Pair Programming Matcher

Final Report

Laith Al-Masri
Computer Science
Virginia Tech
Blacksburg, VA, US
laith21@vt.edu

James Barber
Computer Science
Virginia Tech
Blacksburg, VA, US
jamescbarber32@vt.edu

Linley Spangler
Computer Science
Virginia Tech
Blacksburg, VA, US
linleys39@vt.edu

Sidney Fredericks
Computer Science
Virginia Tech
Blacksburg, VA, US
sidneyf25@vt.edu

Abstract

Finding suitable partners for software engineering projects is often challenging and inefficient due to misalignment in key factors such as schedules, work styles, experience, and goals. These misalignments can result in an imbalance of workload, unclear objectives, miscommunication, and overall disharmony within teams, leading to lower-quality project outcomes. Despite the importance of these factors, there is no current system that effectively matches engineers based on compatibility.

A survey conducted by our team revealed that most engineers rely on existing relationships, the internet, or automatic assignments to find collaborators, but they find these methods inefficient and often unreliable. Respondents noted that having access to a potential partner's skills, qualifications, and availability would improve the selection process. To address this issue, we propose the development of a website that prioritizes matching software engineers based on these crucial factors, aiming to create better-aligned teams and improve project outcomes. This software would streamline the process of finding a well-balanced partner leading to more innovative and efficient projects.

1. Introductions

For many software engineers, it is difficult and inefficient to find a suitable partner for coding projects. There are many factors in a partner that need to be considered, such as schedule, work style, experience, and goals. If these factors do not align for both partners, it can cause an imbalance in workload, confusion over goals, miscommunications, and disharmony among the group, all of which will lead to a lower quality product. Thus, it is essential for any group to find members with such factors in common. However, there is not a current system that helps match software engineers in a way that prioritizes finding a good partner.

In a survey conducted by our team, it was determined that most software engineers find partners using their existing relationships, the internet, or are grouped with someone automatically. Most respondents believed their current way of finding partners was inefficient, and the largest drawback to their current system was that it was often difficult to know if someone would be a good partner without knowing the person first. Respondents also agreed that the ability to see a potential partner's skills, qualifications, and schedules would benefit both parties in a new partner matching system. Therefore, we believe that developing a website to match users based on these qualities will help streamline finding project partners for software engineers, as well as produce better matched teams.

This platform will need to collect relevant information from each user to accurately match them with an equal partner. The algorithm will need to account for a variety of factors and be flexible enough to adapt to a wide range of scenarios. In the end our proposed solution accounts for all these aspects and many more to significantly improve the working conditions for each of our users.

2. Background

There are several key concepts to understand in correlation with this project. The first that is the basis of this idea is pair programming. This is defined as the practice of pairing up two programmers to work on programming tasks [1]. This usually involves two developers sitting together and working at the same computer while sharing the same keyboard, but with the increasing popularity of remote coding platforms this can now be done thousands of miles apart. Pair programming usually produces better results because solutions are thought of faster, problems are identified quicker, and bugs are worked through actively. It also allows each programmer to learn from the other and develop their skills more rapidly.

Another idea that is talked about frequently throughtout the process used is agile development. This is a project management methodology that values individuals and interactions over process and tools [2]. This allowed our team to come together for productive meetings and left us with clear goals to accomplish before we met again. It has many benefits when implemented correctly with a team to include increased flexibility, improved communication, reduced risks, and overall increased production quality.

Although there are many words related to software engineering, the following are some of the key terms that should be understood in an attempt to comprehend this report. Understanding all these important terms will provide for a more

thorough comprehension when other, less used technical terms are mentioned.

