An Exploratory Comparative Study on the Impacts of Technical Support on Student Successes in Computing Project-based Learning

Ahmad D. Suleiman
Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY, USA
as4300@rit.edu

Jan DeWaters
Institute for STEM Education
Clarkson University
Potsdam, NY, USA
jdewater@clarkson.edu

Daqing Hou
Software Engineering Department
Rochester Institute of Technology
Rochester, NY, USA
dqvse@rit.edu

Yu Liu
Electrical & Computer Engineering
Clarkson University
Potsdam, NY, USA
yuliu@clarkson.edu

David Shepherd
Department of Computer Science
Louisiana State University
Baton Rouge, LA, USA
dshepherd@lsu.edu

***Abstract*—In this research paper, we conducted a comparative study to measure the effectiveness of the provided technical support in computing project-based learning (PjBL) courses. Students learn much better by solving authentic real-world problems through PjBL. PjBL in computing education has proven to boost student motivation and engagement while enhancing academic performance. Crucial to PjBL in computing is the technical support that the instructors can provide to students, which is required for sustained, successful learning during project tasks. Without adequate support, PjBL will fall short of accomplishing its goals, leading to a rise in student frustration, a loss of motivation and engagement, and compromised learning outcomes. Measuring the impacts and effectiveness of the provided support is imperative for fostering continuous improvement, informed decision-making, and student success. It enables instructors to assess the impacts of their strategies, improve their approaches, and utilize their resources more effectively. To measure the impacts of technical support on students during PjBL, we performed a comparative study on two undergraduate computing courses in Spring 2024, Fundamentals of Software Engineering and Database Systems. In both courses, students work on two assigned projects, one with little and inadequate support and the other with adequate support. We administered a post-survey after each project was completed. We analyzed students' self-reflection responses across four sub-scales, support satisfaction, motivation, self-efficacy, and project satisfaction. The results show a statistically significant increase in the supported project in the Fundamentals of Software Engineering course and no difference in the Database Systems course. This finding is likely due to other differences between the two projects for the Database Systems course beyond support, such as project scale. Qualitative analysis of students' responses also indicates the need for support by students in the less supported projects. Based on our experience, we reflect on the question of what would constitute a good design for studies that seek to compare two different student learning experiences.**

***Index Terms*—Project based learning, Academic support, Course assessment, Technical Challenges**

I. Introduction

Students learn much better by solving authentic real-world problems through project-based learning (PjBL) [1]. PjBL in computing education has proven to boost student motivation and engagement while enhancing academic performance [2], [3]. Crucial to PjBL in computing is the technical support that the instructors can provide to students, which is required for sustained, successful learning during project tasks. Such support is integral to PjBL's ability to guide, motivate, and enable students to develop essential skills and achieve meaningful learning outcomes.

Without adequate support, PjBL will fall short of accomplishing its goals, leading to a rise in student frustration, a loss of motivation and engagement, and compromised learning outcomes [4]. Measuring the impacts and effectiveness of the provided support is imperative for fostering continuous improvement, informed decision-making, and student success. It enables instructors to assess the impacts of their strategies, improve their approaches, and utilize their resources more effectively.

However, evaluating the effectiveness of support can be challenging due to several inherent issues and concerns, such as subjective assessment methods and the difficulty in quantitatively measuring the provided support. In addition, there can be other important issues to consider when designing such studies, such as how to properly control the dynamics in the learning environment to enable a non-confounded comparative study, diverse learning needs, and fairness in the support that students receive.

To measure the impacts of technical support on students and PjBL, we performed a comparative study at Clarkson University. The comparative study was conducted in two undergraduate computing courses in Spring 2024, Fundamentals of Software Engineering and Database Systems. In each of the two courses, students worked on two assigned projects, one with little and inadequate support and the other with adequate support. In the adequately supported projects, we utilized several support strategies, including additional lectures, laboratory exercises, links to written and video tutorials, incremental project tasks, and project checkpoints. We administered a self-reflection survey after each project was completed to measure students' motivation toward learning, self-efficacy, satisfaction with support, and general project satisfaction. In the adequately supported projects, we utilized several support strategies, including additional lectures, laboratory exercises, links to written and video tutorials, incremental project tasks, and project checkpoints. In this research paper, we present the results of our comparative study, which addresses the following research questions:

- What is the effectiveness of the provided support on student outcomes?
- What is a good study design that measures the effectiveness of support provided to students?

Based on the experience of this comparative study, we also reflect on the question of what would constitute a good design for such studies.

