

Proposal for Team Building Application

Michael Goff*, Shashank Jha[†], Jingjuan Deng[‡] and Bhaskar Sinha[§]

Department of Computer Science, North Carolina State University,

Raleigh, North Carolina 27606

Email: *magoff2@ncsu.edu

[†]sjha5@ncsu.edu

[‡]jdeng8@ncsu.edu

[§]bsinha@ncsu.edu

Abstract—A study was performed to investigate the need for a team formation application based on the previous experiences of surveyed students in various team environments. The results of the study indicated that students generally had a decent time in their previous team experiences but there were some common minor complaints around topics like communication and sharing the workload. Students also much preferred creating their own teams, rather than having one assigned on their behalf.

After considering user preferences we are proposing an intelligent team formation application using one of three different approaches. The first approach would allow users to create a group and allow other users to apply based on posted skill sets that were requested from the team lead. The second approach would create a pool of users, typically from a class roster, and intelligently match up groups based on their skill sets parsed from a third party service like LinkedIn. The final proposed solution is a system where users can provide a list of their preferred projects and then will be intelligently sorted into groups using a weighted lottery with other skill set considerations.

After developing these approaches, they will be tested on potential users to determine which one would be the best overall solution to forming a team in a classroom environment.

Keywords—Teams, software, matching, proposal.

I. INTRODUCTION

One of the most frequent classroom interactions is team formation. Projects in classes often require the formation of a team to complete a larger task. Professors will often let students work amongst themselves to figure out an ideal team, but sometimes they want to simulate a work environment and assign teams randomly. Sometimes teams formed in these two ways lead to problems in productivity and disputes may occur.

A solution to these problems in teams may be to create an application to intelligently let students create teams or have teams created for them. However, a collection of background information was required to see the feasibility of such an application. With background data in mind, we have come up with three approaches to the team formation problem. The first approach would be to allow users to create a group with specific needs and let other students apply. The second approach would be to intelligently match students together from a pool based on their areas of expertise. The third solution would be to have users supply a list of preferred project topics and then the application would sort these users into groups in a weighted lottery while considering areas of expertise and compatibility.

II. LITERATURE REVIEW

Matching is widely used in social websites to find friends and recommend news. Developers have designed an approach to recommend news stories on Twitter to users, based on their preferences [2]. A paper on Social Matching System claims that the difficulties in matching are the set of key research challenges, and the roster of appropriate methods are all ill-defined. The method to solve these is focusing on things people most care about in different situations. In our design, we should focus on the quality that people most care about to get better matching result.

We can also implement team formation by allocating human resources to a team. People used search-based technique such as genetic algorithm and Tabu search to optimize random allocation in project resources management [3]. There has been a dramatic increase in work on Search Based Software Engineering. This technique applies meta-heuristic search techniques such as genetic algorithms, simulated annealing and tabu search to software engineering problems.

According to previous research, we will consider optimizing our design in some ways such as ranking results in an appropriate order, tuning criteria to ensure an available result and adding limitations in a randomized algorithm.

III. STUDY

In order to figure out how to approach a team building application we needed to determine what the user needs. We concluded that the application could be used from both a teacher's perspective and a student's perspective; however, we decided to focus our study on the student aspect since they are the users who are most affected from a team assignment.

A. Research Questions

In this section we describe the different questions we put forth in a Google Survey and why we decided to ask each question. We ultimately decided on a Google Survey in order to get a quick response from a larger pool of people. With an online survey it is easy to push it out to many peers including those beyond the scope of the class.

Q1: Are you a master's student?

We wanted to ask this question to get a sense of the type of students who were responding to our survey. Master's students are different from undergraduate students because they have

different workloads and, as a result, different experiences in team projects.

Q2: How many different team projects do you work on in a year?

We wanted to get a sense of how often people work in group projects. The more group projects they participate in, the more chances they would have of a group being especially poor or exceptional. It would provide a better perspective on the individuals experience overall.

Q3: Do you face problems finding team members with specific skill sets?

This relates to our idea for some of the application approaches. We want to find a way to automate the process of finding a variety of skills and forming teams with the widest breadth of expertise possible. When you have different people who are experts in different areas it may be easier to distribute the workload among the team.

Q4: Which services have you used to form teams before?

We wanted to ask this question to see what kinds of tools students already use to form teams. By having the perspective of the current tools students use, we can investigate what makes those tools useful and what kind of drawbacks they have.

Q5: What experiences did you have with those team members which you found?

We asked this as an open ended question because we wanted to see what kinds of different experiences the users have previously had. Our team has an idea of what could go wrong on a group project but we wanted to avoid anchoring the survey participants to a specific problem and see if there were other pieces that we were missing.

Q6: On a scale of 1 to 5, five being the highest, how successful were your previous team projects?