3. Related Work

It is a common debate. Does one work alone? Is it more efficient or would a partnership or teamwork be better? A study out of Ireland called “Benefits of working in partnership: A model” emerged to address this vary topic. They found that effective partnerships had several key benefits in several specific areas [3]. The first was enhanced productivity and efficiency. Teams that have compatible work styles and schedules experience greater success. The second benefit to working in a team gaining a higher quality of output. When partners that are equally skilled get paired together, they spur one another on to produce a greater product. Thirdly, partnerships can prevent communication breakdown. Because a partnership is one or two people, there is a stronger communication bond between the two parties. If two partners of similar communication styles and preferences are matched, then there is a reduced percentage of miscommunications. Another benefit to partnerships found in the study is that they balance workloads and reduce burnout. Having a working partnership evenly disperses the work leading to less fatigue. Reason number five that partnerships are positive during projects is that they provide an opportunity for skill development and knowledge sharing. Additionally, partnerships increase motivation and commitment. That link keeps one connected to the project that they are completing, and each person can spur the other one on. Lastly, partnerships improve overall problem solving. Two brains lead to opportunities to expand creatively on problems and produce innovative solutions.

After this concept was revealed, the purpose of the project was clear. However, it remained to be seen if it would be the first of its kind or if it was a repeated idea. Several platforms were found that had pieces of our programmer matching platform, but not to the extent that they needed to be. This first was Skill Display which is a platform for visualizing skill sets and matching individuals for collaboration [4]. This one is mostly used by teachers and had very little features that our model would have. The second platform found was a software called Teammates. This is a platform used by educators to form teams based on complementary skills and preferences [5]. It was closer to the goals set out for our model, but also not at the same caliper needed. In the end, the conclusion was that there was no platform that supported the goals that were laid out for this project.

Overall, because there is such a need for positive partnerships a platform ought to be created to facilitate this process. The research shows that innovative ideas increase with the use of evenly matched partnerships, so an effective process should be created. This is where the Pair Programming Matcher comes into play. With this platform, software engineers will easily find partners that meet the project's criteria that they are working on no matter what.

4. Motivating Example

Suppose a software engineer is working on a personal project outside of work, but the project is too large for them to feasibly do on their own, so they would like to work on it with a partner or group. However, their schedule and skill levels may not align with those they work with, and their friends may not be willing or able to help either. Additionally, they may not feel comfortable asking these people to work with them on this. This is where the Pair Programming Matcher would be useful.

The Pair Programming Matcher allows software engineers to find available project partners with the desired skillsets, schedules, and project goals in an efficient manner. Thus, in the scenario described, a software engineer could create an account and find project partners without having to constantly ask the same questions regarding schedules and abilities. Instead, the app would match them with other software engineers who would be a good match according to the information they provided in their profile. From there, they could talk and decide whether they want to work together on the project. This helps software engineers save time and find project partners that are best suited to their needs.

This app is relevant to software engineers because finding project partners with similar abilities and schedules is a common problem many software engineers face. According to our requirements elicitation survey, many believe that finding a project partner who would be a good fit is difficult and frustrating. While many software engineers decide instead to work alone, this is not always the best solution. Having a partner or team to help develop projects helps provide different perspectives and can improve the efficiency and quality of the project. Thus, an app that helps software engineers find good project partners would greatly improve the experience and results of software engineering projects.

5. Implementation

In order to properly build the Pair Programming Matcher, our team chose a client-server architecture to efficiently separate user interface functions from any back-end processes. That being said, this design approach allows scalable management of user profiles, skill matching algorithms, and even some privacy settings, which ensures seamless communication between users while keeping the system modular and maintainable.

a. Chosen Process

Our team plans to adopt the Agile methodology for developing the Pair Programming Matcher. Agile is an iterative, flexible, and adaptive software engineering process that emphasizes collaboration, customer feedback, and incremental progress. This process involves breaking the project into small, manageable iterations or "sprints," allowing us to focus on delivering functional components of the tool at regular intervals. Each sprint will end with a review session to assess progress, address challenges, and plan the next steps, ensuring continuous refinement of the product [6].

b. Why we chose agile

We chose Agile due to its adaptability and its focus on continuous improvement, which aligns perfectly with the goals of our project. Since our tool requires matching users based on individual skill levels, preferences, and project requirements, Agile allows us to remain responsive to changing requirements

and user feedback. The iterative nature of Agile ensures that we can test and improve our matching algorithm with each sprint, delivering a working product early on while refining it based on real-world data and feedback. Additionally, Agile fosters strong collaboration within the team, promoting the teamwork that our Pair Programming Matcher aims to enhance.

c. Requirements Elicitation

We chose to conduct a survey for our requirements elicitation. Below are the questions and responses received:

How do you currently find project partners to work with?