The rest of the paper is organized as follows: We discuss the research background in Section II. We elaborate more on our comparative study design and detail the two student projects in each course in Section III. We present the quantitative and qualitative results of the comparative study in Sections IV and V respectively. We discuss the results in Section VIII and reflect on the limitations and future directions of our comparative study design in Section VII. Finally, we conclude the study in Section VIII.

II. Background

In this section, we will discuss why technical support is crucial for students in PjBL and explore some of the challenges of evaluating such support, with the aim of designing a study that provides insights into the impact of different levels of support on student outcomes.

A. Technical Support

Students in computing PjBL courses often encounter technical challenges that can significantly impact their motivation and overall success. Addressing these challenges effectively requires understanding the specific areas where students need support. One of the most common technical challenges faced by students is the installation and configuration of software packaged tools necessary for their projects [5], [6], such as challenges in setting up programming environments, configuring necessary libraries, configuring databases, etc. Technical support in this area is crucial, as difficulties here can cause significant delays and frustration, leading to a loss of confidence and motivation even before the project begins [7].

Another critical area where students require support is bridging the gap in prerequisite knowledge [8]. PjBL courses often assume a certain level of familiarity with foundational concepts and tools. However, students come from diverse educational backgrounds and may not possess the necessary skills or knowledge to tackle the projects effectively. This gap can include understanding programming languages, version control systems like Git, project management tools like Jira, and other project-specific knowledge. Using teaching assistants [8] and providing additional resources, supplementary lectures, and personalized assistance can help mitigate these gaps, ensuring all students are on a level to reasonably contribute to the project work.

As students progress through their projects, they also often face technical challenges related to the actual execution of the project tasks. These challenges can range from debugging code and resolving software bugs to integrating different components of the project. The unpredictable nature of these challenges, especially in open-ended and greenfield projects, means that students must be equipped with problem-solving skills and have access to timely support. Without adequate support, students may struggle to complete their projects.

Lack of support from instructors can lead to an increased dropout rate by students [9]. Instructors have provided targeted support to students to help maintain student motivation and enhance their learning experience [10] [11] [12] [13] [14].

B. Evaluating Provided Support

It is essential to measure the impacts of the provided support on PjBL outcomes to promote student success, informed decision-making, and continuous improvement. It helps educators assess the impacts of their strategies, improve their methods, and use of resources effectively. Nonetheless, evaluating the effectiveness of support can be challenging due to several inherent reasons. One significant challenge is determining how to quantitatively measure the provided support [15]. Support in an educational environment often involves qualitative elements such as emotional encouragement, personalized feedback, and mentorship, which are not easily captured through numerical metrics. In addition, the reliance on subjective assessment methods and measurements can lead to inconsistent and biased results [16].

Furthermore, there are other important considerations to take into account when designing studies to evaluate the effectiveness of support. For example, properly controlling the dynamics in the learning environment can be complicated [17]. Learning environments are inherently dynamic, influenced by factors such as students' motivation, project attributes, classroom interactions, teaching styles, and other external conditions and factors. Ensuring that these conditions and factors are controlled or accounted for is crucial to obtaining valid comparative results.

Diverse learning needs from students present another layer of complexity [18]. Students have varying skills, competencies, backgrounds, and learning preferences, which means that support strategies that work well for some students may not be effective for others. Designing studies that accommodate this diversity and provide equitable support across different student populations is essential for obtaining comprehensive and inclusive findings.

Ensuring fairness in the support that students receive is also a critical concern. Any evaluation should consider whether all students have equal access to the support being provided. For example, comparative studies that involve giving support to some students and not others can be considered unfair and thus avoided.

III. Methodology

In this section, we elaborate on our exploratory comparative study design and detail the courses and projects students worked on. We also describe the support that we provided to students in the adequately supported project in each course.

A. Comparative Study Design

Our main aim was to measure the effectiveness of our support on student outcomes. We designed an exploratory comparative study, which involves evaluating two or more groups or conditions to identify differences and similarities, making it suitable for assessing the impact of varying levels of technical support in PjBL courses. Note that our study was intended to be *exploratory*, as our goal was to generate *initial* evidence and insights about the effects of technical support on student outcomes in PjBL. Our approach involves assigning two projects to students in the same course, one with little and inadequate support and the other with adequate support. The two-project approach ensured that all students were given equal amounts of technical support. A post-survey is administered at the end of each project to collect students' self-reflection on the support, project, and course.

