Comprehensive Resource Development for the Technical Interview Preparation Club

For Jenna Baird

President, Co-founder

Technical Interview Preparation Club

Vancouver, British Columbia

By

Geneviève Bolduc

University of British Columbia

August 1, 2020

Table of Contents

Abstract

[Placeholder for final draft]

Introduction

The Technical Interview Preparation club (TIP) is a student-led group that meets weekly to build up interview readiness. Currently, the format is weekly video conferencing where students and recent alumni present and solve algorithms that might be asked in interviews by future employers. These questions are pulled from the popular website LeetCode which showcases problems asked by companies including, but not limited to FAANG (Facebook, Amazon, Apple, Netflix and Google). These employers are at the forefront of software innovation and hire many graduating students every year. Currently, LeetCode is the only source the club is utilizing. However, problem-solving is only one aspect of getting hired. Gayle Laakmann McDowell, author of *Cracking the Coding Interview*, outlines many aspects under review in an interview: analytical skills, coding skills, technical knowledge, experience, and culture fit (Laakmann McDowell, 4). Currently, the club does not have a body of resources to sharpen all of these skills, nor do the weekly mock interviews necessarily measure for these.

Problem Statement

The Technical Interview Preparation club does not effectively capitalize on its mandate. Student readiness for these challenging interviews should take into account wider skills than algorithm solving. Otherwise, it is likely that students and recent alumni attend interviews with incomplete skillsets and perform poorly in some aspects.

Proposed Solution

The TIP club should build an accessible and shared body of literature that develops the necessary interview skills. Different exercises and resources should be available to practice said skills. There are many examples of what might be added to this package. For instance, technical knowledge often refers to broader questions in computer science, often around design patterns and practices. Common questions like these could be compiled into a unified document. As for experience, resources could be shared on building individual and group projects on a resume. How to get started? What tutorials are useful? As for culture fit, employers could be interviewed for more detailed insight on what they search for as well as sifting through the popular website Glassdoor to consolidate this information. All of these could be collected and archived in a shared folder.

Scope

To assess the feasibility of the proposed solutions, I plan to pursue five areas of inquiry:

1. What additional skills would members of the club like to practice?
2. What non-algorithm based factors are most important in interviews?
3. What are common theory questions asked in interviews?
4. Where to get started with extracurricular personal and collective technical projects?
5. What are the ideals of company culture at different tech companies, local and international?

Methods

My methods consist of conducting anonymous surveys among members of the TIP and collecting interviewing criteria from the bible of software engineering interviewing: *Cracking the Coding Interview* by Gayle Laakmann McDowell.

My Qualifications

I am a fourth-year computer science student, nearing graduation as well as one of the organizers for the TIP club. I have had multiple technical interviews and have some industry knowledge. I plan to use this to maximize the effectiveness of the club's mandate.

Audience

The intended audience for this formal report is the leadership and membership of the TIP club. With their efforts, a community resource can be built to address the lacking elements of the club.

Conclusion

There are many ways to succeed in the field of computer science, and those who collectively organize to better their skills should be able to access a large body of helpful resources.

Data Section

Analysis of Interview Facets

Based on *Cracking the Coding Interview*, the skills required for successful interviews are analytical skills, coding skills, technical knowledge, experience, and culture fit (Laakmann, 4). Broken into greater depths, analytical skills represent an interviewee's ability to break down a problem and piece together a preferable optimal solution. Coding skills represent the cleanliness and accuracy of the code. Was the solution effectively translated into code? A good solution will only go so far if it is not properly executed. Interviewers will be seeking well-coded solutions. Technical knowledge refers to fundamentals of computer science. Interviewers can assess technical knowledge by asking about any basic concepts or technologies. These questions can be particularly challenging because the body to choose from is extremely vast. Experience can be measured in terms of both soft skills (leadership, drive, initiative) and technical skills (projects, Hackathons, technical work). An interviewee able to highlight those skills and provide detailed insight into their experience will have a competitive edge. Tools like a detailed resume will serve as an asset here. Culture fit is a slightly more nebulous criteria because it can vary between companies. Some of the larger aspects that can be analysed in this context is effective communication and shared values. Does the interviewee express themselves clearly? Does the interviewee's personality mesh well with the interviewers? These guiding principles for interviews provide a holistic perspective on interviews for prospective software engineers. It is based on these principles that Laakmann McDowell was able to develop this roadmap for interview preparation. Every step along the grid below serves to aid prospective interviewees in one or many of the five facets highlighted above.