1. At work or classmates.
2. I do not really have an effective way.
3. I ask online using discussion boards under my class information.
4. Reddit/Twitter/LinkedIn
5. Mutuals/already know them
6. Asking classmates or I am assigned one
7. Usually friends, work, or research
8. My friends
9. Classmates that I sit next to and converse with during class.
10. through classes and assigned projects
11. I find someone with the skill set or knowledge I need to complete the work
12. Usually, I will ask someone I am already friends with

How often do you need to find project partners?

1. If I am at work, often. Otherwise never.
2. Sometimes when I am completing personal projects
3. About once a semester.
4. Depends on the project idea
5. often
6. Not often, only for classes that require groups
7. Not too often. Only really if I am doing a personal project
8. A couple times a year
9. Not that often probably once a month
10. 1-2 times per semester per class
11. Every month
12. At least twice a semester

What do you like/dislike about the current system of finding project partners?

1. It is inefficient.
2. I do not have one.
3. It is quick to see who is available to join a group.
4. It is not practical nor reliable
5. free to choose your own partner
6. I do not really use the current system often, so I cannot really say
7. Um it is low effort
8. I like being able to choose my own partners because it means that I can work with my friends
9. Simple because I already know them.
10. most classes are relaxed about requirements for forming groups, so I like the freedom and flexibility
11. I get to create a network
12. Since I know them first, I can judge that their work ethic is good

What problems/frustrations do you experience with the current system of finding project partners?

1. It is difficult to determine who is good for the project and who is not.
2. I do not really have one.
3. Not sure if they will be a good fit as a partner.
4. I do not have any kind of information that assess the partners I find
5. do not really know much about the person until you meet them face to face
6. It can be awkward to approach random classmates, and if you are randomly assigned someone there is no guarantee they will be helpful
7. If my friends are not interested, it does not really work
8. Sometimes random people get grouped into your group and then you must work with people who you do not know
9. Sometimes working with a friend or someone you talk to most does not help and is not as efficient.
10. when there is limited time to find a compatible partner and difficult to change partners/groups
11. It can be challenging to identify someone with the appropriate skills, experience, and amount of time to help
12. Sometimes it is awkward if things are not working out, but you guys are friends

What specific features/functions would you like to see in a new project partner matching system?

1. Validation of the work they claim. Less ‘fluff’ wording in an application.
2. Easy to use, easy to communicate once you find them
3. See the qualifications or background of the potential candidate.
4. Being able to see their accomplishments send previous areas of interest
5. strengths and weaknesses attribute you can add to yourself
6. A profile with your info where you can show previous projects, an ability to chat with people before committing to work with them, an ability to set what you are looking for in a partner and maybe some kind of algorithm that helps match people based on those criteria
7. Um like a list of projects that people could sign up to work on and give a set amount of time to work on
8. Matching people who have the same schedule.
9. Maybe some sort of grade similarity features so you are paired with someone who puts in similar amount of effort

10. a dedicated 5-20 minute mingle, depending on the class size, allowing people to chat
11. Sort by skill set, capacity, working style, preferred meeting times
12. A way to rate yourself on how early/late you start, or how much you tend to procrastinate

What features do you believe are the most important to have in a new project partner matching system?