The study used motivation as a key indicator of student outcomes because of the established links with student engagement, self-confidence, self-efficacy, and academic success [19]–[21]. The survey instrument used across all sampling events was framed from the Motivation Strategies for Learning Questionnaire (MSLQ) [22], which contains 81 questions in two main categories that are designed to be segmented to meet the needs of the researcher or instructor. We adapted items to measure student motivation in terms of their intrinsic goal orientation, extrinsic goal orientation, task value, help-seeking, peer learning, metacognition, time and study environment, self-regulation, and elaboration. We selected 24 questions from the MSLQ to represent student motivation, supplemented by 9 self-efficacy items, adapted from [23]. An additional 14 questions were created specifically for this project: 11 questions to measure students' perception of the project regarding satisfaction and engagement, and 3, framed specifically for each project, to measure students' satisfaction with the provided support on lack of prerequisite knowledge, challenges in installation and configuration of packaged software tools, and challenges while completing project tasks. All questions used a 5-point Likert-type scale. The overall reliability of the questionnaire, as measured with Cronbach's alpha, was 0.965, with alpha values ranging from 0.906 to 0.957 across the three subscales.

The questionnaire also contains five open-ended questions designed to provide a better perspective on how students perceive the provided support and the project's value, including one for each of the three support categories, one for general support comments and suggestions, and one for job preparedness in the workplace.

The questionnaire was administered to all students following completion of each project using the Qualtrics platform. The data was then downloaded into Excel for analysis. Likert-type responses were converted to a numerical rating scale based on the preferred direction of response, ranging from 1 (least preferred) to 5 (most preferred). These numerical values were used to calculate the average mean values for each question and each sub-scale.

We employed an inductive and semantic thematic analysis approach from the guideline outlined in [24] to analyze the open-ended responses. This method ensured that the coding and theme development were guided by the data content. Responses were initially organized by question in a spreadsheet. Through an open coding process, each response was analyzed and assigned one or more codes as applicable. New codes were generated whenever necessary, based on the content of the responses.

B. The Courses and Projects

The comparative study was conducted in two undergraduate computing courses in Spring 2024, Fundamentals of Software Engineering and Database Systems at Clarkson University.

1) Course 1: Fundamentals of Software Engineering: Students enrolled in this course are mostly in their junior and senior years in computer engineering and software engineering majors. The main objective of the course is to improve students' software engineering skills by doing more diverse medium-scale projects. A secondary objective is to learn to use the Python programming language and the software ecosystem built around it.

- *GPU Project (little support)*: In the GPU project, students worked in pairs for a period of 2 weeks. When working with images, it can be very inefficient/slow to process tens of thousands or even millions of pixels sequentially, as done when using CPUs. This project tasked students to use graphics processing unit (GPU) programming to implement parallel processing to efficiently speed up heavy-computing image processing operators when manipulating millions of pixels, such as edge detection, vertically flipping an image, doubling an image, etc.
- *RSA Project (adequate support)*: In the RSA project, students worked in pairs for a period of 2 weeks. The goal of this project is to help students understand and implement the Rivest-Shamir-Adleman (RSA) cryptographic algorithm [25] through a series of hands-on tasks.

2) Course 2 - Database Systems: Students enrolled in this course are mostly in their junior and senior years in computer engineering, software engineering, and computer science majors. The goal of this course is to provide students with a solid foundation in database design and maintenance, equipping them with the essential skills needed to work with large-scale software systems. The course uses an interactive textbook on zyBooks [26].

- *Breached Project (little support)*: The Breached project lasted two weeks and engaged students in teams of two. The objective is to develop a secure web service that allows users to verify whether their email and/or password appear in a large database of millions of breached credentials (purchased from the black market). The project involves two main tasks: first, hashing and storing millions of breached email and password pairs into a database; and second, creating a simple website using the Flask web framework where users can check their credentials against the database. A key design goal is to convince a user to trust that the system will not steal or leak their credentials.
- *University Management Project (adequate support)*: Students worked in 3-5 person teams to complete this five week project. The goal is to design and implement a comprehensive web-based university management system using the Django web framework connected to a MySQL database. The system will integrate an existing university database that the students have worked on earlier in the semester to support different functionalities for three types of users: admins, professors, and students. The project also provides opportunities for students to engage in the full life cycle of database design.

C. Provided Support

We provided the following support strategies on the adequately supported projects:

1) Additional Lectures: Additional lectures were given to students specifically to support the project. In the RSA project, for example, there were additional lectures to cover mathematics fundamentals such as greatest common divisor (GCD) and modular multiplicative inverse (MMI). In the university management project, recorded lectures were given to students on the Django web framework and MySQL Python connector.

2) Laboratory Exercise: Laboratory exercises were designed to allow students to practice the necessary skills needed for the project. In the RSA project, for example, a pre-project laboratory exercise was given to students to practice GCD, extended Euclidean algorithm (EEA), and string to integer encoding. In the university management project, there was a laboratory exercise to support students with the installation and usage of the Django Python web framework

3) Links to Written and Video Tutorials: Students were provided with online links to written and video tutorials that can be helpful to the project. Links to Django models, Django templates, and HTML tutorials were given in the university management project. Slides on RSA fundamentals were given in the RSA project.

