the activity system (such as uneven distribution of tasks or perceived time constraints stemming from community commitments). Numerous other studies also document particular experiences of project-based learning in higher education (e.g. (Rush et al., 2007; Hassan et al., 2008; Fernandes et al., 2013; Pereira et al., 2017; Requies et al., 2018)). The majority of these studies rely on qualitative evaluations of students' actions and achievements, which do not consistently enable to establish a direct cause-and-effect relationship between project-based learning instruction and positive student outcomes. Numerous works delve into the precise implementation of project-based learning within higher education in the field of IT. In the work of Sindre et al. (Sindre et al., 2018), the authors assess the appropriateness of project-based learning within the IT context in general, and with respect to the curriculum guidelines from the ACM/IEEE Task Force on Computing Curricula in particular. This study concludes that this approach effectively adjusts to the swiftly evolving skill requirements for upcoming IT professionals. Fioravanti et al. (Fioravanti et al., 2018) report an experience that integrates project-based learning and project management in a Software Engineering course.

There exists a limited number of works focusing on the extent of project statement and solution openness within project-based learning. Hulls et al. (Hulls et al., 2015) detail the transformation of (a portion of) a traditional C++ programming course (first-year of Mechanical and Mechatronics Engineering students at the University of Waterloo, Canada) into an open-ended project-based learning course. The research centered on motivational factors and concludes that this approach led to a substantial upsurge in student enthusiasm. An interesting facet of this study is the students' choice between predefined projects (such as a Roomba-like robot, an elusive alarm clock, or a maze-solving robot) or their individual project proposals. The authors report how the prevalence of students opting for their own concepts escalated over time, eventually constituting the majority.

3 MATERIALS AND METHODS

3.1 Data Acquisition and Analysis

To conduct this study, an analysis of the final deliverables from projects completed by attendees of the PIT course was necessary. The analysis covered a span of ten academic years, ranging from 2012/2013 to 2022/2023. A total of 125 projects, engaging over 400 students, underwent scrutiny. Data processing was facilitated through the utilization of project descriptions that students contributed to a Wiki platform as part of their concluding assignments.

Student project descriptions were manually processed and, for each project, a machine-readable project summary was generated in JSON format. Summaries included project focus (eg ”mobile app”), programming languages, technologies involved, and all tools of project management involved. However, because the information in the student project descriptions is of heterogeneous quality, in many cases, it was necessary to manually process the project deliverables one by one.

A Python-based tool was created to collate all the JSON summaries and perform automated computations on the evolving frequency of distinct technological components over time. Relative frequencies were adopted (e.g., ”50% of projects in the first semester of 2012/2013 employed Java”) instead of absolute values due to variations in project counts across different courses. The resultant numerical data points were saved in a collection of gnuplot-formatted data files and are presented as line charts to track the shifts in technologies across the years.

Furthermore, for a comprehensive analysis of the correlation between these shifts and real-world trends, a tool wase created (also with Python) to extract numerical information from Google Trends (Google, 2024).

4 RESULTS AND DISCUSSION

In the upcoming sections, we present the findings of the study. A line chart is provided for each project aspect, illustrating the changing frequencies of various student choices over time. Since PTI spans a full semester, each time period corresponds to one semester (20xx-1 represents the initial autumn semester, while 20xx-2 corresponds to the subsequent spring semester).

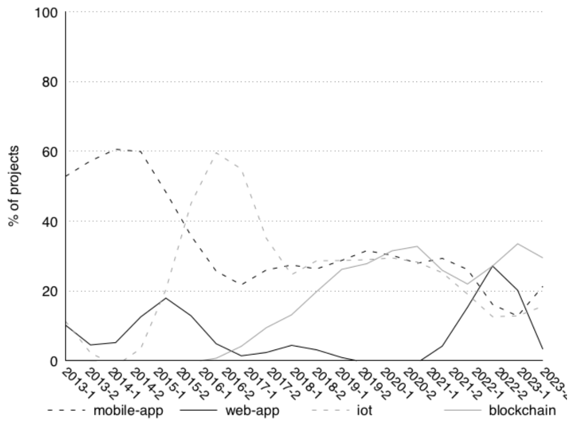

Figure 1: Line chart showing the evolution of the overall technological focus of the projects.