1. A good algorithm that considers the needs of the projects and the user together.
2. Efficiency
3. The ability to see information about a potential partner.
4. Finding people with desired skillsets
5. each person's schedules
6. Having a profile, a matching system, and an ability to communicate with both people you matched with and people you are interested in working with
7. Um see the question above
8. How soon before the due date does you like to submit things, how late at night are you willing to stay up to work on homework?
9. To have a fair partner with decent number of similarities to yourself
10. for medium to large scale projects, an honest representation of work ethic
11. Ability to contact someone in the system i.e., a button to email the person

To have it find matches that are in the same class section if necessary, and to include contact information or automatically send notifications

6. Deployment Plan

The deployment strategy our team has chosen to adopt is Blue-Green Deployment. This strategy allows us to maintain two identical environments, one for staging (Blue) and the other for production (Green). During deployment, we perform all updates and testing in the Blue environment while users continue to access the stable Green environment. Once the updates are tested and validated in Blue, traffic is seamlessly switched to Green, ensuring minimal downtime for users. This approach is ideal for our platform as it provides a straightforward rollback mechanism—if any issues arise, we can simply revert to the previous environment by switching traffic back. Additionally, Blue-Green deployment reduces risks associated with live deployments and ensures the reliability of updates before they reach the user base. While it requires additional resources to maintain both environments, the benefits of ensuring a stable and seamless user experience outweigh the costs, making it the most effective strategy for our project [7].

7. Discussion

A possible opportunity and extensions for this application is expanding to other fields such as mechanical engineering or even non-engineering fields. Software engineering is far from the only field that benefits from having a partner or team to work on personal projects with. Extending the scope of the application to these fields would benefit people beyond just software engineers and improve the efficiency of all kinds of projects. Another extension for this application is partnering with other social media sites to help expand the reach of the app. For example, searching for matches on the app could have the option of automatically posting to linked social media sites that you are looking for a project partner, which could help improve the efficiency even further.

Some limitations of this project would be that the effectiveness would rely on users being honest in their profiles. If users exaggerate their skill level to find partners that are better at them so they can offload the work rather than having an equal partnership, then people will be less likely to trust the matches they receive. Additionally, skill level and schedules are not the only things important in an effective partnership. While giving the option to chat with a match before agreeing to work together can help mitigate this, users still may be unsatisfied with the partner they are matched with due to factors that do not appear until after they have already partnered up. Thus, the program cannot guarantee a good partnership.

8. Conclusion

In conclusion, the Pair Programming Matcher addresses a critical gap in the software engineering field by providing a streamlined solution for finding compatible project partners. By considering key factors such as schedules, work styles, experience, and goals, the platform ensures better-aligned teams, reducing inefficiencies and miscommunication. With the ability to assess skills, qualifications, and availability, this innovative tool will not only enhance collaboration but also improve project outcomes. Our proposed solution is designed to foster stronger partnerships, leading to more productive and harmonious teamwork for software engineers.

REFERENCES

[1] Codementor. (n.d.). Pair programming. Retrieved December 16, 2024, from https://www.codementor.io/pair-programming

[2] OpenText. (n.d.). What is Agile development? Retrieved December 16, 2024, from https://www.opentext.com/what-is/agile-development

[3] Boydell, L. R. and Rugkåsa, J. 2007. Benefits of working in partnership: A model. *Crit. Public Health*. 17, 3 (2007), 217–228. DOI: https://doi.org/10.1080/09581590601010190.

[4] SkillDisplay. (n.d.). SkillDisplay. Retrieved December 16, 2024, from https://www.skilldisplay.eu/

[5] TEAMMATES. (n.d.). *TEAMMATES: Online peer feedback for student teams*. Retrieved December 16, 2024, from https://teammatesv4.appspot.com/web/front/home

[6] Agile Alliance. 2024. What is Agile Software Development? Agile 101. Agile Alliance. Accessed September 27, 2024. https://www.agilealliance.org/agile101/

[7] AWS Documentation. 2024. Blue/Green Deployments. AWS Whitepapers. Amazon Web Services. Accessed December 16, 2024. https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/bluegreen-deployments.html#:~:text=A%20blue%2Fgreen%20deployment%20is,running%20the%20new%20application%20version.