4) Incremental Project Tasks: The RSA project is divided into four parts, intentionally designed to be incremental, with each subsequent task being more challenging than the previous one. This allows students to gradually adjust to the project's complexity, thereby sustaining their confidence.

5) Project Checkpoints: The university management project has four checkpoints, during which students must present their design and implementation progress to the course instructor in class for feedback. This allows students to receive constructive feedback from the instructor, helping them identify and correct issues early in the project. It also ensures that students stay on track with their project milestones, reducing the risk of falling behind schedule.

IV. Quantitative Results

This section presents the quantitative analysis of a comparative study conducted in two courses. All reported statistical significance results are from paired t-tests between the two projects using a one-tailed t-test with $P_{value} < 0.05$ as the threshold for statistical significance. Effect sizes of statistical signal results are also reported using Cohen's d [27]. The pooled standard deviation (SD_{pooled}) of the two projects is used to calculate Cohen's d value.

A. Fundamentals of Software Engineering Course

In the fundamentals of software engineering course, 28 students enrolled in the course and participated in the projects. We collected 16 responses in the GPU project and 12 in the RSA project, and we were able to match 10 responses from both projects. Table I presents a quantitative analysis of support questions, focusing on the differences in student perceptions between the GPU project with mean $\bar{x}_u$ and the RSA project (i.e., adequately supported) with mean $\bar{x}_s$.

Students felt that there was sufficient and adequate support given to them in the process of completing the RSA project tasks more than the GPU project, as indicated by the large effect size (based on the standard interpretation of d by the literature) in Table I. While not statistically significant, there was an increase in how students felt about the support they received in understanding project prerequisite knowledge in the RSA project compared to the GPU project. Students did not perceive any difference in the support provided for the installation and configuration of software tools between the projects.

TABLE I
QUANTITATIVE ANALYSIS OF SUPPORT QUESTIONS IN FUNDAMENTALS OF SOFTWARE ENGINEERING COURSE

Support Questions	$\bar{x}_u$	$\bar{x}_s$	d	P_{value}
I was given sufficient and adequate support in understanding the project's prerequisite knowledge, concepts, and skills.	3.9	4.5	-	0.084
There was sufficient and adequate support given to troubleshoot installation and configuration issues.	4.1	4.2	-	0.424
There was sufficient and adequate support given in the process of completing the project tasks	3.7	4.7	1.1	**0.016**$^{+}$

$^{+}$ Statistically significant differences at $P < 0.05$

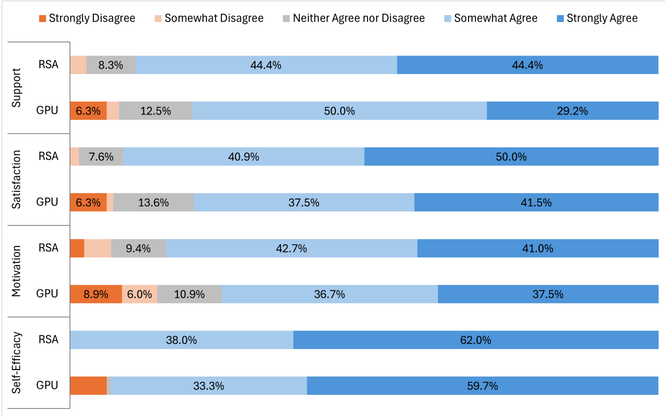

Fig. 1. Survey Responses to the Four Sub-scales between the Two Projects in Fundamentals of Software Engineering Course

Fig. 1 presents average student responses to each of the four question sub-scales (i.e. support, project satisfaction, motivation, and self-efficacy). In all sub-scales, the average mean score for the RSA project is better than the GPU project, although the differences were not significant. The percentage of respondents who somewhat agree or strongly agree with the support-related questions increased from 79.2% (for GPU) to 88.9% (for RSA). For project satisfaction questions, it increased from 79.0% to 90.9%, for motivation questions, from 74.2% to 83.7%, and from 93.1% to 100.0% for the self-efficacy sub-scale.

Although there were no significant differences between the two projects in the average mean scores on the motivation, self-efficacy, and project satisfaction subscales, there was a statistically significant increase in the RSA project in some individual survey questions, as shown in Table II. For example, student satisfaction with the RSA project is better than the GPU project in terms of understanding course content and material, receiving technical guidance from the instructional team, and considering taking a course with similar projects. Students were also more motivated by the RSA project as it provoked their curiosity to investigate topics beyond the course requirements, they put in more effort to finish it, and they had better peer learning. The effect size of these significant results ranges from medium to large.

