Maximizing Individual Learning Goals through customized Student-Project Matching (SPM) in CS Capstone Projects

Jason Lee Weber
Computer Science
University of California, Irvine
Irvine, USA
weberjl@uci.edu

Barbara Martinez Neda
Computer Science
University of California, Irvine
Irvine, USA
barbarm@uci.edu

Sergio Gago-Masague
Computer Science
University of California, Irvine
Irvine, USA
sgagomas@uci.edu

Jennifer Wong-Ma
Computer Science
University of California, Irvine
Irvine, USA
jwongma@uci.edu

***Abstract*—This full innovative practice paper describes a computational tool designed to optimally match students to industry-sponsored capstone projects in a software engineering capstone course for Computer Science undergraduates at an R1 University (R1U). In the context of these capstone courses, where students stand at the culmination of their academic journey, aligning students' personal learning goals and existing computing skills with team formations becomes critical. This paper presents the Student-Project Matching Tool (SPMT), created to help students find the best available industry-sponsored projects based on their desired learning outcomes, project requirements, and their interests in each project. To choose the learning outcomes they aim to achieve, students can select from a list of predefined software engineering categories and the skills needed to achieve proficiency in each category. The initial list of technical skills for each category was recorded from job postings on a variety of well-known job-search websites, and was further refined by the capstone program's industry partners. Allowing students to select the skills they will work on ensures that they have opportunities and exposure to the skill sets required for employment while still working on one of their most appealing projects.**

We have developed and piloted the SPMT, which utilizes student vectors to represent their interests and experiences across various software engineering skill sets. Similarly, this tool uses vectors to represent the skills required by each available project, aligning with the exact dimensions as those of the student vectors. The SPMT calculated Euclidean distances between the student interest and project requirement vectors. Next, the resulting Euclidean distances were multiplied with weights associated with students' level of interest in each industry-sponsored project. Subsequently, we framed the student-project matching process as a linear sum assignment problem, aiming to minimize the total sum of Euclidean distances between each student-project pair.

The output of the SPMT process consistently matched students with teams that met their software engineering interests and project priorities. Our results reveal increased engagement and growth toward students' desired learning outcomes and computing skills. Specifically, after the first term of the capstone sequence, most students self-reported higher levels of proficiency growth in the skills within their desired software engineering category. This suggests that the SPMT effectively provides students with valuable learning experiences relevant to their career interests and representative of real-world settings.

***Index Terms*—capstone projects, computing skills, computer science**

I. Introduction

In today's rapidly evolving and competitive world, the alignment of educational experiences with industry standards and personal aspirations is not only beneficial for students, but vital. To be competitive on the job market, students must be equipped with more than theoretical knowledge. As capstone students stand at the culmination of their academic journey, their projects serve as a final bridge between theory and practice. For most, they hope to work on a project that is personally engaging and reinforces the practical skills needed to achieve their personal career goals. Moreover, their capstone project must be authentic to what they will see in a paid position to truly prepare them for similar work in industry. Therein lies the problem that every capstone program instructor or director has faced: How do we provide students with authentic and engaging projects that are relevant to each student's specific goals?

For many, the solution is to directly partner with sponsors from industry on projects. Under the guidance of these sponsors, student teams work on industry projects, many of which see real-world deployment. However, the inclusion of a third party in capstone courses adds an extra layer of complexity. Industry sponsors and their projects each come with their own unique set of requirements.

For example, a project implementing a chatbot using OpenAI's GPT-3 model will require familiarity with natural language processing and deploying AI models in cloud environments like AWS or Azure. Similarly, a project focused on developing a data visualization dashboard might require proficiency in using industry-standard tools such as Tableau, Power BI, and handling large datasets in SQL and NoSQL databases.

Additionally, since students may consider certain sponsors more desirable than others due to name recognition, there may be dissonance between project preferences and students' learning and career goals. Thus, capstone instructors have three main criteria to consider when preparing their programs: **individual learning goals**, **project preferences**, and **sponsor requirements**.

For instructors who decide to partner with industry sponsors, balancing these criteria when forming student teams becomes a difficult, but essential problem. In this paper, we present the Student-Project Matching Tool (SPMT) as a solution. The SPMT is a series of student and sponsor intake surveys that feed into the algorithmic Student-Project Matching (SPM) Recommender system that optimizes student-project assignments for individual learning goals, project preference, and sponsor requirements. The subsequent sections outline the existing algorithmic solutions to student-project matching, identify gaps in the literature that the SPMT addresses, and detail the methodology of our intake surveys, the SPM Recommender system, and our evaluation. We then present and discuss the quantitative and qualitative results from deploying the SPMT, concluding with reflections and suggestions for future work.