This question was meant to gauge the success students have had in previous projects. If the responses were mostly negative, we could use it to justify a need for a new tool in team formation.

Q7: What were some problems that you had with team members in the past?

We wanted to dig into those negative experiences users had about working others to see if there were some potential problems that we could help address in our application. This was another open ended question where we wanted to explore the types of issues that could occur without framing the participant's thoughts.

Q8: What other qualities aside from technical expertise do you look for in a potential team member?

We know the main benefit of finding a good team member is their technical expertise, but we wanted to dig deeper and see if there were other major aspects that were worth exploring like social benefits or other unknown intangibles.

Q9: In general, while doing a project, which is better for you? Randomly Assigned Teams or Freely Formed Teams.

This was a two option question. We wanted to know what students would prefer in a team project. If there was an overwhelming preference for one or another, we could shape our application to use student input in forming teams instead of an emphasis on matching using algorithms.

Q10: While doing a project do you prefer to collaborate online or in person with your team?

This was another social test of sorts. We wanted to judge whether an online application would be beneficial for these users. If they prefer to work online then perhaps the social aspects of a team would not be as important since there would be much less face-to-face social interaction.

Q11: How often do you assess a potential team member's skills before forming a team?

We wanted to see how much a user has even thought about or considered the skills of other students before committing to working with them in a team. If students thought a lot about those considerations then the usefulness of an application for matching based on skills would be beneficial.

Q12: If there was a potential web application for team creation, would it be helpful to you?

This was a direct question relating to our application. We wanted to gauge how much interest there would be in an application that could help students to form teams. If there was a lot of interest we could be really making a difference in the ease of team formations with our application.

IV. SURVEY RESULTS

We received 41 responses in our survey at the time of writing through the Google Survey. Below is a collection of the results and our thoughts regarding the outcome.

A. Results

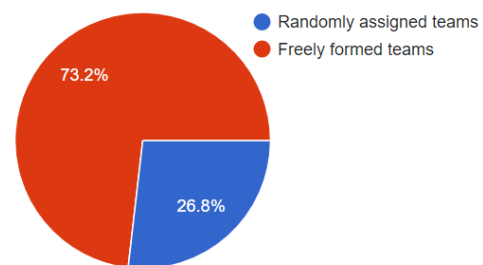


Fig. 1. Randomly Assigned vs. Freely Formed Teams

In figure 1, most of the participants much preferred to freely form teams over random assignment. This result indicates that students would rather control their own group by having a say on who is involved. When it is a randomly assigned team there is always a risk of having a known slacker in the group who would have been explicitly avoided if it were a student chosen team. Another reason could be that students would prefer to work with their friends on a project since they already know the kind of work ethic their peers have and enjoy their existing time together.

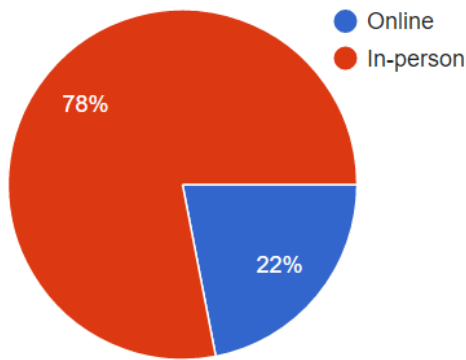


Fig. 2. Would you rather work together online or in person?

Similar to freely formed teams, working in person was much more popular than working online by a very similar ratio, as seen in figure 2. The preference seems to imply that groups are more successful that work together in person rather than online. The question format could have been more specific about online meeting to include web video conferences or something similar. Perhaps ideas and best spread in person rather than through the Internet. The results also imply that a heavier social similarity will lead to a stronger likelihood of success among teams.

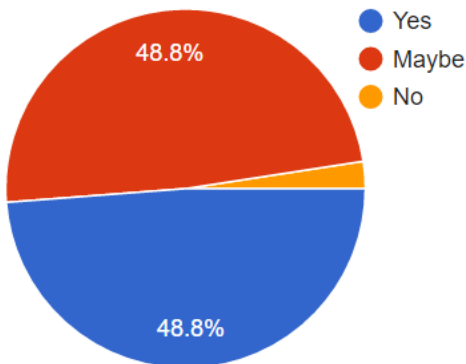


Fig. 3. Would a potential web application be useful to you?

The results from Q12 were promising for our application. As shown in figure 3, most of the participants marked either yes or maybe in considering the utility of a team-building application. In retrospect, it might have been better to make this kind of a question a yes or no style of question. The addition of maybe as an answer choice lets the survey participant answer without actually considering the application itself. However, about half of the participants did explicitly mark 'yes' for the question so some interest could exist.

