Use of Search Tools in Software Development: A Study of Microservice-Based Team Projects

Yi Meng Lau[a], Christian Michael Koh[b] and Lingxiao Jiang[c]
School of Computing & Information Systems, Singapore Management University, Singapore

Keywords: Software Development, Microservices Architecture, Search Engines, Large Language Models, Collaborative Learning, Peer Learning, Self-Directed Learning.

Abstract: Universities are increasingly integrating real-world projects into software engineering curricula to prepare students for careers involving complex concepts like Microservices Architecture (MSA). Students frequently struggle with such concepts within limited class time and turn to various search tools and online resources for additional help. Search tools are also widely used in the software development industry. While search engines, like Google and Yahoo!, can provide quick solutions, they pose the risk of information overload. Large Language Models (LLMs) such as ChatGPT, offer the advantage of delivering more precise answers. Studies have shown that LLMs can comprehend codes, assist in system architectural design, and suggest solutions, potentially enhancing the learning experience for students. This study aims to determine how students make use of search tools for their team projects in a software development course that teaches MSA. It will also explore if search tools can enhance learning in team projects by facilitating collaborative, peer, and self-directed learning, and propose methods to address any limitations.

1 INTRODUCTION

Learning in software engineering has extended beyond attending instructor-delivered lessons focusing on knowledge acquisition and computational thinking. Universities are incorporating real-world practical projects into their software engineering and STEM courses to align theoretical knowledge with workplace demands, integrating technical and soft skills (Ceh-Varela, Canto-Bonilla, & Duni, 2023; Guo, Saab, Post, & Admiraal, 2020). These projects provide students with the opportunity to explore technical concepts, working individually and in teams to prepare them for their future careers (Podeschi, 2016; Ceh-Varela, Canto-Bonilla, & Duni, 2023). For example, many universities have included Microservices Architecture (MSA) (Bogner, Fritzsch, Wagner, & Zimmermann, 2021) in their software engineering curricula (Lau, Koh, & Jiang, 2024). MSA involves breaking down the functions of an application into small, self-contained code modules called microservices that are highly reusable for agile software development (Bucchiarone, Dragoni, Dustdar, Larsen, & Mazzara, 2018). These concepts can be abstract and hard for students to understand within limited class time.

Studies have shown that software developers and students extensively leverage online resources to learn new programming languages and technical concepts, clear doubts, and augment their existing knowledge (Brandt, Guo, Lewenstein, Dontcheva, & Klemmer, 2009; Liu, et al., 2021). Search engines such as Google and Yahoo! are widely used by industry practitioners to source and locate information (Rangaswamy, Giles, & Seres, 2009). Likewise, online resources are valuable information sources for students working on software projects, especially when they are tackling new or unfamiliar topics (Griffiths & Brophy, 2005; Walraven, Brand-Gruwel, & Boshuizen, 2009). The abundant online resources available in software development cater to students seeking to learn programming at both novice and advanced levels (Lu & Hsiao, 2017).

While search engines have been the primary tools for students to quickly and conveniently start their

[a] https://orcid.org/0009-0000-7470-2556
[b] https://orcid.org/0009-0004-1149-7478
[c] https://orcid.org/0000-0002-4336-8548

Lau, Y. M., Koh, C. M. and Jiang, L.
Use of Search Tools in Software Development: A Study of Microservice-Based Team Projects.
DOI: 10.5220/0013206000003932
In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 232-242
ISBN: 978-989-758-746-7; ISSN: 2184-5026

information searches (Liu, Zamir, Li, & Hastings, 2018), search results can be overwhelming or unfavorable. Students face challenges of information overload, uncertainty about which keywords to use, and the need to spend more time reading and interpreting the results (Griffiths & Brophy, 2005; Hahnel, Goldhammer, Kröhne, & Naumann, 2018; Mahdi, Ahmad, Ismail, Natiq, & Mohammed, 2020).

This has evolved with the introduction of Large Language Models (LLMs), which enable more conversational queries with more precise answers. LLMs can enhance productivity in software development through features like auto-completion, code suggestions, and interactive chat dialogues (Valový & Buchalcevova, 2023). Software engineering graduates will increasingly find themselves in LLM-driven environments, suggesting a need for software engineering education to adapt and innovate to accept the use of LLMs (Kirova, Ku, Laracy, & Marlowe, 2024).

Given the complex nature of software development, students are likely to depend on search engines or LLMs (collectively called "search tools" in this paper) for their courses. Rather than viewing search tools as merely a tool, the search process can also be part of learning (Brandt, Guo, Lewenstein, Dontcheva, & Klemmer, 2009; Lu & Hsiao, 2017). Online searches seem like an individual activity (Ghosh, Rath, & Shah, 2018).

This study aims to determine how students use search tools for their team projects in a software development course that teaches MSA, where students need to learn and practice breaking down a monolithic application into small, self-contained, reusable microservices, developing the solution as an assembly of microservices, and deploying the tested solution in containerized environments. We will also evaluate if search tools can be integrated as learning resources to enhance the student's learning experience and facilitate collaborative, peer, and self-directed learning through a proposed framework.

Since teamwork is essential for software developers to collaboratively analyze problems from multiple perspectives (Shuto, et al., 2016), we will also examine the limitations of search engines and LLMs in facilitating collaborative efforts (Raibulet & Fontana, 2018) and suggest methods to address these challenges.

We gathered feedback through an online survey from students enrolled in a software development course. With the insights from the survey, we developed a web-based prototype to address these gaps. The prototype is piloted with a smaller group of students.