II. Related Work

Centering capstone programs on industry-sponsored projects has long been seen as a boon, with [1] reporting more than two decades ago that having a party that "truly cares" about the end result leads to improved student performance. Since then, many have continued research on best practices for facilitating industry participation in capstone courses. Reference [2] finds that semester-long projects with smaller team sizes work best, and [3] finds that clear and detailed project descriptions are necessary for students to make efficient use of project time, especially at the beginning of the capstone. From the sponsor perspective, [4] finds that companies see partnerships with capstone programs as an opportunity to develop low-priority projects that would not otherwise be worked on. This leads to the conclusion that, however low the stakes may be, sponsors are true stakeholders in their capstone projects. This buy-in shows in project success; In a comparison of capstone projects led by instructors and sponsors, [5] finds that industry projects lead to higher performance in skills development and general project execution than faculty projects. However, success is not solely dependent on sponsors. Reference [6] finds that student buy-in, while important to project success, is not always present, underscoring the need to match students with projects they are interested in. When asked to identify their considerations for project selection, [7] finds that the top two considerations for students are to gain experience in a particular field or technology and to receive exposure to a particular company for potential internships or employment. We then see that students must be matched to projects in a way that considers students' desired skill development, their own project preferences, and sponsor needs.

There is a long history of student-project matching in capstone courses. In the early 2000s, work from [8], [9] frames student-project matching as simple optimization problems, only considering student preferences with simple constraints like teammate preference and each team needing access to a car. In the late 2000s, we began to see some complexity added to these optimization problems, with [10]–[12] now considering lecturer preferences for projects and students. In these capstone courses, lecturers must be assigned to mentor a project and group of students. Although this marks the first inclusion of mentor preferences in the problem formulation of student-project matching, it is not directly extendable to industry sponsors. In these capstones, the lecturers are already familiar with their students and do not come prepared with projects. Industry sponsors, conversely, come prepared with projects they want to be completed and are not likely to know any students and thus will not have direct student preferences. However, recent research has begun considering sponsor preferences as side constraints to their optimization problem formations. Reference [13] moves closer to a skills-focused approach by adding a constraint that each project team must have students from at least two disciplines, while [14] adds a constraint to allow sponsors to require specific skills. Similarly, [15] suggests that each team should have sufficient "soft skills" between group members, such as interpersonal and leadership skills.

Students' project preferences are at the center of each problem formulation mentioned above. Although sponsor preferences or skills are occasionally considered additional side constraints, there is a lack of skills-focused project matching. Existing solutions do not consider each student's desired learning goals and the skills applicable to each project.

III. Research Goals

With SPMT, our goal is to directly incorporate personal learning goals, project skill requirements, and student preferences into the optimization problem instead of acting as side constraints. In this way, we aim to maximize and personalize student learning opportunities by matching students to projects that will provide the skills needed to achieve their personal goals. We also aim to create a flexible tool that can scale vertically to larger capstones and horizontally to capstones in various fields. To this end, we pose the following research questions:

1) Does the SPMT contribute to student growth toward personal learning goals?
2) Does the SPMT contribute to student engagement?
3) Does the SPMT match students with their top choices at a similar rate as manual matching?

4) Does the SPMT maintain the same level of sponsor satisfaction as with manual matching?

IV. Student Project Matching Tool

The SPMT consists of three phases (see Figure 1). The first is the data collection phase, where a series of intake surveys are given to students and sponsors for use as input to the SPM Recommender. The second is the matching phase, where the SPM Recommender matches students to projects according to a cost matrix created from the intake surveys. Finally, in the evaluation phase the quality of these matches is assessed with automatically generated metrics combined with student and sponsor feedback surveys.

A. Data Collection

The data collection phase consists of three separate surveys: two intake surveys (one for students and one for sponsors), and a project interest survey for students to rank their interest in each project. At the center of the intake surveys is a set of skills that employers consider to be the "gold standard" of skills needed to be successful in a variety of software engineering disciplines in which CS students may be interested. To establish this gold standard, we first identified six main software engineering job categories (see Table I) and searched for positions in these categories on three well-known job search websites [16]–[18]. On each site, we performed a separate search for each category and extracted the first ten job listings from each search. We then extracted a set of all the industry skills that were mentioned for each job listing.[1] Lastly, a superset of skills for each category was created by including skills that appeared in at least 70% of the job listings for a given category. The final list of skills, which we will refer to as the "gold standard", is shown in Table I.