▸ Preparation Map

The following map should give you an idea of how to tackle the interview preparation process. One of the key takeaways here is that it's not just about interview questions. Do projects and write code, too!

```
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│   1+Years       │  ───▶ │ Build projects  │  ───▶ │ Learn multiple  │
│ (before         │       │ outside of      │       │ programming     │
│  interview)     │       │ school/work.    │       │ languages.      │
└─────────────────┘       └─────────────────┘       └─────────────────┘
                                                              │
                                                              ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Students: find  │       │ Build website / │       │ Expand Network. │
│ internship and  │  ◀──  │ portfolio       │  ◀──  │                 │
│ take classes    │       │ showcasing your │       │                 │
│ with large      │       │ experience.     │       │                 │
│ projects.       │       └─────────────────┘       └─────────────────┘
└─────────────────┘
        │
        ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Professionals:  │       │                 │       │ Continue to     │
│ focus work on   │  ───▶ │  3 - 12 Months  │  ───▶ │ work on         │
│ "meaty"         │       │                 │       │ projects. Try to│
│ projects.       │       └─────────────────┘       │ add on one more │
└─────────────────┘                                 │ project.        │
                                                    └─────────────────┘
                                                              │
                                                              ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Read intro      │       │ Make target     │       │ Create draft of │
│ sections of CtCI│  ◀──  │ list of         │  ◀──  │ resume and send │
│ (Cracking the   │       │ preferred       │       │ it out for a    │
│ Coding          │       │ companies.      │       │ resume review.  │
│ Interview).     │       └─────────────────┘       └─────────────────┘
└─────────────────┘
        │
        ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Learn and       │       │ Implement data  │       │ Form mock       │
│ master Big O.   │  ───▶ │ structures and  │  ───▶ │ interview group │
│                 │       │ algorithms from │       │ with friends to │
│                 │       │ scratch.        │       │ interview each  │
└─────────────────┘       └─────────────────┘       │ other.          │
                                                    └─────────────────┘
                                                              │
                                                              ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Do several mock │       │ Do mini-projects│       │                 │
│ interviews.     │  ◀──  │ to solidify     │  ◀──  │  1 - 3 Months   │
│                 │       │ understanding of│       │                 │
│                 │       │ key concepts.   │       └─────────────────┘
└─────────────────┘       └─────────────────┘
        │
        ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Continue to     │       │ Create list to  │       │                 │
│ practice        │  ───▶ │ track mistakes  │  ───▶ │    4 Weeks      │
│ interview       │       │ you've made     │       │                 │
│ questions.      │       │ solving         │       └─────────────────┘
└─────────────────┘       │ problems.       │
                          └─────────────────┘
                                                              │
                                                              ▼