The study seeks to address the following research questions:

- RQ1: Are students using search engines or LLMs in their team projects?
- RQ2: Can search engines enhance learning in team projects by facilitating collaborative, peer, and self-directed learning?
- RQ3: Can LLMs enhance learning in team projects by facilitating collaborative, peer, and self-directed learning?
- RQ4: If there is an application that can integrate the benefits of various search tools, what additional features would students like to have?

The rest of this paper is organized as follows: Section 2 shares related works. Section 3 describes the proposed framework. Section 4 presents the approach and findings of this study. Section 5 discusses the limitations of this study that present potential areas for future work. Section 6 concludes.

2 RELATED WORKS

2.1 Software Development Projects

The teaching of software engineering courses is complex as they involve a wide range of concepts, algorithms, software design principles, and tools (Ceh-Varela, Canto-Bonilla, & Duni, 2023). Software development courses tend to emphasize technical skills at the expense of soft skills (Tubino, Morgan, Wood-Bradley, & Cain, 2023).

Industries have cited soft skills as essential work-ready skills and found that novice software developers are reportedly disadvantaged by their lack of soft skills (Christensen & Paasivaara, 2022). Software developers are expected to collaborate effectively in teams to solve real-world problems (Kuusinen & Albertsen, 2019) prompting universities to introduce team-based projects to address this need (Hamer, Quesada-López, Martínez, & Jenkins, 2021; Christensen & Paasivaara, 2022). Team projects have proven effective in improving technical proficiency and soft skills and encouraging students to take responsibility for their learning (Ceh-Varela, Canto-Bonilla, & Duni, 2023). To encourage collaboration and teamwork, universities have explored the use of industry tools like Microsoft Project, GitHub, and Trello in software development courses team projects (Raibulet & Fontana, 2018; Macak, Kruzelova, Chren, & Buhnova, 2021; Christensen & Paasivaara, 2022).

A challenge instructors often face in team projects is to assess individual student contributions versus team contributions and to establish effective feedback mechanisms to track students' project development and facilitate timely interventions (Hamer, Quesada-López, Martínez, & Jenkins, 2021). Git logs have been used and analyzed to track team progress and understand the team dynamics (Hamer, Quesada-López, Martínez, & Jenkins, 2021; Macak, Kruzelova, Chren, & Buhnova, 2021).

2.2 Searching as Part of Learning

Ghosh et al. share that learning is a significant outcome of searching (Ghosh, Rath, & Shah, 2018). As we seek information to address specific problems, we often explore related areas (Rieh, Collins-Thompson, Hansen, & Lee, 2016). The consideration of keyword choices and relevance assessments are cognitive processes akin to thinking and learning (Vakkari, 2016).

Online searches are a three-stage search process of formulating the query, selecting sources, and interacting with them (Vakkari, 2016) exhibits parallels between search behavior and learning behavior suggesting that it is critical learning rather than passive reception (Rieh, Collins-Thompson, Hansen, & Lee, 2016). These search activities allow individuals to integrate pieces of information, create new insights, evaluate, and apply knowledge, and reconstruct their understanding of the search topic (Vakkari, 2016; Ghosh, Rath, & Shah, 2018). At the same time, students learn to recognize and appreciate diverse perspectives and viewpoints (Rieh, Collins-Thompson, Hansen, & Lee, 2016).

2.3 Use of Search Engines in Software Development

Search engines have become a key source of information for industry practitioners, including those in software development (Rangaswamy, Giles, & Seres, 2009; Sadowski, Stolee, & Elbaum, 2015). Developers do code searches to increase productivity and quality. It is observed that 74% of the time these searches are queried via natural language, although recent advancements in machine learning aim to improve search accuracy (Liu, et al., 2021). Brandt et al. share that developers often concurrently engage in web-based research, learning, and code writing, adopting a just-in-time approach to learn new skills or refresh their memory on existing context or syntax (Brandt, Guo, Lewenstein, Dontcheva, & Klemmer, 2009).

With the sheer amount of information available online, code searching is not limited to finding reusable codes but also understanding what a particular piece of code is doing, or fixing a bug (Sadowski, Stolee, & Elbaum, 2015; Grazia & Pradel, 2023). Search engines have the challenge of having a vast amount of available information, complicating the task of locating relevant data (Mahdi, Ahmad, Ismail, Natiq, & Mohammed, 2020). Researchers have investigated various methods to mitigate information overload with algorithms, systems, filtering tools, and custom querying languages (Mahdi, Ahmad, Ismail, Natiq, & Mohammed, 2020; Grazia & Pradel, 2023).

2.4 Use of Large Language Models in Software Development

Large Language Models (LLMs) have transformed software development by assuming traditional functions previously completed on search engines. The study by Nam et al suggests that LLMs aid in code comprehension and suggest codes and API endpoints during application development (Nam, Macvean, Hellendoorn, Vasilescu, & Myers, 2024). Researchers have attempted to use LLMs to generate code explanations and compile them into an e-book for web software development to determine whether these explanations generated by LLMs are useful for learning (MacNeil, et al., 2023). For more advanced topics such as software architecture designs, bots are developed using ChatGPT to help novice software architects gain experience through artificial intelligence decision support to provide rapid architecting software systems (Ahmad, et al., 2023).

LLMs like ChatGPT have significantly impact STEM education. Beyond serving as an information repository, ChatGPT functions as a practice question bank, revision tool, code debugger for students, or even as an educator lesson plan assistant (Banerjee, Srivastava, Adjeroh, Reddy, & Karimian, 2023). By adapting to changes in question structure, it is a valuable tool for both students and instructors rather than a threat to the education system (Banerjee, Srivastava, Adjeroh, Reddy, & Karimian, 2023).

While students find that these tools contribute to their learning and have become an integral part of coding, they agree that fundamental concepts in software engineering should be learned without relying on AI technologies (Valový & Buchalcevova, 2023; MacNeil, et al., 2023).