1) Sponsor Intake Survey: The Sponsor Intake Survey consists of 38 questions aimed at understanding the sponsors and their projects. The first five questions focus on information on the project, sponsor, and sponsor mentorship availability. The purpose of these questions is to provide capstone staff with the information they need to create the Project Interest Survey detailed below. A key request during this intake survey is that sponsors record a short video detailing their project. The remaining 33 questions in the intake survey correspond directly to the gold standard for software engineering skills listed in Table I. Each sponsor is asked to select all categories under which their project falls and rate how important they are to their project on a four-point Likert scale. For each selected category, sponsors were asked to rate the relevance of the gold standard skills that fall under that category on additional four-point Likert scales.

2) Student Intake Survey: The Student Intake Survey consists of 72 questions designed to understand the students' unique backgrounds and career goals. The first 13 questions ask for personal information from students, such as courses and grades, confidence in finding a job, and demographic information. Although not directly used in the matching process, these questions provide capstone staff with insight into their class' unique backgrounds and allow for evaluation of the recommended matches during the evaluation phase. The remaining 59 questions specifically ask students about their familiarity with the gold standard software engineering skills and if they were interested in developing skills under each category. As with the Sponsor Intake Survey, familiarity with each skill is recorded on a four-point Likert scale.

3) Project Interest Survey: The last survey in the data collection phase, the Project Interest Survey, is uniquely tailored to each course offering. The sponsor intake survey responses are used to create a list of video overviews and text descriptions of sponsored projects. For each project, students are asked to review the provided videos and text and rate their interest on a three-point Likert scale (high interest, somewhat interest, no interest).

B. SPM Recommender

The SPM Recommender is the core of the SPMT, as it is the algorithmic component of the tool. First, responses from the three initial surveys form a feature vector for each student and each project, representing their unique interests and requirements. Next, a cost matrix is formed by taking the Euclidean distances between each possible pair of student and project vectors. Each value of this matrix is then scaled according to the ranks the students gave individual projects in the Project Interest Survey. In the final step, the matching problem is formulated as a linear sum assignment problem, which is subsequently solved using the Hungarian Algorithm [19], [20].

1) Generating Student and Sponsor vectors: Each element of a student vector represents the student's interest level in a specific software engineering category, as reported in the Student Intake Survey. The responses are encoded as follows: 1 for high interest, 0.5 for mild interest, and 0 for no interest.

For project vectors, a different encoding approach is used based on the Sponsor Intake Survey. Sponsors indicate the importance of each category to their project using the following ratings: major project component, significant project component, small project component, and not a project component. To avoid undervaluing skills that are less central to a project, the categories are encoded with the following values: 1 for major component, 0.8 for significant component, 0.6 for small component, and 0 for not a component. This approach ensures a more balanced and accurate representation of project requirements, preventing unintended biases that could arise from a simplistic encoding scheme.

2) Calculating and Scaling Cost Matrix: The second step of the SPM Recommender is to calculate a cost matrix with the distances between each student-project pairing. For our distance metric, we use Euclidean distance, which is calculated with the n-dimensional extension of the well-known Pythagorean Theorem. For an n-dimensional student vector $\vec{s}$ and an n-dimensional project vector $\vec{p}$, the Euclidean distance is calculated with the following equation:

[1] Note that we consider the desired familiarity with a specific tool or framework to be a skill.