4.1 Technological Project Scope

Although the projects involve multiple components, they usually revolve around a main technology (e.g. blockchain). Figure 1 shows the evolution of the four most common global technological focus of the projects:

- *web-app*: projects whose main workload is dedicated to web application development. Typically, this kind of projects involves client-side web technologies (e.g. HTML, Angular, etc.) and server-side web technologies (HTTP servers, application servers, etc.).
- *mobile-app*: projects whose main workload is dedicated to mobile application development. Typically, this kind of projects involves client-side mobile technologies (Android SDK, iOS SDK, Unity, React Native, etc.) and server-side technologies such as Web APIs.
- *iot*: projects whose main workload is dedicated to Internet of Things (IoT) related technologies (e.g. Raspberry Pi, Arduino, webcam, sensors or actuators).
- *blockchain*: projects whose main workload is dedicated to blockchain related technologies (e.g. Ethereum, Solidity, MetaMask, etc.).

Comparing the students’ choices with the real technology trends in Figure 2, it can be seen that they closely correspond.

In a preceding period (spanning the 2000s decade) without accessible numerical data, the technological landscape of typical projects transitioned from non-HTTP distributed applications (such as CORBA or Java RMI) to projects primarily centered around web-based technologies, referred to as the *web-app* focus. Towards the end of that decade, the project emphasis

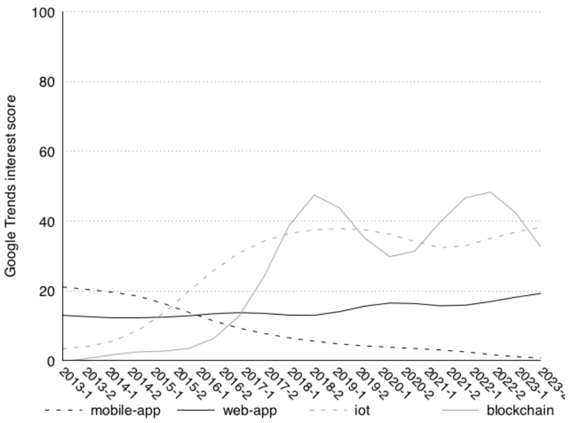

Figure 2: Google Trends interest score (related to search frequency) in the range 0-100 for the topics "web development", "android software development", "internet of things" and "blockchain" (Google, 2024).

shifted to the creation of mobile applications (*mobile-app*).

As the study commences in the academic year 2012/2013, mobile applications continued to dominate, yet a nascent trend, the Internet of Things (IoT), had already begun capturing student interest. The mid-2010s saw a prevalence of IoT projects, peaking during the first semester of the 2016/2017 academic year. Simultaneously, another technological wave emerged, the blockchain. This innovation swiftly grasped student attention, although its trajectory has been stabilizing in more recent times.

The approaches referenced are not exhaustive, but they do represent the most prevalent ones. Over recent semesters, certain projects have integrated elements linked to artificial intelligence. However, this aspect lies outside the primary course objectives and remains relatively uncommon within the scope of this study.

4.2 Programming Languages

Figure 3 displays the evolution of students' choices of programming languages over time. When comparing these results with the popularity of programming languages according to Google Trends (Figure 4), it can be observed that the programming languages used in the projects not only align with current trends but also have the potential to predict them.

The authors have noted that students tend to exhibit a natural inclination toward experimenting with new languages, which is inversely proportional to their willingness to use languages they perceive as outdated. IT professionals do not have the same freedom, neither the same disposition probably, and the adoption of new programming languages in the professional field is slower.

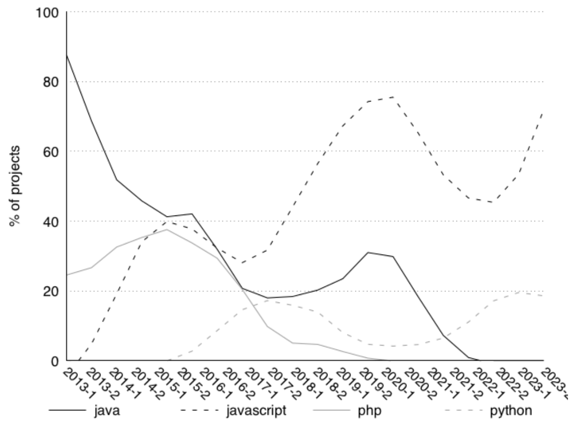

Figure 3: Line chart showing the relative frequencies of the students choice of programming languages over time.