3 PROPOSED FRAMEWORK TO EVALUATE SEARCH TOOLS

Students use search engines or LLMs to support their learning (Rieh, Collins-Thompson, Hansen, & Lee, 2016). However, the effectiveness of these search tools in supporting teamwork, particularly in collaborative, peer, and self-directed learning among students working on microservice-based team projects, has not been studied. Figure 1 presents the proposed framework, along with the team project schedule. Search tools can be used throughout the team project phases.

Collaborative learning happens when students work together to solve a problem with shared responsibilities (Laal & Laal, 2012; Loes, 2022). Through this interaction, students teach and learn from each other, leading to ***peer learning*** (Topping, 2005; Lane, 2016). This encourages ***self-directed learning*** as students take initiative, identify resources, and develop strategies to contribute effectively to the group (Topping, 2005). Collaborative and peer learning enhance teamwork and soft skills that students can develop while working together (Lane, 2016), essential for the software development industry (Kuusinen & Albertsen, 2019; Christensen & Paasivaara, 2022; Ceh-Varela, Canto-Bonilla, & Duni, 2023).

In ***collaborative learning***, we examine if search tools can facilitate the following in students: (1) knowledge and ideas exchange in "Group Activity", which helps to strengthen their comprehension of the subject matter and hone their (2) "Communication" skills by sharing perspectives and articulating thoughts as they work toward a (3) "Common Goal". They are encouraged to (4) "Provide Feedback" as part of (5) "Active Participation" in the learning process, contributing their viewpoints and insights, rather than passively receiving information (Laal & Laal, 2012; Loes, 2022).

In ***peer learning***, we examine if students can (1) "Gain New Perspectives" by reviewing problems and exchanging information based on their peers' findings. This mirrors the process of learning through searches, where students learn to appreciate diverse perspectives and develop flexible thinking skills and the ability to construct reasoned conclusions (Rieh, Collins-Thompson, Hansen, & Lee, 2016). This can potentially (2) "Accelerate their Learning" process instead of searching for data individually. Students will benefit from (3) "Giving and Receiving Feedback" of search results with their peers, enabling them to be (4) "Actively Engaged" in the learning process (Topping, 2005).

In ***self-directed learning***, students take ownership of their learning (Loyens, Magda, & Rikers, 2008) which is a necessary skill for a software developer. The concept of information searching, querying, and evaluating search results aligns with self-directed learning (Loyens, Magda, & Rikers, 2008; Vakkari, 2016). Students need to be (1) "Ready to Learn" and aware of their "Learning Needs" to know what and how to query. To reach their (3) "Individual Goals", students display (4) "Active Involvement" in understanding, analyzing, and evaluating search results (Rieh, Collins-Thompson, Hansen, & Lee, 2016), creating new knowledge that can contribute to their team.

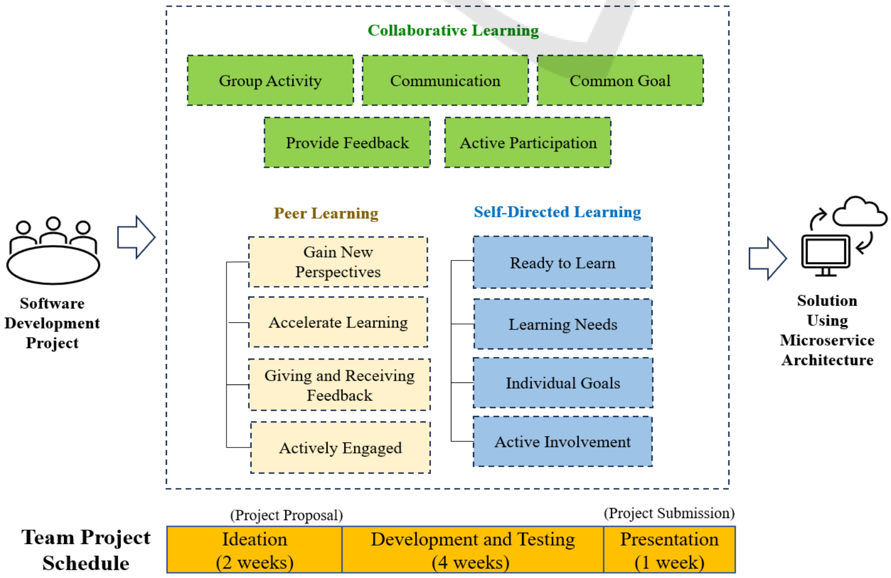

Figure 1: Proposed Framework to Evaluate Search Tools for Collaborative, Peer, and Self-Directed Learning.

4 STUDY APPROACH AND FINDINGS

The pilot study was conducted with Information Systems students enrolled in the Year 2 undergraduate Enterprise Solutions Development course. At the end of their team project, students complete a voluntary survey to provide their feedback. The response rate is 59.1% (71 out of 120 students).

With insights from this survey, we identified the gaps in search tools and developed a prototype to address these gaps. Eight students separated into two groups were involved to test out the prototype in a 4-hour face-to-face session. Students completed a short survey after that which helped us verify our understanding of the identified challenges.

Both surveys included Likert scale questions where 1 represents the least favorable response and 10 represents the most favorable, open-ended, and multi-select questions. Both surveys are endorsed by the university's Institutional Review Board. Numbers in brackets in subsequent sections represent the average Likert scores for the questions.

4.1 Enterprise Solutions Development Course Team Project

The Enterprise Solutions Development (ESD) course explores how enterprises adapt applications to meet evolving organizational and customer needs. While monolithic applications were traditionally favored for their simplicity, the need for faster releases has led organizations to adopt microservices architecture.