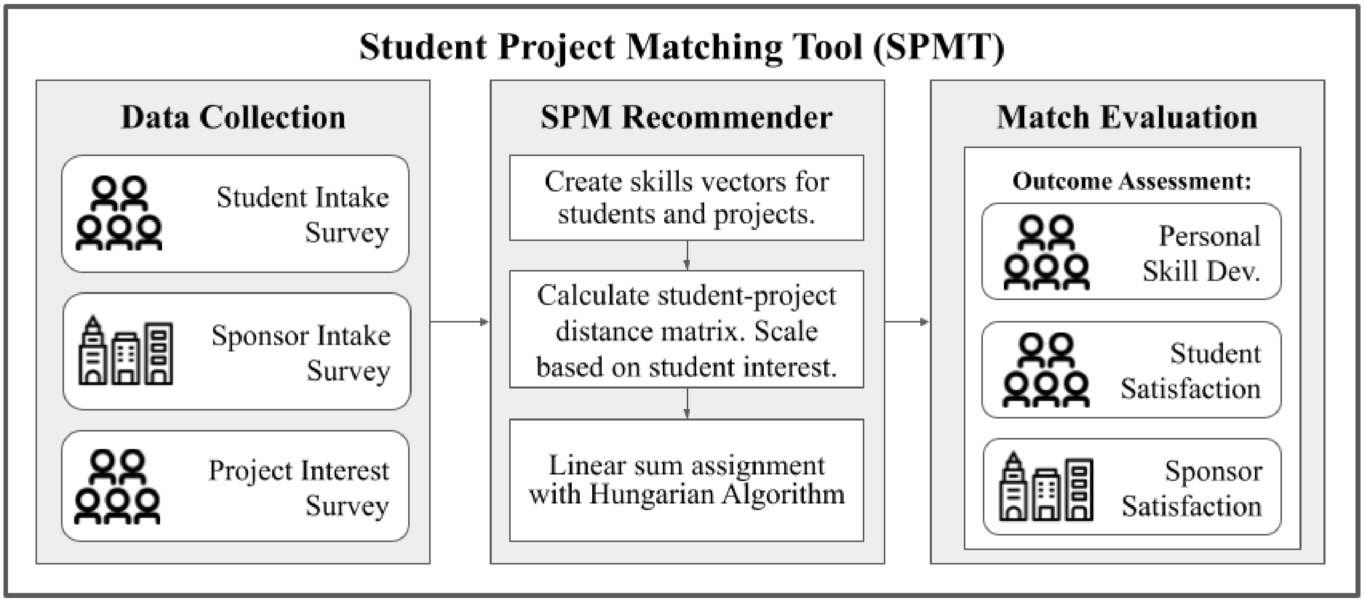

Fig. 1. Phased Workflow of the Student Project Matching Tool: Data Collection, SPM Recommender, and Match Evaluation

Front-end - Web	**Front-end - Mobile**	**Back-end**	**Databases**	**ML/AI**	**Data Science**
UI/UX	UI/UX	Java	Oracle	Python	Python
HTML	Kotlin	PHP	MySQL	OO Lang.	JavaScript
CSS	Java	Python	PostreSQL	PyTorch	R
Javascript	Swift	C#	MongoDB	TensorFlow	Tableau
React	Unix OS	Ruby		AWS	Power BI
Angular		REST APIs		Docker	Spark
Vue.js		TCP/IP			Hadoop

TABLE I
GOLD STANDARD FOR SKILLS, TOOLS AND FRAMEWORKS IN EACH JOB CATEGORY. STUDENTS RANK THEIR LEVEL OF FAMILIARITY IN THE STUDENT INTAKE SURVEY AND SPONSORS RANK THE RELEVANCE TO THEIR PROJECTS IN THE SPONSOR INTAKE SURVEY.

$$d(\vec{s},\vec{p}) = \sqrt{(\vec{s}_1 - \vec{p}_1)^2 + (\vec{s}_2 - \vec{p}_2)^2 + \ldots + (\vec{s}_n - \vec{p}_n)^2}$$

However, for our use case this can be simplified to the L2 norm of the element-wise difference between vectors:

$$d(\vec{s},\vec{p}) = ||\ \vec{s} - \vec{p}\ ||$$

Note that for a large n, the "curse of dimensionality" will take hold, reducing the effectiveness of Euclidean distance [21]. However, since we are only considering interest and requirements around the six software engineering categories in Table I, our n is relatively small.

Once distances have been calculated and compiled in a cost matrix, we scale these costs according to the Project Interest Survey responses. For each element of the cost matrix, that is, for each student-project combination, we check the student's interest in that project. If the student has high interest, we do not scale the cost. If the student has a moderate interest, we multiply the cost by 1.5. Lastly, if the student has low interest, we scale the cost by 3. In practice, these scaling values are first compiled into a matrix with the same size as our cost matrix and then multiplied element-wise with the cost matrix.

The scaling values used in our SPMT deployment were specifically chosen after iteratively tuning them for medium and low interest. We found that the mentioned values produced the largest number of top choices while minimizing low-interest matches. In future SPMT deployments, whether for a different capstone sequence or the same one, these values may need to be adjusted to achieve optimal results.

3) Linear Sum Assignment and the Hungarian Algorithm: With the cost matrix complete, the SPMT is now almost ready to be formulated as a linear sum assignment problem. That is, our matching problem can almost be described in terms of agents being assigned tasks in a way that minimizes the linear sum of the costs associated with these assignments. A key requirement of this type of problem formation is that we need a square matrix so that each agent will be assigned exactly one task, and each task will be assigned exactly one agent. To meet this requirement, we duplicate each project column in the cost matrix to provide more "slots" for students. For our specific capstone offering, this meant having at least five equivalent columns in the cost matrix for each project. In our case, where the number of projects does not divide the number of students, we first calculated the remainder r (the number of projects needing six students instead of five). We then increased the r most desirable projects to six slots. The most desirable projects were determined by a simple sum of the number of students who expressed high interest in each project.