TABLE II
Statistical Significant Survey Question Responses between Two Projects in Fundamentals of Software Engineering Projects

Question	$\bar{x}_u$	$\bar{x}_s$	d	P_{value}
Q11: This class provided material that provoked my curiosity to investigate topics beyond the course requirements.	3.9	4.7	0.88	**0.043**$^+$
Q16: Even when the course materials were dull and uninteresting, I managed to keep working until I finished them.	3.6	4.4	0.83	**0.018**$^+$
Q23: When studying or working on assignments for this course, I often set aside time to discuss course material with a group of students from the class.	3.1	3.9	0.60	**0.043**$^+$
Q24: I'm certain I understood the most difficult material presented in this course.	3.8	4.6	0.91	**0.035**$^+$
Q42: The instruction team (instructors and teaching assistants) was helpful in providing technical guidance when I worked to complete assignments.	3.9	4.5	0.65	**0.041**$^+$
Q43: These assignments contributed to my understanding of course content/ material.	4.0	4.8	0.92	**0.035**$^+$
Q44: I would consider enrolling in other courses that use these types of assignments.	3.7	4.5	0.89	**0.035**$^+$

$^+$ Statistically significant differences at $P < 0.05$

TABLE III
Quantitative Analysis of Support Questions in Database Systems Course

Support Questions	$\bar{x}_u$	$\bar{x}_s$	d	P_{value}
I was given sufficient and adequate support in understanding the project's prerequisite knowledge, concepts, and skills	3.75	3.92	-	0.377
There was sufficient and adequate support given to troubleshoot installation and configuration issues	3.75	4.08	-	0.219
There was sufficient and adequate support given in the process of completing the project tasks	4.17	3.92	-	0.306

B. Database Systems Course

In the database systems course, 64 students enrolled in the course and participated in the project. We collected 29 responses in the breached project and 20 in the university management project, and were able to match 12 responses from both projects. Table III presents a quantitative analysis of support questions, focusing on the differences in student perceptions between the Breached project with mean $\bar{x}_u$ and the University project (i.e., adequately supported) with mean $\bar{x}_s$.

Unlike the software engineering course, the database systems course did not show any statistically significant improvements in support across the three evaluated questions. There was no significant difference in how students felt about the support given to them in the process of completing the project tasks, or how they felt about the support they received in understanding project prerequisite knowledge in the Breached project compared to the University project. Students did not perceive any difference in the support provided for the installation and configuration of software tools between the two projects either.

We present average student responses to each of the four question sub-scales (i.e. support, project satisfaction, motivation, and self-efficacy) in Fig. 2. In all sub-scales, the average mean in the University project is slightly better than the Breached project. The percentage of respondents who somewhat agree or strongly agree with the support-related questions increased from 91.6% to 95.5%. For project

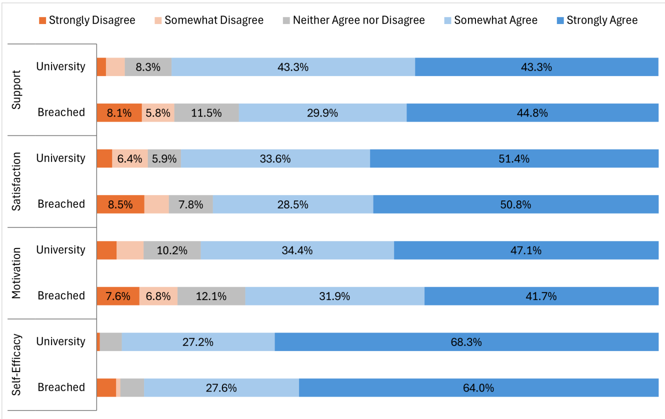

Fig. 2. Survey Responses to the Four Sub-scales between the Two Projects in Database Systems Course

satisfaction questions, it increased from 79.3% to 85%, for motivation questions, it increased from 73.6% to 81.5%, and from 91.6% to 95.6% for the self-efficacy sub-scale.

Although there were no significant differences between the average mean scores on the motivation, self-efficacy, and project satisfaction subscales, there was a statistically significant increase in the University project in two individual survey questions, as shown in Table IV with medium effect sizes. Students in the University project tend to seek more help from their peers. In addition to help-seeking, students explain the material to their peers more often in the University project. These were consistent with the fact that the University project was a much larger project than the Breached project, in terms of both team size and project length.

V. Qualitative Results

This section presents the results of the thematic analysis of the support-related open-ended questions from both courses. The key themes related to student support will be highlighted, along with representative comments. Note: The quoted comments have been edited (where applicable) for spelling and grammar errors to improve clarity and readability.