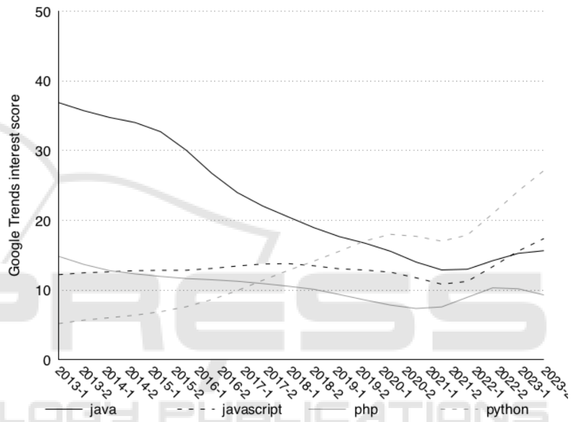

Figure 4: Popularity of programming languages according to Google Trends (Google, 2024).

Another conclusion drawn is that the students' preferences for using programming languages in their projects do not appear to align with the programming languages they have learned during their career (mainly C++ and Java). Through the last years, it has been observed how Node.js (JavaScript) and Python have been gaining share until they have become the most widely used languages, both in the client and the server side. JavaScript outperforms Python in our data, inversely to what is observed in Google Trends. The authors attribute this to the fact that Python is massively used by data science related tasks, which are generally out of the scope of the course.

4.3 Server-Side Technologies

Figure 5 illustrates the progression of server-side technologies employed in the projects. In the initial reporting period, students frequently opted for Java Servlet (e.g., the Apache Tomcat open-source Java Servlet Container) for the backend of their client-

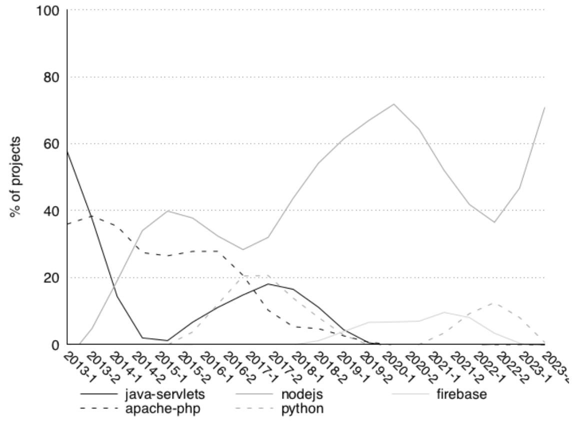

Figure 5: Line chart showing the relative frequencies of the students choice of server-side technologies over time.

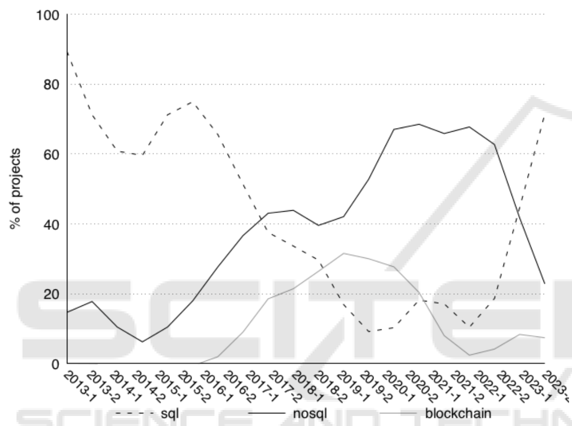

Figure 6: Line chart showing the relative frequencies of the students choice of database technologies over time.

server applications. This choice was likely influenced by its use in introductory labs within the course. Another common alternative during this period was the Apache HTTP server paired with PHP CGIs, a choice typically made by students with some professional experience (a circumstance less common at the beginning of the decade but now widespread). While the use of CGIs has dwindled, HTTP servers continue to feature prominently in many projects, with Apache gradually giving way to NGINX. In parallel with the rise of Node.js, the Express web application framework has seen increased adoption. Similarly, students who favor Python tend to select the Flask micro web framework for their backend solutions.