It is also worth noting that we allowed students to identify students they would like as teammates, understanding that this would prioritize work with that teammate over their project preferences. After the SPM Recommender created its list of recommendations, we considered all bidirectional teammate requests and made swaps only if there was an available slot on the team. That is, we only moved students to a team of five (making it a team of six) to keep our group sizes relatively uniform. We also considered swapping the student with a filled slot if the other student would be moved to a project with higher priority. However, we did not need to do this in the end; in practice, very few students prioritized specific teammates over project preferences.

Now that we have a square matrix, we use Kuhn's classic Hungarian Algorithm [19] to find the optimal assignments. In our case, we specifically use *scipy*'s *linear_sum_assignment()*, which implements the Jonker-Volgenant variant of the Hungarian Algorithm detailed in [20].

4) Implementation: The SPM Recommender is implemented in the Python programming language in a Jupyter notebook for ease of use. As long as the information from the intake surveys is properly formatted and saved in the same folder as the notebook, the SPM Recommender will run successfully. The tool itself can be found on the authors' GitHub [22].

C. Match Evaluation

Once students are matched with projects, we move on to the final phase of the SPMT: evaluation. Our evaluation metrics correspond to our four research questions. First, we can immediately calculate how many students were matched with projects in which they reported high, medium, and low interest (RQ 3). Near the end of the projects, both the students and the sponsors receive a survey to report their satisfaction with their teams. Students are asked how much they have grown in each skill category corresponding to their project, cross-referenced with their interests from the Student Intake Survey (RQ 1). Students are also asked to rate their satisfaction with their project and the learning opportunities it provided them as a metric of engagement (RQ 2). Lastly, sponsors are asked to report their satisfaction with their team according to various factors, including project progress and student preparedness (RQ 4). The results of the deployment of the SPMT for a six-month capstone course are discussed in the following section.

V. Results and Discussion

We deployed the SPMT in the 2024 offering of an undergraduate software engineering capstone program at an R1 University. The sequence is two-quarters long, starting in January and ending in June. Although required to be taken back to back, one or two students usually drop between winter and spring quarters. Since only a few students left the sequence between quarters, we do not anticipate this affecting the results. To establish a baseline for SPMT performance, we compare the results with the 2022 and 2023 offerings of the capstone. In 2022, only student preferences for projects were considered, and teams were created manually. Although teams were still formed manually in 2023, student and sponsor skills were considered when matching. See Table II for a breakdown of the number of students in each term.

Year	2022	2023	2024
Winter	28	47	72
Spring	28	47	69

TABLE II
2022-2024 Enrollments per quarter of the capstone sequence

In the remainder of this section, we explore the SPMT's evaluation metrics. Specifically, we look at student-preference alignment, student feedback, and sponsor feedback.

1) Student-Preference Alignment: To evaluate the matches, we first consider the quality of the match as it relates to students' project preferences. In the intake survey, students rate projects based on their level of interest in each, expressing high, moderate, or low interest. In 2022, 86% of students were matched with a project of high interest. In 2023, these numbers drop slightly to 83%, and in 2024, they rise slightly to 88%. We establish two null hypotheses that the proportion of students matched with one of their top choices does not vary between 2024 and previous years. A two-proportion z-test between 2022 and 2024 yields a p-value of approximately 0.77. Similarly, a two-proportion z-test between 2023 and 2024 yields a p-value of 0.42. In both cases, we do not reject the null hypothesis at any reasonable significance level, suggesting that the SPMT does not significantly change the proportion of students matched to their top choices. This means that even though the SPMT factors additional considerations into its matching than was done in previous course iterations, students are still assigned to their top projects at the same rate as before.

2) Student Feedback: Student feedback is quantitatively collected in two ways. First, we look at student engagement. As a metric to garner insight into student engagement, we asked students near the end of the 2024 capstone how satisfied they were with their project and the learning opportunities it provided them. Students responded on a five-point Likert scale, ranging from "extremely dissatisfied" to "extremely satisfied". As shown in Figure 2, only a few outliers are dissatisfied with their projects and the learning opportunities they provided. In total, these outliers make up only 11% of students.