	Team Project Schedule	Course Content
Week 1		Introduction
Week 2-6		Lectures + Lab Exercises
Week 7	Ideation	
Week 8		Midterm break
Week 9	(Project Proposal)	Consultation
Week 10-12	Development and Testing	Lectures + Case Studies
Week 13	Presentation (Project Submission)	
Week 16		Examination

Figure 2: ESD Course Team Project Schedule.

In this course, students design and implement MSA using tools like Flask and Flask-SQLAlchemy for Python-based microservices and databases. Key topics include communication patterns (e.g., HTTP, AMQP), data transformation (e.g., JSON, XML), web interfaces (e.g., HTML, CSS, JavaScript), Docker deployment, and API Gateway management (e.g., Kong). The course is assessed by class participation, quizzes, a team project and an examination The course schedule is provided in Figure 2.

The team project requires students to work in groups of 5-6 to identify a real-world problem and build a solution using concepts and technologies taught in the course. Their solution must include at least three user scenarios, integrating both complex and simple microservices with independent data stores. It should involve third-party service calls, establish HTTP or AMQP communication between microservices, and feature a web-based user interface. The microservices must be containerized and may be managed via Kubernetes. Project teams must incorporate at least one technology not covered in the course, such as using a new programming language or a database type. Students are graded as a team and individually based on their contributions.

Due to limited class time, although all the technical concepts needed for the project solutions were introduced in class, they were not covered in detail. Students were encouraged to read additional self-study materials beyond class time and use various search tools as they like.

4.2 Use of Search Engines or LLMs

Survey results showed that 67 (94.4%) students use LLMs or search engines for their team projects. Four students (5.6%) indicated they do not use either, as they prefer to get help from their peers and reason that they do not trust LLMs' coding abilities.

In their use of search engines, students indicated that the results from search engines often provide a broad perspective of information (7.328) (see SE1 in Figure 3) and require them to discern applicable information (7.657) (see SE2 in Figure 3). They need precise keywords and multiple queries (8.149) (see SE3 in Figure 3) to find specific information. Some students have difficulty remembering what was previously queried (6.881) (see SE4 in Figure 3).

Students prefer LLMs' search results over those of search engines (7.925) (see L1 in Figure 4). On some occasions, LLMs provide a generalized perspective (8.164) (see L2 in Figure 4) and require well-crafted prompts to get the desired results (8.746) (see L3 in Figure 4), but their natural language capability simplifies prompt creation (8.284) (see L4 in Figure 4). LLMs facilitate easier revisiting of prior search results (8.358) (see L5 in Figure 4).

For their project requirements, students prefer the results from LLMs, as they can identify different personas and how these personas will interact with the application they intend to build (7.821). Students can leverage the results of LLMs to identify required microservices for their project (7.776). The code structures offered by LLMs provide a good starting point for their solutions (7.970) and accelerate their development process (7.806).

Answer to RQ1: The survey data shows that students use search tools in their projects with a higher preference for LLMs.

4.3 Search Engines for Collaborative, Peer and Self-Directed Learning

Next, we evaluate the effectiveness of search engines for students in facilitating learning in their team projects with the proposed framework in Figure 1.

The team project itself has created a "Common Goal", which is to be completed collaboratively, with each student contributing their share before integrating these contributions into a final solution. This is their "Group Activity".

While the search engine is not a tool to create a "Common Goal", students use search engines to find relevant information and resources to help them solve problems. From the response to the open-ended question in the survey, students mentioned that the first "Group Activity" is for all team members to search for ideas for their team project.

The broad perspective of information (7.328) (see SE1 in Figure 3) from search results serves as a valuable starting point to facilitate knowledge and idea exchange within the team. However, most students will conduct search processes individually (7.394) (see SE6 in Figure 3), making sense of the data individually before using external tools such as Google Docs to share with teammates (7.239) (see SE5 in Figure 3) which are usually copying and pasting of search results without much context, with little "Communication" among the team. "Active Participation" primarily occurs through their contributions in Google Docs and verifying the search results of their teammates (7.881) (see SE7 in Figure 3) to "Provide Feedback".

In peer learning, sharing search results allows team members to "Gain New Perspectives" from the findings and can potentially "Accelerate Learning". If one teammate is unable to find information through their search, another may have the answer. "Giving and Receiving Feedback" can be achieved by verifying each other's search results via external tools (7.881) (see SE7 in Figure 3). Evidence of being actively engaged can be seen through the history of shared document tools.

Students must contribute to their team. Their "Individual Goals" are to complete their part of the project before integrating it with others. Their "Learning Needs" are met when they identify the resources necessary to complete their tasks. By independently assessing the validity of search results (7.657) (see SE2 in Figure 3), creating precise keywords (8.149) (see SE3 in Figure 3), and overcoming challenges such as retracing previous searches (6.881) (see SE4 in Figure 3), they demonstrate "Active Involvement". Their openness to feedback indicates their "Readiness to Learn".