4.4 Database Technologies

Figure 6 illustrates the evolution of database technologies employed in the projects, revealing three distinct periods. Prior to 2015, the majority of projects favored object-relational database management systems, primarily centered around MySQL Community Edition, alongside options like PostgreSQL. The span between 2015 and 2021 witnessed a shift in preference towards NoSQL database management systems, with MongoDB taking the lead, accompanied by Cassandra, CouchDB, and others. Notably, MongoDB's impressive success, constituting about 90% of NoSQL database instances during that timeframe, can be attributed to its scalability and seamless integration with Node.js. This era also introduced the emergence of blockchain-related technologies. Despite not mirroring the functionalities of traditional databases precisely, numerous students adopted blockchain to store all the data (although such instances often lacked rationale and have been rectified over time). From 2021 onward, dynamics have evolved. The previously ascending trajectory of NoSQL databases has tapered,

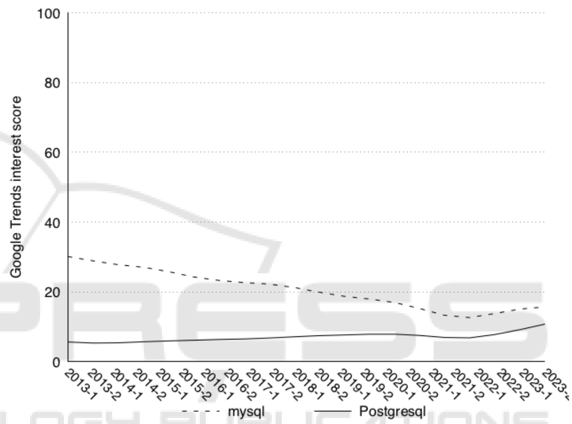

Figure 7: Google Trends interest score (related to search frequency) in the range 0-100 for the topics "MySQL" and "PostgreSQL" (Google, 2024).

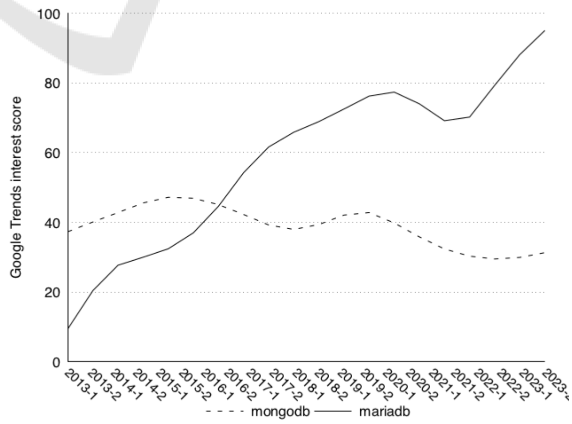

Figure 8: Google Trends interest score (related to search frequency) in the range 0-100 for the topics "MongoDB" and "MariaDB" (Google, 2024).

allowing object-relational databases to regain some prominence. However, there have been shifts in specific product preferences. Among object-relational database management systems, MySQL's predominant position has waned, making room for alternatives such as MariaDB and PostgreSQL (see figures 7 and 8). In the realm of NoSQL databases, MongoDB retains a strong presence in many projects, though certain students are exploring substitutes like CouchDB or RethinkDB. The momentum behind blockchain has also dwindled, with its application becoming more targeted, typically in conjunction with traditional database utilization.

4.5 DevOps

Figure 9 illustrates the progression of several DevOps-related tools employed within the projects. Technologies and practices associated with the DevOps methodology have experienced substantial growth in student projects, mirroring developments in the professional sphere (see Figure 10). The initial technology in this domain was Docker, which underwent rapid expansion. Presently, almost all projects integrate Docker containers for diverse software components. Subsequently, Kubernetes, the container orchestration system, emerged. While highly regarded by students, its complexity and limited benefits for prototype development lead many to shy away from its use. More recently, tools related to continuous integration and continuous delivery (CI/CD), such as Jenkins, have surged in popularity. Jenkins is a preferred choice among students, yet many also opt for the integrated features offered by GitHub or GitLab. A multitude of other tools in this field have surfaced in projects, including Ansible, Chef, Terraform, Prometheus, Grafana, ArgoCD, and numerous others. While not essential for prototype development, their prevalence in the professional realm motivates students to capitalize on the opportunity to acquire proficiency in their usage.