Question 3 did not present much surprise in its results, shown in figure 4. The choice 'Sometimes' took a vast majority of the votes as students seem to have a problem from time to time. This is another instance of a question where 'sometimes' probably should not have been an option or this question should have been presented as a scale. We are unable to get much real information from these results. There was a small edge of people never having problems over those who have problems all the time though, but that might just be a result

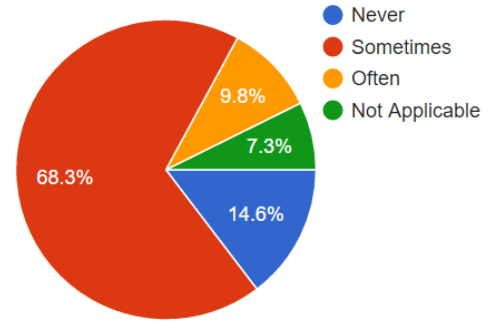


Fig. 4. Have you ever had problems finding team members with specific skill sets?

of the somewhat small sample size.

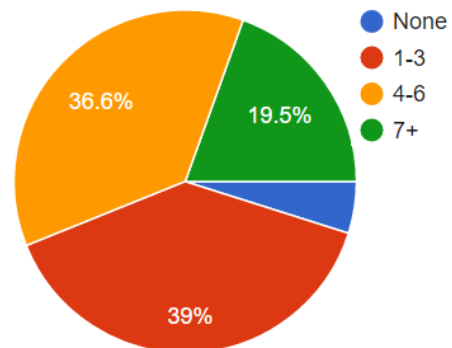


Fig. 5. What's the number of team projects you have worked on in a year?

The spread of the number of team projects that students partake in a year were not too surprising, as seen in figure 5. About a third participated in 4-6 projects a year which roughly lines up with a full workload for two semesters of work at school as a graduate student. Having about 40 percent working in only 1-3 projects was somewhat surprising since that accounts for only about 1-2 projects a semester. Most classes seem to focus on group work, at least here at NC State. Perhaps there was some influences of the 25 percent of people who were not master's students taking this survey.

Question 5 was an open end question regarding experiences with team members. Since we avoided asking for any positive or negative experience, we were surprised to see that most of them replied with negative ones. Lack of coordination, work division, different goals, different expectations, schedule conflict, etc were mostly mentioned. In question 7, where we specifically asked for problems they faced in past, responses were similar. Hence, proving our point that existing team creation methods are not perfect. If we can address these problems, our product is sure to have a user base.

Question 8 was again an open ended question. By asking this, we wanted to know what exactly people are looking for beside the technical skills. It would be useful for us to understand our target group. Thus we will design our application and search criteria accordingly.

Once again in question 11, we made a mistake by putting in sometimes as an answer choice. As shown in figure 6,

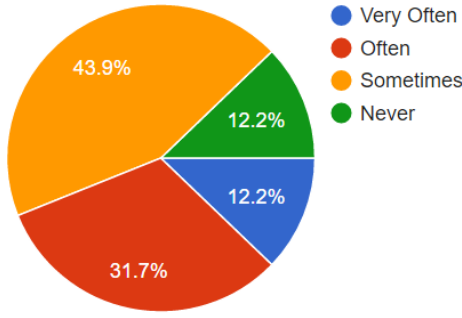


Fig. 6. How often do you assess the skills of potential team members?

'sometimes' was a dominant answer with about 44 percent of the votes. A better scale would have been to use a number system to try to split out the 'sometimes' votes into something more granular. These results otherwise seem to indicate that assessing others skills is a pretty common occurrence with about 44 percent of the votes spread between 'very often' and 'often' which implies that relevant skills are important to team success.

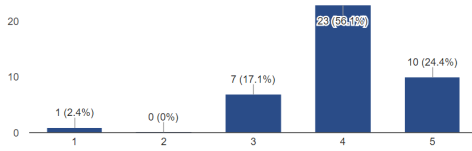


Fig. 7. How successful were your previous team projects?

The success of previous team projects for most survey participants were a 4 out of 5 indicated on figure 7. The majority of responses were leaning towards more success with the vast majority in the 4 and 5 categories with a few average experiences at 3. There was only 1 of the 41 participants who had a strongly negative experience. We think these statistics can be improved even further with the help of a team generation tool. With the tools help, more people can have a better group experience and push the graph more to the right.

V. APPROACHES

First, to start using our application, a user needs to create their profile. Rather than adding all the details one by one, we will be fetching information from their LinkedIn profile. Hence, skill sets, experience, previous project work, etc will all be added in one click. They will also have the option to add (or remove) any more details.

Approach 1

According to our survey, around 78% of people face a problem finding team members with specific skill sets. Hence, we want to address that problem. The first approach is based on what we are already familiar with: a user searching for other team mates. But rather than showing other users randomly, we will display the result according to the number of skill set matches. If a user is looking for team mates who are good in python, database management and web designing, he can

add those in search criteria. Then, output will be displayed in descending order of matches. The user can then connect with whoever they are interested in working with. There is also a threshold value that indicates the least percentage of skills that should match. a user will be able to toggle that value depending upon their needs.