The second method we use to gather student feedback is by assessing self-identified areas for improvement. Students are asked how they think their skills have grown in each of the relevant gold standard skill categories. This is formatted as a five-point Likert scale, with values from 1 to 5 corresponding to "No Growth", "Minimal Growth", "Average Growth", "Moderate Growth", and "Major Growth", respectively. Figure 3 shows that 79% of the students list at least one category as

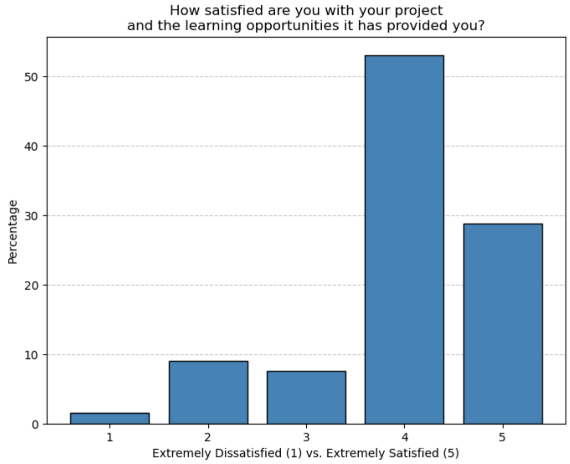

Fig. 2. Student satisfaction with their projects and the learning opportunities provided by them. Possible answers ranged from Extremely Dissatisfied (1) to Extremely Satisfied (5).

having average or higher improvement. Interestingly, when we look at the percentage of students who report an average or higher improvement in at least half of the relevant categories, we see the same proportions. That is, the same 79% of students who report growth in at least one category also report having average or higher growth in at least half of the categories relevant to their project.

Taking a closer look at the low growth students, they are spread among 57% of teams with only one or two students on each, suggesting that their low growth is not related to their project or sponsor. To verify that their low growth was not due to dissatisfaction with their project match, we looked at project assignments of low growth students and found that 77% of them were matched with their top choices. Although this is lower than the class percentage (88%), this small difference is unlikely to explain the large shift in skill growth.

3) Sponsor Feedback: Halfway through the capstone, sponsors are asked to give feedback on their teams. Specifically, they are given a survey to fill out consisting of six Likert scale questions and three text response questions. We see in Figure 4 that no sponsors are unhappy with their team's progress, regardless of year. To check the similarity of responses between 2024 and prior years, we run two non-parametric Mann-Whitney U tests. Comparing the 2022 median with the 2024 median yields a p-value of 0.881 and comparing 2023 with 2024 yields a p-value of 0.303. Both of these p-values fall below typical significance thresholds, suggesting that the SPMT has not significantly impacted sponsors' overall ratings of their teams and the progress made by the half-way point of the capstone sequence.

Although not asked in 2022, sponsors were asked in 2023 and 2024 to rate how prepared their students were for their project. In Figure 5, we see that in both years sponsors gave only neutral or positive scores to all teams. A t-test comparing the means of each year yields a p-value of 0.706.

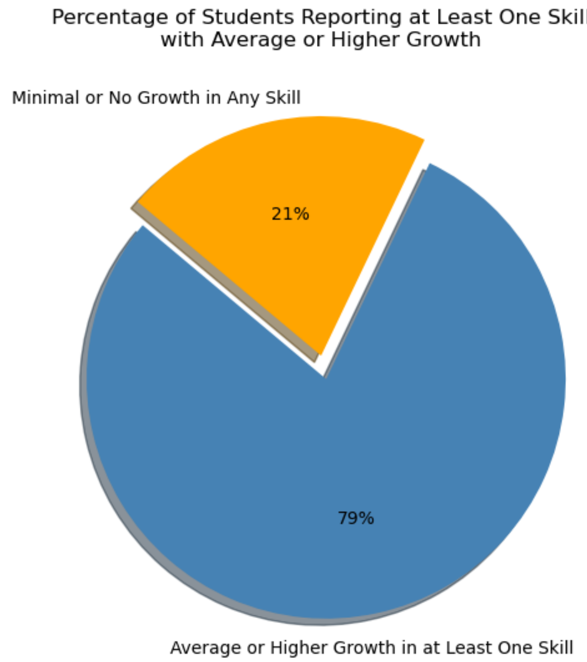

Fig. 3. Distribution of students' self-reported skill growth. The blue slice represents students who reported average, moderate, or major growth in at least one category. The orange slice represents students who did not report average or higher growth in at least one category; that is, they reported minimal or no growth in all relevant categories.