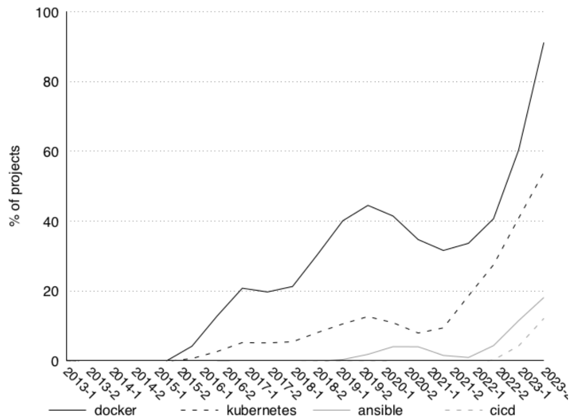

Figure 9: Line chart showing the relative frequencies of different DevOps related components found in the projects over time.

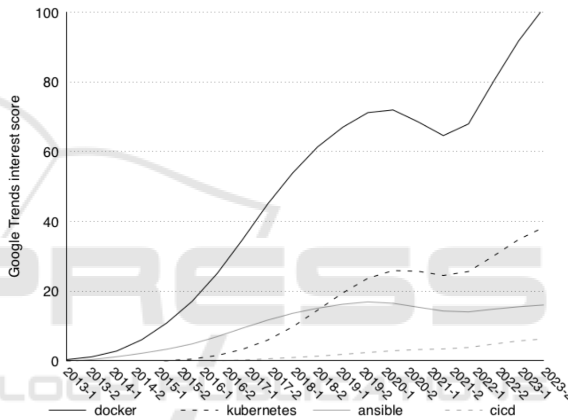

Figure 10: Google Trends interest score (related to search frequency) in the range 0-100 for the topics "Docker", "Kubernetes", "Ansible" and "CI/CD" (Google, 2024).

5 CONCLUSIONS

This study reviewed the evolution of technology choices in projects during a decade of an IT-related course with an open project-based learning methodology. Relevant conclusions are derived with respect to changes of the used technologies and the adaptation of the skills learned to the trends of the main IT technology domains. A systematic analysis of the data, with a special emphasis on the open-statement and open-solution methodology, aims to pave the way towards guiding future IT higher education courses design.

The analysis centered around five distinct facets: the global technological focus, the utilized programming languages, server-side technologies, database management systems, and DevOps-related tools. Our analysis demonstrated noteworthy shifts in these five dimensions as they were implemented in the course projects, occurring within relatively short timeframes and closely mirroring technological trends. Additionally, our results reveal the shrinking lifecycles of technological trends and the swift proliferation of potential solutions for each problem. This remarkably dynamic environment, which poses a definite challenge for IT educators, appears to be a non-issue for students. They have demonstrated an adeptness at aligning their project's learning content with prevailing technological trends, a flexibility facilitated by the open-ended project framework.

It is worth mentioning that some students opt for projects of a more fundamental nature, but they are very few and do not appear in the graphs. This is unsurprising, given that the course is taken by students in their final year, who are preparing to enter the professional realm. This can also be attributed to the fact that students have already acquired fundamental techniques and methodologies through a range of other courses throughout their academic journey. In fact, the synergy between courses, like the PBL-based course concentrating on current technologies, and other courses dedicated to fundamental methods and techniques, appears advantageous. This approach equips students with the capacity to engage with present technologies for immediate industrial applicability, while their fundamental knowledge empowers them to adeptly navigate the swift shifts in technological landscapes.

A question that remains is to determine if the number of students participating a the group project influences the observed relation between the open project-based methodology and technological trends. In future work it could be interesting to analyze whether small groups of 2-3 students are more likely to align their project with current technologies than bigger groups of 4-5 students, and become able to recommend an optimal project group size.

ACKNOWLEDGEMENTS

This work is partially supported by the Spanish Ministry of Science under contracts PID2019-107255GB, PID2019-106774RB-C21, PID2021-126248OB-I00 (MCIN/AEI/10.13039/501100011033/FEDER), PDC2023-145809-I00 (PDC/AEI/10.13039/501100011033), the Recovery and Resilience Facility of the European Union, and by the SGR programme of the Catalan Government under contracts 2021-SGR-00478 and 2021-SGR-01059.

REFERENCES

A. Johnson, P. (1999). Problem-based, cooperative learning in the engineering classroom. *Journal of Professional Issues in Engineering Education and Practice*, 125.