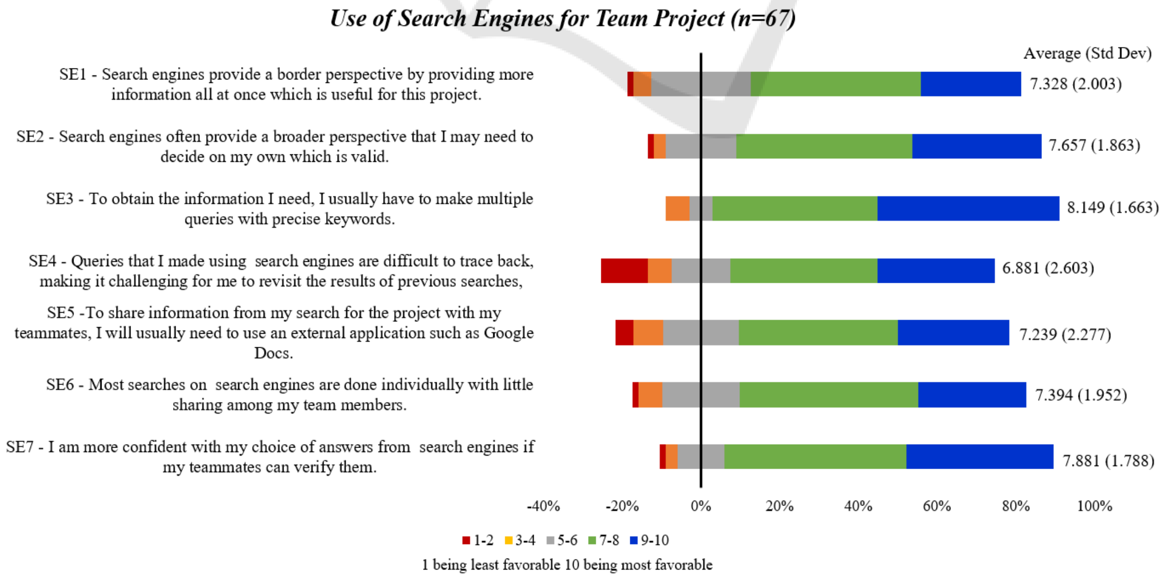

Figure 3: Results from the Survey on the Use of Search Engines.

Students noted that search engines often fail to resolve bugs or explain issues, requiring them to adapt code to fit project specifics, which highlights another aspect of self-directed learning.

Answer to RQ2: Search engines often lack an effective means for team members to communicate easily, provide feedback to one another, and share information. Tracing and sharing of search results can be challenging. However, search engines do enhance learning in other areas.

4.4 Large Language Models for Collaborative, Peer, and Self-Directed Learning

The effectiveness of LLMs in facilitating students' learning in their team projects is also evaluated using the proposed framework in Figure 1.

Like search engines, the team project serves as a "Common Goal" and creates opportunities for "Group Activity". The scenarios generated by LLMs provide a good starting point for project teams to define their user scenarios and the microservices needed to develop solutions (7.493) (see L9 in Figure 4), as students work toward their "Common Goal". Although search results can be shared through the URLs provided by LLMs, students are using external tools to "Provide Feedback" to their team members, which they value (8.106) (see L8 in Figure 4). One concern that students have is that the URLs will not contain information from future searches that continue later (6.896) (see L6 in Figure 4). While LLMs do facilitate some forms of "Communication" through sharing, "Active Participation" still relies heavily on contributions to shared documents, although team members can be actively searching for information individually via the LLMs.

Peer learning supported by LLMs resembles search engines but enhances the sharing of search results by providing documented steps on how each result is derived, making the process more accessible.

LLMs can facilitate self-directed learning. With a clear "Individual Goal" it defines the "Learning Needs". Students use LLMs to find information to solve their problems (7.925) (see L1 in Figure 4). By crafting well-structured prompts (8.746) (see L3 in Figure 4) in natural language (8.248) (see L4 in Figure 4) and assessing the validity of LLMs' responses (8.164) (see L2 in Figure 4), students become "Actively Involved" in their learning. LLMs also facilitate easy retrieval of past searches (8.358) (see L5 in Figure 4), helping students plan their project's next steps efficiently. They appreciate teammate validation of their search results (8.106) (see L8 in Figure 4) indicating their "Readiness to Learn".

Answer to RQ3: The survey results show that LLMs contribute to self-directed learning with their natural language prompts with better results. Although LLMs provide an easier method for sharing and tracking back search results, they lack the ability for team members to communicate and provide feedback to one another.

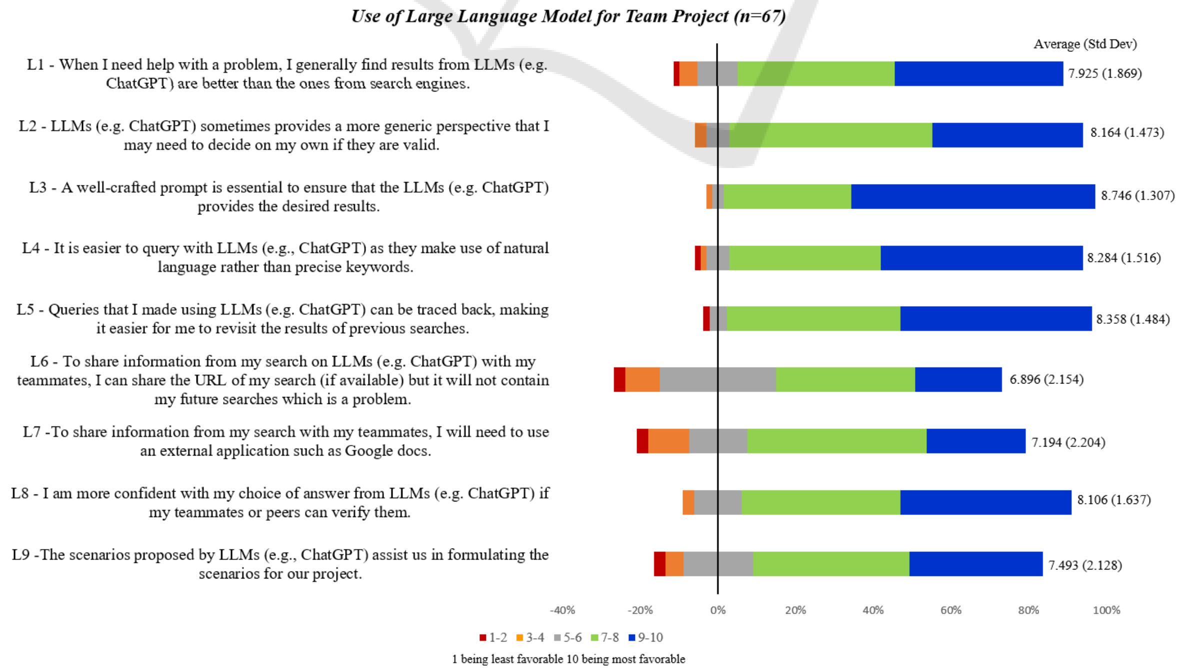

Figure 4: Results from the Survey on the Use of Large Language Models.