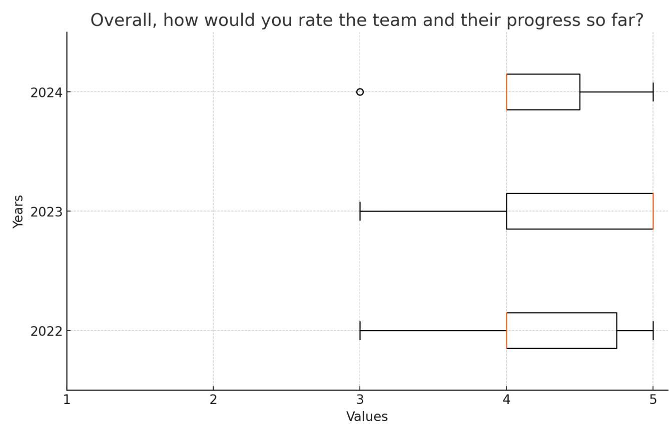

Fig. 4. Sponsors' overall satisfaction with teams and progress at the half-way point of the capstone sequence. Responses from before (2022 and 2023) and after (2024) the SPMT was deployed are included.

As before, this p-value falls below the typical significance threshold, so we conclude that the SPMT has not significantly impacted students' preparedness for their projects. Moreover, these results suggest that using the SPMT to match students to teams where they can develop the skills they want has not left sponsors with incapable students. Instead, we see that the sponsor experience remains statistically unchanged.

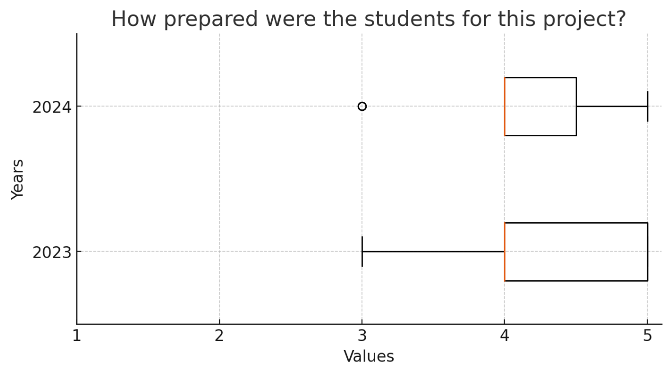

Fig. 5. Students preparedness in the 2023 and 2024 iterations of the capstone. Values reported by sponsors at the half-way point of the sequence.

VI. Conclusion

The Student-Project Matching Tool (SPMT) has proven effective in enhancing student satisfaction and promoting skill growth within the context of industry-sponsored capstone projects. A significant majority of students reported notable skill development, with 79% indicating at least average growth in skills pertinent to their projects. Importantly, the low growth observed in a small subset of students was not attributable to specific sponsors or project dissatisfaction, suggesting other factors may be at play.

The SPMT has successfully maintained a high level of alignment between student preferences and project assignments, achieving an 88% match rate with projects of high interest, comparable to the results of manual assignments from previous years. Sponsor satisfaction has also remained consistent, with positive feedback regarding project progress and student preparedness. These findings indicate that the SPMT effectively balances skill-based matching without compromising stakeholder satisfaction.

In summary, the SPMT represents a significant advancement in optimizing capstone project assignments and aligning educational experiences with student career aspirations and industry needs. Continued refinement and iteration of this tool hold promise for further enhancing capstone courses and better preparing students for their future careers.

VII. Future Work

Despite the demonstrated effectiveness of the SPMT, several areas warrant further investigation and development. Future research should aim to understand the underlying causes of low or no skill growth reported by a minority of students. Identifying whether these issues are related to project matching could inform modifications to the SPMT to better support affected students.

Additionally, the SPMT requires validation with larger and more diverse populations. While the tool was tested on a relatively large cohort within a specific computer science capstone course, it will be crucial to extend its application to capstone courses in other disciplines and institutions. This broader testing will help determine the tool's generalizability and effectiveness across different educational contexts.

Exploring the integration of more sophisticated matching algorithms and incorporating additional variables, such as student interpersonal skills and project team dynamics, could further enhance the SPMT's accuracy and utility. By addressing these areas, future iterations of the SPMT can continue to improve the alignment of student projects with individual learning goals and industry requirements.