Anggraeni, D. M., Prahani, B. K., Suprapto, N., Shofiyah, N., and Jatmiko, B. (2023). Systematic review of problem based learning research in fostering critical thinking skills. *Thinking Skills and Creativity*, 49:101334.

Chinowsky, P., Brown, H., Szajnman, A., and Realph, A. (2006). Developing knowledge landscapes through project-based learning. *Journal of Professional Issues in Engineering Education and Practice*, 132:118–124.

De los Ríos, I., Cazorla, A., Díaz-Puente, J., and Yagüe, J. L. (2010). Project based learning in engineering higher education: two decades of teaching competences in real environments. *Procedia, Social and Behavioral Sciences*, 2:1368–1378.

Fernandes, S. R., Mesquita, D., Flores, M., and Lima, R. (2013). Engaging students in learning: Findings from a study of project-led education. *European Journal of Engineering Education*, 39:55–67.

Fioravanti, M. L., Sena, B., Paschoal, L. N., Silva, L. R., Allian, A. P., Nakagawa, E. Y., Souza, S. R., Isotani, S., and Barbosa, E. F. (2018). Integrating project based learning and project management for software engineering teaching: An experience report. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 806–811. ACM.

Giannakos, M. N., Aalberg, T., Divitini, M., Jaccheri, L., Mikalef, P., Pappas, I. O., and Sindre, G. (2017). Identifying dropout factors in information technology education: A case study. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1187–1194.

Gibbes, M. and Carson, L. (2014). Project-based language learning: an activity theory analysis. *Innovation in Language Learning and Teaching*, 8(2):171–189.

Gijselaers, W. H. (1996). Connecting problem-based practices with educational theory. *New Directions for Teaching and Learning*, 1996(68):13–21.

Google (cited Jaunary 2024). Google trends website. https://trends.google.com.

Hassan, H., Domínguez, C., Martínez, J., Perles, A., Albaladejo, J., and Capella, J. (2008). Integrated multicourse project-based learning in electronic engineering. *International Journal of Engineering Education*, 24:581–591.

Hulls, C., Rennick, C., Bedi, S., Robinson, M., and Melek, W. (2015). The use of an open-ended project to improve the student experience in first year programming. In *Proceedings of the Canadian Engineering Education Association*.

Padmanabhan, G. and Katti, D. (2002). Using community-based projects in civil engineering capstone courses. *Journal of Professional Issues in Engineering Education and Practice*, 128.

Pereira, M., Barreto, M., and Pazeti, M. (2017). Application of project-based learning in the first year of an industrial engineering program: Lessons learned and challenges. *Production*, 27.

Requies, J. M., Agirre, I., Barrio, V. L., and Graells, M. (2018). Evolution of project-based learning in small groups in environmental engineering courses. *Journal of Technology and Science Education*, 8:45–62.

Ruikar, K. and Demian, P. (2013). Podcasting to engage industry in project-based learning. *International Journal of Engineering Education*, 29:1410–1419.

Rush, M., Newman, D., and Wallace, D. (2007). Project-based learning in first year engineering curricula:

Course development and student experiences in two new classes at mit. *International Conference on Engineering Education, 2007 ICEE Annual Conference Proceedings, Coimbra, Portugal.*

Sindre, G., Giannakos, M., Krogstie, B. R., Munkvold, R. I., and Aalberg, T. (2018). Project-based learning in it education: Definitions and qualities. *UNIPED*, 41(2):147–163.

Stewart, R. A. (2007). Investigating the link between self directed learning readiness and project-based learning outcomes: the case of international masters students in an engineering management course. *European Journal of Engineering Education*, 32(4):453–465.

Suseno, N., Aththyby, A. R., Harjati, P., Ratini, ., and Turmudi, D. (2023). The role of problem formulation in problem-based learning to improve the students' curiosity. *Education Quarterly Reviews*, 6(1).

Vasilevskaya, M., Broman, D., and Sandahl, K. (2015). Assessing large-project courses: Model, activities, and lessons learned. *ACM Trans. Comput. Educ.*, 15:20:1–20:30.

Veselov, G., Pljonkin, A., and Fedotova, A. (2019). Project-based learning as an effective method in education. In *roceedings of the 2019 International Conference on Modern Educational Technology (ICMET 2019)*, pages 54–57.