4.5 Additional Features that Can Complement Search Tools

The survey asked students if an application that integrates the benefits of search tools would assist them in their team project and what additional features they would like to have.

Students expressed the need for a way to integrate and share search results. Search results like the initial definition of business scenarios related to the problem, along with the structure of API endpoints and code architecture, were highlighted as particularly useful (7.985). Additionally, students noted that providing a way to offer timely feedback in a consolidated manner (7.803) was particularly useful. Given the complexity of MSA solutions, which require extensive teamwork, such features would be highly beneficial.

The ability to keep track of search queries throughout the development process (7.923) would allow teams to easily refer to previous search results, providing valuable resources for fixing bugs or finding solutions. A means of creating documentation alongside the search results would also be useful (8.721).

Answer to RQ4: Students have requested an easier way to integrate their search results and share them with the team. They also want the application to provide timely feedback, allow them to track all queries as they build the application, and assist with documentation throughout the development process.

4.6 Addressing the Limitations of Search Tools with Prototype

The survey results revealed that search tools lacked features to effectively facilitate "Communication", "Providing Feedback" and "Active Participation" within the team. Together with the suggested additional features, we developed a prototype to address these limitations.

The prototype is developed as a web application hosted on Amazon Web Services (AWS) which is query ChatGPT-3.5 via available APIs. ChatGPT was chosen as the search tool because it has available APIs and shares similarities with the results from search engines and LLMs.

The prototype allows the team leader to define the team's problem statement ("Common Goal") at the outset. At any stage, students can seek feedback or assistance from their team members ("Timely Feedback on") and contribute to the documentation of the application they are developing. Search results are tracked and made available to the team, promoting transparency ("Shared Search Results"). This approach fosters "Active Participation" within the project team. Figure 5 shares the screenshots of the prototype.

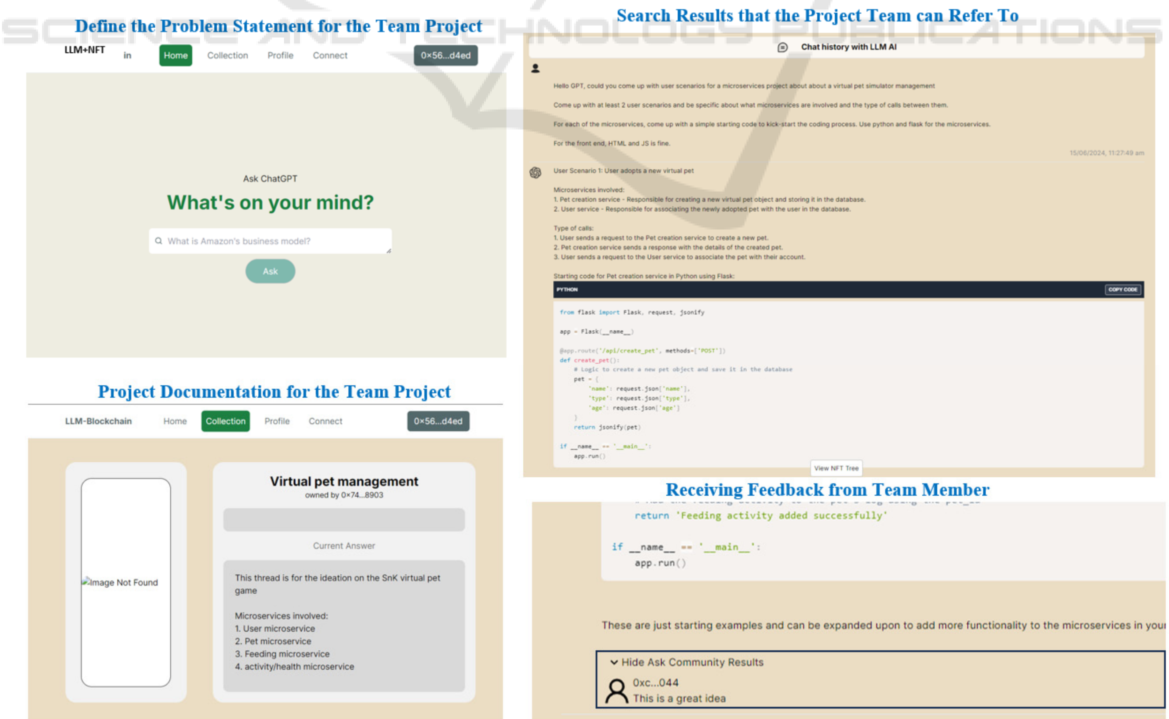

Figure 5: Sample Screenshot of the Prototype.

4.6.1 Evaluating the Prototype

A small group of 8 students from the same course was engaged to test the prototype by building a scaled-down version of their team project. They provided feedback on the prototype through a short survey. The survey results are presented in Table 1.

Table 1: Survey Results for the Prototype.