A. Fundamentals of Software Engineering Course

Few respondents approach the open-ended questions. There were 11 comments on the GPU project and 6 comments on

TABLE IV
Statistical Significant Survey Question Responses between Two Projects in Database Systems Course

Question	$\bar{x}_u$	$\bar{x}_s$	d	P_{value}
Q12: When I couldn't understand the materials in this course, I would ask another student in the class for help.	3.2	3.9	0.55	**0.048**$^+$
Q33: When studying for this course or working on course projects, I often tried to explain the material to a classmate or friend.	3.0	3.8	0.74	**0.027**$^+$

$^+$ Statistically significant differences at $P < 0.05$

the RSA project across all the questions. We have identified 5 key themes, as follows:

1) Appreciation of Additional Lectures and Laboratory Exercise: Students appreciated the support of additional lectures and laboratory exercises. One student from the RSA Project comments:

> *"There was dedicated lecture time and an additional practice lab to ensure total understanding of the topics required for understanding how to solve the project problem."*

Another student in the RSA project comments:

> *"A lot of the lecture and material presented provided adequate examples to understand the fundamental materials."*

One student from the GPU project comments:

> *"The professor gave us a lecture and assigned a lab on how GPUs work."*

2) Self-Directed Learning: Due to the lack of sufficient support provided in the GPU project, students rely on themselves to learn independently and on external resources to understand and complete the project. A student in the GPU project comments:

> *"I felt like I had to learn all of these things entirely on my own."*

Another GPU project student comments:

> *"To understand GPU Programming with CUDA, I mainly referred to documentation and explanations found in various online sources (example codes, videos, etc.), along with sample code provided by the instructor."*

3) Installation and Configuration Support: In the GPU project, a student expresses the need for more support in the installation and configuration of tools:

> *"It would have been beneficial to the course to provide students with alternate means or directions to access GPU capabilities, such as utilization of VS Code and installing Jupyter Notebook and CuPy for kernel access."*

While in the RSA Project, a student felt there was adequate support:

> *"I was not in a position to need support in troubleshooting issues; the provided materials and documentation on how to do so were satisfactory."*

4) Tasks Support: Students felt that the support during project tasks was generally adequate, though some suggested that more emphasis could be placed on certain topics. A student in the GPU Project comments:

> *"The support provided was adequate, and I believe the topic could be made a larger portion of the course as its importance grows in the industry."*

The RSA project benefited from clear explanations, reducing the need for further support during development. A student comments:

> *"The main support for the project for me was the definitive explanation of topics required for development. Further support during development was not needed."*

5) Challenging and Intriguing: Students felt that the support during project tasks was generally adequate, though some suggested that more emphasis could be placed on certain topics. One student comments:

> *"Although adequate, the project was challenging for me."*

Another student comments:

> *"It was intriguing to see the differing ways that one problem could be solved."*

B. Database Systems Course

There were 24 comments on the Breached project and only 5 comments on the University project across all the questions. We have identified 3 key themes as follows:

1) Lack of Prerequisite Knowledge: Many students in the Breached project felt that the skills necessary to complete the project were not taught adequately before the project was assigned, leading to learning on the fly. A student wrote the following comment:

> *"A lot of the skills required for this project were not taught to us ahead of time, we had to learn them on the fly during the project."*

Another student also comments:

> *"While I understand this isn't a Python course, he didn't go over Python a lot and kind of assumes you know it already."*

A student wished there was a support activity to learn some of the needed project knowledge:

> *"...a quick workshop outside of class for Flask would have been helpful to learn the basics of Flask in a learning environment where we can ask questions I would have gone."*

2) No Support Needed: Some students mentioned they did not need any assistance in the Breached project. A student comments:

> *"Didn't require help during download, however, Professor *** is generally very helpful"*

Another student comments:

> *"I didn't need assistance so I didn't know the best choice to select above. But the professor constantly sent out emails to notify us of his and the TA's office hours."*

3) Adequate Support: Some students in both projects felt that the support given was adequate. A student in the Breached project comments:

> *"Yes, the support from the professor was adequate to complete the project in time."*

Another student in a university project comments:

> *"adequate and useful to help us complete the project"*

VI. Discussion

In the fundamentals of software engineering course, both projects were well-received by the students, but they were more satisfied with the RSA project as they felt it was more adequately supported during project tasks than the GPU project. Students did not perceive any difference in the support they received during the installation and configuration of tools. One factor that might influence this is that in the GPU project, students used Google Colab, a Jupyter Notebook service provided by Google, which requires no setup and offers free access to computing resources such as GPUs [28]. There was also a significant increase in the supported project in seven other survey questions on intrinsic goal orientation, effort regulation, peer learning, and project satisfaction.