Notice that here user does not only mean a single person. It can also be a team who is looking to add someone with specific skill set. Hence, it can be used to start a team from scratch or help complete one.

Approach 2

Our second approach is targeted for someone under whom a pool of people is already available. The best example can be a professor. Lets say they have a project in mind for which they want their class to work. One way to divide class into teams would be to let the students decide their team mates by themselves. But doing so would mean they would not be prepared for a real life scenario where they may have to work with new team in office. Thus, there is need of a second approach, which is randomly assigning a team. This poses one big problem. What if all the members have similar skill sets and it does not cover all the requirements for completion of project? It would be much better if the skill sets are distributed in an even way. This way, everyone in the team can take lead in whatever their strong point is. Everyone can contribute to the project. Also, if they are not familiar with other skill sets, they can learn from their own team mates. Hence, it would be a win-win situation.

To achieve optimized random teams, the person in charge needs to define the total number of members in each team and the total number of resources available. Our application will take care of team creation in a fair way. If the user base is only 4-5, the solution provided by us might be not be standing out. But considering the fact that few classes have student count around 100, manually dividing fairly would be exhausting. This is when our second approach comes into play.

Approach 3

The third approach is based on randomly generating algorithm to create project teams. Since we notice that even though 73.2% percent of participants' experience are freely formed teams, there are still 26.8% other cases in which people formed teams by randomly assigning. Additionally, in some cases freely formed teams is not practical nor well estimated. So randomly assigned team is a candidate solution to solve these problems.

Random algorithms are usually used in program testing [5] and system scheduling[6]. This kind of algorithms have some advantages: (1) Efficiency: since the result is generated by random factors, it saves many time from considering and calculating complex factors. (2) Fairness: ensure every eligible candidates have the same chance to compete for joining a team. And two separate runs may have different results. (3) Objectiveness: the result is determined by criteria in program, not by any person.

We combine random algorithm and requirement limitations to generate quality teams in a easy way. To illustrate this in an example, assuming we have a talent pool and some topics

available(topics are came up by a manager, a faculty, a student or people who wants to form a team). People in the talent pool can choose at most 5 topics which have slots fit for their skills. The topics they choose have different weights according to their preference. There may be several candidates that apply for the same team. In this case, we should have several runs, people who haven't been chosen in the first run will apply for their second topic in the next run. So on and so forth until every candidates find their teams.

VI. CONCLUSION

Our study shows the problems in team formation and teammate matching. The survey indicates that most of the participants have problem in finding team members with specific skills. So we will design team formation solutions that focus on searching and matching teammates with skills required.

Some software solves similar problems such as friends matching, news recommendations and project resource allocation. Inspired by previous research, we designed 3 solutions to solve team formation problem. The first solution is user searches for other team mates by some criteria. The second solution is randomly generating teams in a talent pool. This solution is efficient but the result may not satisfy every team member. The third solution is generating teams using users' preferred list, this solution is a two-way selection.

To evaluate the teams generated in our solution, We collect LinkedIn profiles from the survey as talent pool. It is difficult to make sure every teams are formed perfectly, evaluation should be various according to different situation: (1) For one individual, evaluate the ability to find optimal team. (2) For one specific team, evaluate the ability to find optimal team members in talent pool. (3) For every teams, evaluate the throughput of satisfied scores.

REFERENCES

- [1] Loren Terveen. Social Matching: A Framework and Research Agenda. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2005.
- [2] Owen Phelan. Using twitter to recommend real-time topical news. *RecSys '09 Proceedings of the third ACM conference on Recommender systems*, 2009.
- [3] Massimiliano Di Penta. The use of Search-Based Optimization Techniques to Schedule and Staff Software Projects: an Approach and an Empirical Study. *SOFTWAREPRACTICE AND EXPERIENCE*, 2009.
- [4] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [5] Andrea Arcuri. A Hitchhikers Guide to Statistical Tests for Assessing Randomized Algorithms in Software Engineering. *Software Testing, Verification and Reliability*, 2016.
- [6] Carl A. Waldspurger. Lottery Scheduling: Flexible Proportional-Share Resource Management. *OSDI 94 Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation*, 1994.

VII. CHIT WORDS

- bkggaeau
- chfxoaua
- cjhgooe
- ckfhiieo
- clfnuuui
- dbhfoaui
- dcgnauou
- ddfdaiaa
- dhcneieu
- dmgneuio
- fbdfuuuo
- fcgcaeii
- flcsaiuo
- fmhzoaio
- gbljuuue
- gccmioio