Survey Questions (n=8)	Average (Std Dev)
The application allows my team to share search results (e.g., business scenarios, API endpoints, code structure, etc.) and is useful for us.	8.875 (0.927)
The application that keeps track of our search queries as we build our solutions would be useful for us.	8.125 (1.763)
The application allows me or members of my team to revisit the results of previous searches which is useful for us.	8.375 (1.798)
The application that enables my teammates or peers to provide timely feedback to each other is useful for us.	8.125 (1.452)
The application can assist us in building our documentation (e.g., project specifications, API endpoints, microservices design) as we develop our solutions would be useful for us.	7.625 (1.495)

Students valued the collaborative features in the prototype, particularly the ability to share search results (8.875), track search queries (8.125), and revisit previous search results (8.375). Students also found the tool beneficial for facilitating timely feedback among peers when they needed clarification or assistance (8.125). However, the scores were slightly lower for the prototype's ability to help them with documentation, likely due to the primitive functionality of accepting only text data in the application.

Overall, the feedback highlights the application's effectiveness in enhancing collaboration and documentation during the project development process.

5 LIMITATIONS AND FUTURE WORKS

The study focused on a single software development course, potentially limiting its scope. Similar studies can be conducted with other software development courses, with more participants, to verify the effectiveness of the proposed framework and that search tools are indeed helpful for students learning software development (Sadowski, Stolee, & Elbaum, 2015; Brandt, Guo, Lewenstein, Dontcheva, & Klemmer, 2009; Nam, Macvean, Hellendoorn, Vasilescu, & Myers, 2024).

Likewise, the study can be extended to include other types of search tools and the development of effective prompts for LLM queries.

Feedback received was collected via a voluntary online survey, reflecting student perceptions. Factors like search patterns, accuracy, and prompt usage were not evaluated. These could provide valuable insights into students' understanding and interests in particular technical topics and if the use of search tools spurs interest in software development.

The prototype that is developed has a very basic user interface and features. Enhancements could involve integrating with search engines like Google or other LLMs, not limiting to ChatGPT. Team project often use multiple forms of communication such as Telegram or Zoom that can be integrated.

Another area worth exploring could involve analyzing search trends by the students. These trends could indicate when students are searching aimlessly or are uncertain about their next steps, potentially signaling a need for assistance (Hamer, Quesada-López, Martínez, & Jenkins, 2021) .

The issue of individual and team contributions is consistently a concern for instructors (Macak, Kruzelova, Chren, & Buhnova, 2021). The individuals' contributions to the project can be used to evaluate the effectiveness of collaborative, peer, and self-directed learning in the project. The search results and project team documentation can serve as references for future similar projects, allowing them to build upon existing work rather than starting from scratch.

6 CONCLUSION

Software development courses cover a wide range of topics and content to prepare students for industrial workplace demands. The complexity of MSA and various technical solutions add challenges to teaching and learning the topics and can make it difficult for students to apply these concepts.

Software developers and students are using search tools to assist them with productivity, solve bugs, and learn new languages and techniques. Rather than being viewed as solely a tool used to find answers, search tools can be integrated as part of the learning process.

This study examines the use of search tools in team projects within a software application development course involving MSA, focusing on how these tools can facilitate collaborative, peer, and self-directed learning. The study first investigates how students are using search engines and Large Language Models (LLMs). With a proposed framework, it evaluates the potential gaps in the effectiveness of these tools.

It was found that both search engines and LLMs have limitations in supporting teamwork effectively, primarily due to the lack of features for sharing information, communicating among team members, and providing feedback. Since teamwork is an essential part of software development, alongside the required technical skills, addressing these challenges is essential.

Based on student feedback, a prototype was built and tested with a smaller group of students to verify its effectiveness and understanding. The prototype's features—facilitating information sharing, problem-solving, and team feedback—could further enhance the effectiveness of search tools to better facilitate and support collaborative, peer, and self-directed learning in software development courses.

ACKNOWLEDGMENTS

The authors would like to express gratitude to the students and instructors involved in this course for making this study possible.

REFERENCES

Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Aktar, M. S., & Mikkonen, T. (2023). Towards Human-Bot Collaborative Software Architecting with ChatGPT. *27th International Conference on Evaluation and Assessment in Software Engineering*, (pp. 279 - 285). Oulu, Finland.

Banerjee, P., Srivastava, A., Adjeroh, D., Reddy, Y. R., & Karimian, N. (2023). Understanding ChatGPT: Impact Analysis and Path Forward for Teaching Computer Science and Engineering. *TechRxiv*.

Bogner, J., Fritzsch, J., Wagner, S., & Zimmermann, A. (2021). Industry practices and challenges for the evolvability assurance of microservices: An interview study and systematic grey literature review. *Empirical Software Engineering, 26*(104).

Brandt, J., Guo, P. J., Lewenstein, J., Dontcheva, M., & Klemmer, S. R. (2009). Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code. *SIGCHI Conference on Human Factors in Computing Systems*, (pp. 1589-1598). Boston, USA.

Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S. T., & Mazzara, M. (2018). From monolithic to microservices: An experience report from the banking domain. *IEEE Software, 35*(3), 50-55.

Ceh-Varela, E., Canto-Bonilla, C., & Duni, D. (2023). Application of Project-Based Learning to a Software Engineering course in a hybrid class environment. *Information and Software Technology, 158*, 107189.

Christensen, E. L., & Paasivaara, M. (2022). Learning Soft Skills through Distributed Software Development. *International Conference on Software and System Processes and International Conference on Global Software Engineering*, (pp. 93-103). Pittsburgh, PA, USA.

Ghosh, S., Rath, M., & Shah, C. (2018). Searching as Learning: Exploring Search Behavior and Learning Outcomes in Learning-Related Tasks. *2018 Conference on Human Information Interaction & Retrieval*, (pp. 22-31). New Brunswick, NJ.

Grazia, L. D., & Pradel, M. (2023). Code Search: A Survey of Techniques for Finding Code. *ACM Computing Surveys, 55*(11), 1-31.