On the other hand, students in the database systems course did not perceive any difference in support between the two projects. The lack of difference might be caused by the difference in scale between the Breached and University projects. The University project requires more effort from students to complete, has a larger team size, and has a longer project duration. Both projects also use the Django web framework. In hindsight, it is not feasible to measure the difference in support given to students in these two projects. The larger team size was also reflected in the statistical increase in help-seeking and peer learning in the University project.

VII. Reflection and Future Direction

Reflecting on the question of "what would constitute a good design for such studies", while this current design provided some insight into the effectiveness of our support, this study has some limitations and several areas for improvement. This study was limited by the small number of survey respondents, particularly in the Fundamentals of Software Engineering course. While we believe the survey instrument used in this study is appropriate, improvements will be made in the support questions shown in Table I, as they can be perceived as double-barreled (i.e., ask two things). Carryover effects for repeated exposure to the intervention was limited by selecting projects with less in common. Enhancements can be made by selecting project pairs that are similar in scale but different in the technical stack to avoid overlap.

VIII. Conclusion

Students learn much better by solving authentic real-world problems through PjBL. Crucial to PjBL in computing is the technical support provided by the instructors. Without adequate support, PjBL will fall short of accomplishing its goals, leading to a rise in student frustration and a loss of motivation and engagement. To enable instructors to assess the impacts of their strategies, improve their approaches, and more effectively utilize their resources, it is imperative to measure the effectiveness of the provided support. To measure the impacts of technical support on students and PjBL, we performed a comparative study on two undergraduate computing courses in Spring 2024, Fundamentals of Software Engineering and Database Systems. In both courses, students work on two assigned projects, one with little and inadequate support and the other with adequate support. At the end of each project, we administered a self-reflection survey across four sub-scales: support satisfaction, motivation, self-efficacy, and project satisfaction.

The quantitative analysis shows the statistical significance of the adequate support we provided in the fundamentals of the software engineering project. Meanwhile, in the database systems course, there is no difference in the support provided between the two projects, likely due to the difference in scale between the projects. In hindsight, it may not be feasible to measure the difference in support given to students in projects with relatively large differences in scale. Qualitative analysis reveals that students appreciate the supported projects and express a need for more support for the less supported projects in both classes. It also indicates that when students perceive a lack of support, they resort to learning independently, using external resources to understand and complete the project work.

Future work will use the lessons learned from this study to refine and improve the study design. This includes selecting project pairs that are similar in scale but different in their technical stack to avoid overlap and ensure a balanced comparison. Additionally, efforts will be made to increase the number of survey respondents to gain more comprehensive insights.

Acknowledgment

This work was partially supported by the U.S. National Science Foundation Awards DUE-2111318 and DUE-2111294.

References

[1] S. Boss and J. Krauss, *Reinventing Project-Based Learning: Your Field Guide to Real-World Projects in the Digital Age*. International Society for Technology in Education, 2022. [Online]. Available: https://books.google.com/books?id=ZK-FEAAAQBAJ

[2] P. Morais, M. J. Ferreira, and B. Veloso, "Improving student engagement with project-based learning: A case study in software engineering," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 16, no. 1, pp. 21–28, 2021. [Online]. Available: https://doi.org/10.1109/RITA.2021.3052677

[3] C. Gupta, "The impact and measurement of today's learning technologies in teaching software engineering course using design-based learning and project-based learning," *IEEE Transactions on Education*, vol. 65, no. 4, pp. 703–712, 2022. [Online]. Available: https://doi.org/10.1109/TE.2022.3169532

[4] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating project-based learning: Sustaining the doing, supporting the learning," *Educational psychologist*, vol. 26, no. 3-4, pp. 369–398, 1991. [Online]. Available: https://doi.org/10.1080/00461520.1991.9653139

[5] M. A. Nelson and L. Ponciano, "Experiences and insights from using github classroom to support project-based courses," in *2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)*. IEEE, 2021, pp. 31–35. [Online]. Available: https://doi.org/10.1109/SEENG53126.2021.00013

[6] D. Davenport, "Experience using a project-based approach in an introductory programming course," *IEEE Transactions on Education*, vol. 43, no. 4, pp. 443–448, 2000. [Online]. Available: https://doi.org/10.1109/13.883356

[7] Z. Jeremic, J. Jovanovic, and D. Gasevic, "An environment for project-based collaborative learning of software design patterns," *International Journal of Engineering Education*, vol. 27, no. 1, p. 41, 2011.