REFERENCES

[1] K. Christensen and D. Rundus, "The capstone senior design course: an initiative in partnering with industry," in *33rd Annual Frontiers in Education, 2003. FIE 2003.*, vol. 3, Nov. 2003, pp. S2B–12, iSSN: 0190-5848. [Online]. Available: https://ieeexplore.ieee.org/document/1265941

[2] P. M. Griffin, S. O. Griffin, and D. C. Llewellyn, "The Impact of Group Size and Project Duration on Capstone Design," *Journal of Engineering Education*, vol. 93, no. 3, pp. 185–193, 2004, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2168-9830.2004.tb00805.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2168-9830.2004.tb00805.x

[3] K. S. Stresau and M. W. Steiner, "A Framework for Developing a Deeper Understanding of the Factors that Influence Success and Failure in Undergraduate Engineering Capstone Design Experiences," Jun. 2020. [Online]. Available: https://peer.asee.org/a-framework-for-developing-a-deeper-understanding-of-the-factors-that-influence-success-and-failure-in-undergraduate-engineering-capstone-design-experiences

[4] V. Rawal, S. T. O'Shields, and J. D. Summers, "Comparison of motivations and perceptions of capstone benefits for industry sponsors: An interview-based study of faculty and industry," *International Journal of Mechanical Engineering Education*, vol. 50, no. 2, pp. 269–290, Apr. 2022, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/03064190211005479

[5] M. M. Morsy, M. N. Uddin, and F. Znidi, "Multidisciplinary Capstone Senior Design Projects: A Comparative Analysis of Industry–Sponsored and Faculty–Sponsored Projects Using Comprehensive Performance Metrics," *Education Sciences*, vol. 14, no. 3, p. 314, Mar. 2024, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2227-7102/14/3/314

[6] A. Badir, R. O'Neill, K.-D. Kinzli, S. Komisar, and J.-Y. Kim, "Fostering Project-Based Learning through Industry Engagement in Capstone Design Projects," *Education Sciences*, vol. 13, no. 4, p. 361, Apr. 2023, number: 4 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2227-7102/13/4/361

[7] R. A. Hart and T. W. Polk, "An Examination of the Factors that Influence Students' Capstone Project Choices."

[8] C. W. Kirkwood, "Selecting Student Project Teams When it Really Matters: An Optimization-Based Approach Using Internet Resources," *INFORMS Transactions on Education*, vol. 4, no. 3, pp. 9–27, May 2004, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/ited.4.3.9

[9] P. R. Harper, V. de Senna, I. T. Vieira, and A. K. Shahani, "A genetic algorithm for the project assignment problem," *Computers & Operations Research*, vol. 32, no. 5, pp. 1255–1265, May 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054803003204

[10] D. J. Abraham, R. W. Irving, and D. F. Manlove, "Two algorithms for the Student-Project Allocation problem," *Journal of Discrete Algorithms*, vol. 5, no. 1, pp. 73–90, Mar. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570866706000207

[11] D. F. Manlove and G. O'Malley, "Student-Project Allocation with preferences over Projects," *Journal of Discrete Algorithms*, vol. 6, no. 4, pp. 553–560, Dec. 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570866708000476

[12] A. H. A. El-Atta and M. I. Moussa, "Student Project Allocation with Preference Lists over (Student, Project) Pairs," in *2009 Second International Conference on Computer and Electrical Engineering*, vol. 1, Dec. 2009, pp. 375–379. [Online]. Available: https://ieeexplore.ieee.org/document/5380457

[13] T. L. Magnanti and K. Natarajan, "Allocating Students to Multidisciplinary Capstone Projects Using Discrete Optimization," *Interfaces*, vol. 48, no. 3, pp. 204–216, Jun. 2018, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/inte.2017.0940

[14] N. Rezaeinia, J. C. Góez, and M. Guajardo, "Efficiency and fairness criteria in the assignment of students to projects," *Annals of Operations Research*, vol. 319, no. 2, pp. 1717–1735, Dec. 2022. [Online]. Available: https://doi.org/10.1007/s10479-021-04001-7

[15] A. Bowers, M. R. Bowers, N. Bryan, P. Letizia, and S. Murphy, "Forming Student Teams to Incorporate Soft Skills and Commonality of Schedule," *INFORMS Journal on Applied Analytics*, vol. 53, no. 2, pp. 111–127, Mar. 2023, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/inte.2022.1129

[16] "Job Search: ZipRecruiter." [Online]. Available: https://www.ziprecruiter.com/

[17] "Job Search | Indeed." [Online]. Available: https://www.indeed.com/

[18] "Jobs | LinkedIn." [Online]. Available: https://www.linkedin.com/jobs/

[19] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[20] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, Aug. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7738348/

[21] M. Koppen, "The curse of dimensionality."

[22] B. Martinez Neda, J. L. Weber, S. Gago-Masague, and J. Wong-Ma, "FIE 2024 SPMT v1.0 Release." [Online]. Available: https://github.com/Teaching-and-Learning-in-Computing/SPMT/releases/tag/v1.0