Griffiths, J. R., & Brophy, P. (2005). Student searching behavior and the web: use of academic resources and Google. In *library trends.*

Guo, P., Saab, N., Post, L. S., & Admiraal, W. (2020). A review of project-based learning in higher education: Student outcomes and measures. *International Journal of Educational Research, 102*, 101586.

Hahnel, C., Goldhammer, F., Kröhne, U., & Naumann, J. (2018). The role of reading skills in the evaluation of online information gathered from search engine environments. *Computers in Human Behavior, 78*, 223-234.

Hamer, S., Quesada-López, C., Martínez, A., & Jenkins, M. (2021). Using git metrics to measure students' and teams' code contributions in software development projects. *CLEI Electronic Journal , 24*(2).

Kirova, V. D., Ku, C. S., Laracy, J. R., & Marlowe, T. J. (2024). Software Engineering Education Must Adapt and Evolve for an LLM Environment. *55th ACM Technical Symposium on Computer Science Education V. 1, 1*, pp. 666 - 672. Portland, USA.

Kuusinen, K., & Albertsen, S. (2019). Industry-Academy Collaboration in Teaching DevOps and Continuous Delivery to Software Engineering Students: Towards Improved Industrial Relevance in Higher Education. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET).* Montreal, QC, Canada.

Laal, M., & Laal, M. (2012). Collaborative learning: what is it? *Procedia - Social and Behavioral Sciences, 31*, 491-495.

Lane, S. (2016). Promoting collaborative learning among students. *American Journal of Educational Research, 4*(8), 602-607.

Lau, Y. M., Koh, C. M., & Jiang, L. (2024). Teaching Software Development for Real-World Problems using a Microservice-Based Collaborative Problem-Solving Approach. *IEEE/ACM 46th International Conference on Software Engineering Education and Training (ICSE-SEET'24),*, (p. 12). Lisbon, Portugal.

Liu, C., Xia, X., Lo, D., Gao, C., Yang, X., & Grundy, J. (2021). Opportunities and Challenges in Code Search Tools. *ACM Computing Surveys (CSUR), 54*(9), 1-40.

Liu, J., Zamir, H., Li, Y., & Hastings, S. K. (2018). Search systems and their features: What college students use to find and save information. *Library and Information Science Research, 40*, 118-124.

Loes, C. N. (2022). The Effect of Collaborative Learning on Academic Motivation. *Teaching & Learning Inquiry, 10*.

Loyens, S. M., Magda, J., & Rikers, R. M. (2008). Self-Directed Learning in Problem-Based Learning and its Relationships with Self-Regulated Learning. *Educational Psychology Review, 20*, 411-427.

Lu, Y., & Hsiao, I.-H. (2017). Personalized Information Seeking Assistant (PiSA): from programming information seeking to learning. *Information Retrieval Journal, 20*, 433-455.

Macak, M., Kruzelova, D., Chren, S., & Buhnova, B. (2021). Using process mining for Git log analysis of projects in a software development course. *Educational and Information Technologies, 26*, 5939–5969.

MacNeil, S., Tran, A., Hellas, A., Kim, J., Sarsa, S., Denny, P., . . . Leinonen, J. (2023). Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. *54th ACM Technical Symposium on Computer Science Education V. 1*, (pp. 931 - 937). Toronto, Canada.

Mahdi, M. N., Ahmad, A. R., Ismail, R., Natiq, H., & Mohammed, M. A. (2020). Solution for Information Overload Using Faceted Search–A Review. *IEEE Access, 8*, 119554-119585.

Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., & Myers, B. (2024). Using an LLM to Help With Code Understanding. *IEEE/ACM 46th International Conference on Software Engineering*, (pp. 1-13). Lisbon, Portugal.

Podeschi, R. J. (2016). Building IS professionals through a real-world client project in a database application development course. *Information Systems Education Journal, 14*(6), 34.

Raibulet, C., & Fontana, F. A. (2018). Collaborative and teamwork software development in an undergraduate software engineering course. *Journal of Systems and Software, 144*, 409-422.

Rangaswamy, A., Giles, C. L., & Seres, S. (2009). A Strategic Perspective on Search Engines: Thought Candies for Practitioners and Researchers. *Journal of Interactive Marketing, 23*, 40-60.

Rieh, S. Y., Collins-Thompson, K., Hansen, P., & Lee, H.-J. (2016). Towards searching as a learning process: A review of current perspectives and future directions. *Journal of Information Science, 42*(1), 19-34.

Sadowski, C., Stolee, K. T., & Elbaum, S. (2015). How Developers Search for Code: A Case Study. *2015 10th Joint Meeting on Foundations of Software Engineering*, (pp. 191 - 201). Bergamo, Italy.

Shuto, M., Washizaki, H., Kakehi, K., Fukazawa, Y., Yamato, S., & Okubo, M. (2016). Learning effectiveness of team discussions in various software engineering education courses. *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*, (pp. 227-231). Dallas, Texas.

Topping, K. J. (2005). Trends in Peer Learning. *Educational Psychology, 25*(6), 631-645.

Tubino, L., Morgan, K., Wood-Bradley, G., & Cain, A. (2023). Shaping a Tool for Developing Computing Students' Professional Identity - Industry Perspectives. *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET).* Melbourne, Australia.

Vakkari, P. (2016). Searching as learning: A systematization based on literature. *Journal of Information Science, 42*(1), 7-18.

Valový, M., & Buchalcevova, A. (2023). The Psychological Effects of AI-Assisted Programming on Students and Professionals. *IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 385-390). IEEE.

Walraven, A., Brand-Gruwel, S., & Boshuizen, H. P. (2009). How students evaluate information and sources when searching the World Wide Web for information. *Computers & Education, 52*(1), 234-246.