[8] A.-M. Majanoja and T. Vasankari, "Reflections on teaching software engineering capstone course." in *CSEDU (2)*, 2018, pp. 68–77. [Online]. Available: https://scitepress.org/papers/2018/66656/66656.pdf

[9] D. F. Onah, J. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: behavioural patterns," *EDULEARN14 proceedings*, pp. 5825–5834, 2014. [Online]. Available: https://wrap.warwick.ac.uk/65543/

[10] I. Stamouli, E. Doyle, and M. Huggard, "Establishing structured support for programming students," in *34th Annual Frontiers in Education, 2004. FIE 2004.* IEEE, 2004, pp. F2G–5. [Online]. Available: https://doi.org/10.1109/FIE.2004.1408612

[11] J. G. De Souza, M. Flavell, A. D. Suleiman, D. Shepherd, J. DeWaters, M. M. Small, Y. Liu, and D. Hou, "The importance of project-scale scaffolding for retention and experience in computing courses," in *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2023, pp. 01–09. [Online]. Available: https://doi.org/10.1109/FIE58773.2023.10343225

[12] K. Ala-Mutka, T. Uimonen, and H.-M. Jarvinen, "Supporting students in c++ programming courses with automatic program style assessment," *Journal of Information Technology Education: Research*, vol. 3, no. 1, pp. 245–262, 2004. [Online]. Available: https://learntechlib.org/p/111452/

[13] R. Hardt, "A software maintenance-focused process and supporting toolset for academic environments," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 360–370. [Online]. Available: https://doi.org/10.1109/ICSME46990.2020.00042

[14] D. C. Shepherd, F. Fronchetti, Y. Liu, D. Hou, J. DeWaters, and M. M. Small, "Project-sized scaffolding for software engineering courses," in *Proceedings of the First International Workshop on Designing and Running Project-Based Courses in Software Engineering Education*, 2022, pp. 27–31. [Online]. Available: https://doi.org/10.1145/3524487.3527362

[15] V. Gidugu, E. S. Rogers, S. Harrington, M. Maru, G. Johnson, J. Cohee, and J. Hinkel, "Individual peer support: A qualitative study of mechanisms of its effectiveness," *Community mental health journal*, vol. 51, pp. 445–452, 2015.

[16] N. K. Park, M. Y. Chun, and J. Lee, "Revisiting individual creativity assessment: Triangulation in subjective and objective assessment methods," *Creativity Research Journal*, vol. 28, no. 1, pp. 1–10, 2016. [Online]. Available: https://doi.org/10.1080/10400419.2016.1125259

[17] M. Osman, "Controlling uncertainty: a review of human behavior in complex dynamic environments." *Psychological bulletin*, vol. 136, no. 1, p. 65, 2010. [Online]. Available: https://psycnet.apa.org/doi/10.1037/a0017815

[18] A. Roy, F. Guay, and P. Valois, "Teaching to address diverse learning needs: Development and validation of a differentiated instruction scale," *International Journal of Inclusive Education*, vol. 17, no. 11, pp. 1186–1204, 2013. [Online]. Available: https://doi.org/10.1080/13603116.2012.743604

[19] P. R. Pintrich and D. H. Schunk, *Motivation in education : theory, research, and applications*. Merrill, 1996. [Online]. Available: https://cir.nii.ac.jp/crid/1130000796846016000

[20] M. M. Chemers, L.-t. Hu, and B. F. Garcia, "Academic self-efficacy and first year college student performance and adjustment." *Journal of Educational psychology*, vol. 93, no. 1, p. 55, 2001.

[21] K. K. Perkins, W. K. Adams, S. J. Pollock, N. D. Finkelstein, and C. E. Wieman, "Correlating Student Beliefs With Student Learning Using The Colorado Learning Attitudes about Science Survey," *AIP Conference Proceedings*, vol. 790, no. 1, pp. 61–64, 09 2005. [Online]. Available: https://doi.org/10.1063/1.2084701

[22] P. R. Pintrich *et al.* (1991) A manual for the use of the motivated strategies for learning questionnaire (mslq).

[23] S. E. Powers, J. DeWaters, S. J. Grimberg, D. Hou, and M. M. M. Small, “Clics–integrating data from campus sustainability projects across disciplines,” in *2015 ASEE Annual Conference & Exposition*, 2015, pp. 26–351.
[24] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.
[25] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[26] “zybooks, a wiley brand,” https://www.zybooks.com/, [Accessed 20-05-2024].
[27] G. M. Sullivan and R. Feinn, “Using effect size—or why the p value is not enough,” *Journal of graduate medical education*, vol. 4, no. 3, pp. 279–282, 2012.
[28] “Google colaboratory (colab),” https://colab.google/, [Accessed 19-05-2024].