┌─────────────────┐       ┌─────────────────┐       ┌─────────────────┐
│ Begin applying  │       │ Review / update │       │ Create interview│
│ to companies.   │  ◀──  │ resume.         │  ◀──  │ prep grid       │
│                 │       │                 │       │ (pg 32).        │
└─────────────────┘       └─────────────────┘       └─────────────────┘
                                                              │
                                                              ▼
```
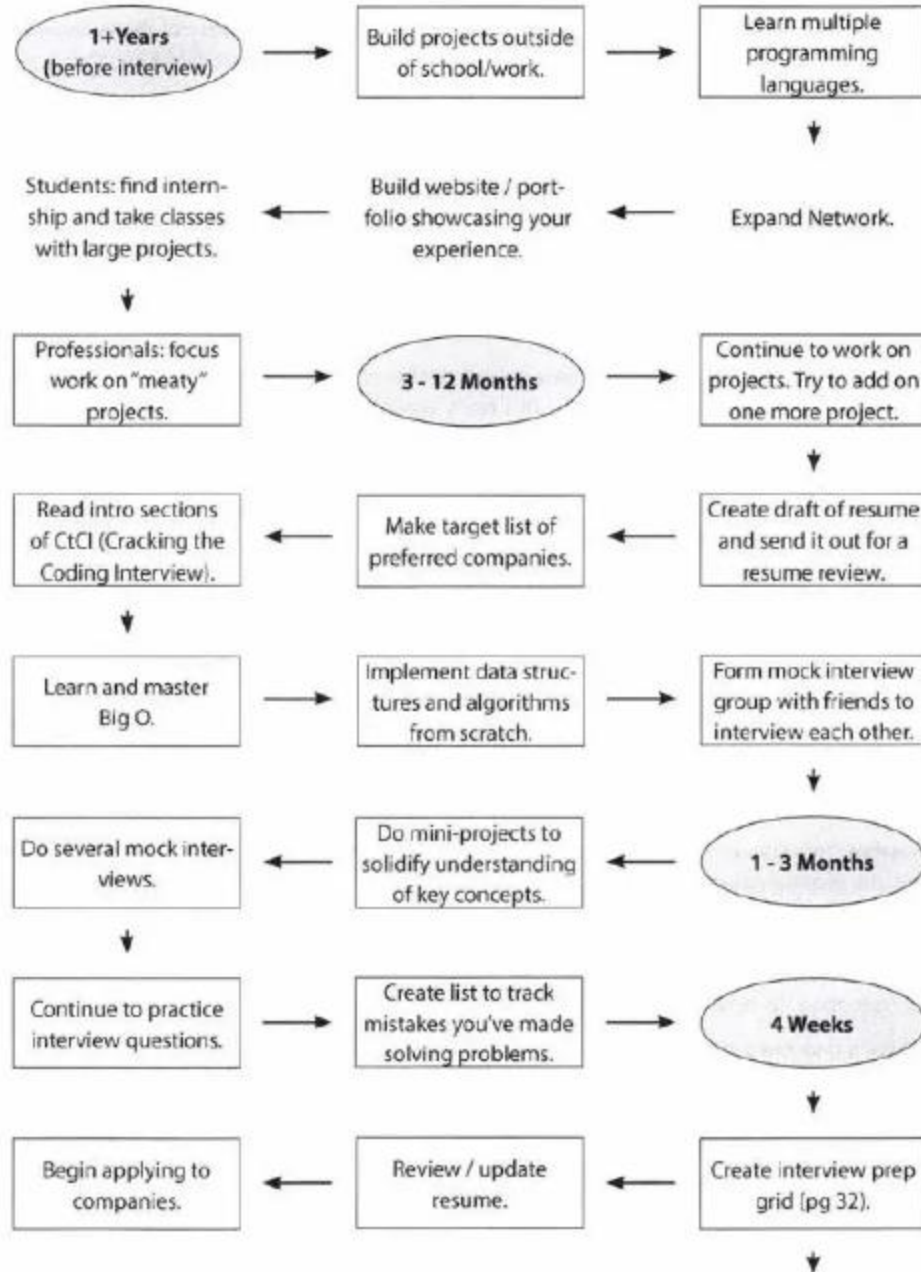
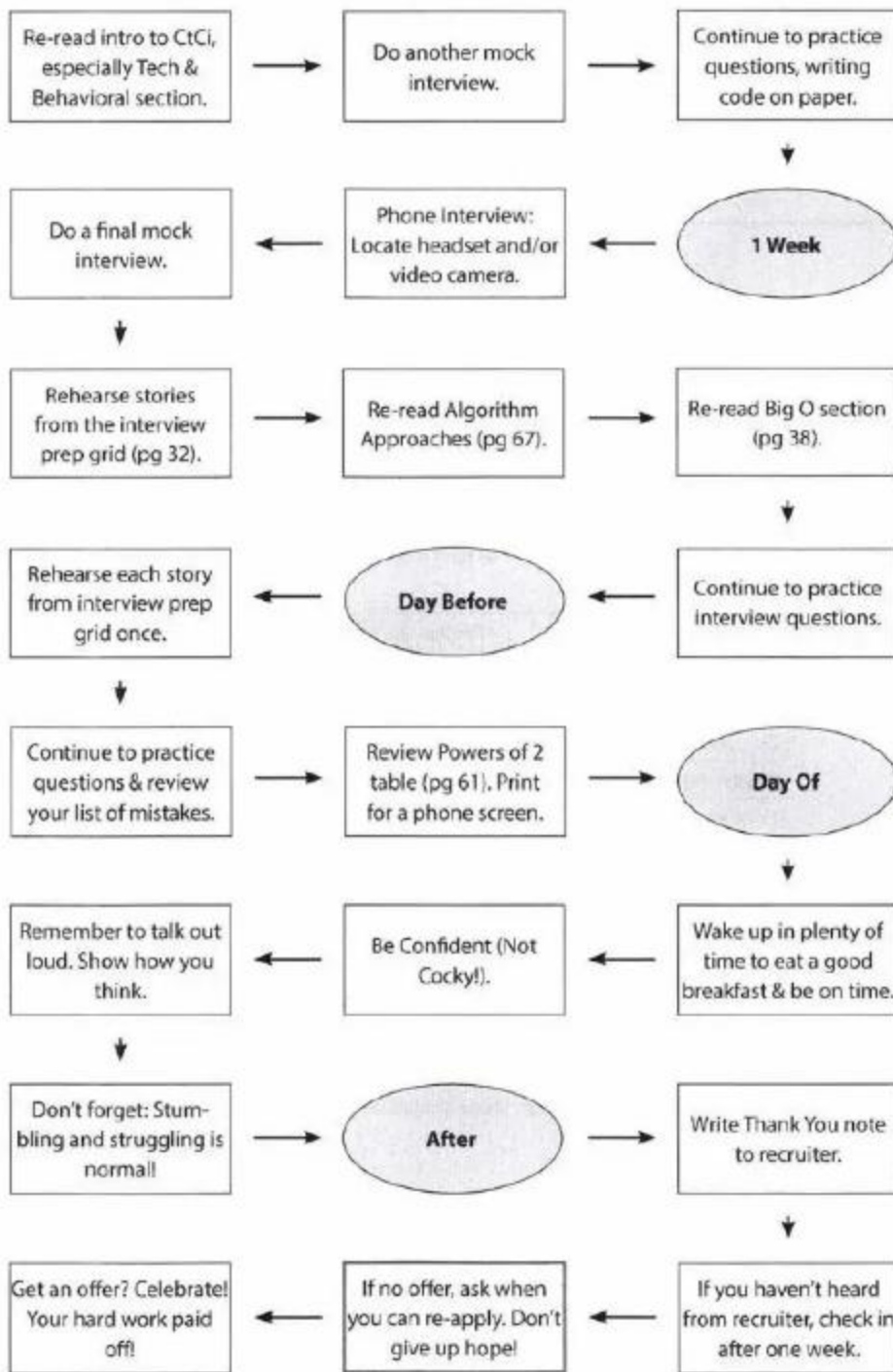Figure 1. Preparation Map, Part 1 (Laakmann McDowell, 30).

Figure 2. Preparation Map, Part 2 (Laakman, McDowell, 31).

These visuals provide us with a concrete path to for a prospective interviewee to follow. They are relevant because such a roadmap can provide us with information invaluable to the development of resources for a club that purports the mandate of preparing such interviewees.

Membership Requests

The surveys conducted polled the TIP membership for multiple factors primarily based on experience (interviews and work) as well as prospective requests for club improvements. Ten members of the existing seventeen filled out the surveys. Requests from the membership were assessed by the following questions:

1) What could the TIP Club improve on?
2) Would you like a unified folder that anonymously catalogues the working and interview experience of TIP members?
3) If so, where would you like to see this information?
4) What, if anything, would you like to see specifically catalogued?

TIP Club General Improvements

Question 1 from the above list was a long-answer text question designed to assess any shortcomings of the current TIP format. The main results could be categorized into three primary categories and a few outliers.

Category 1: Theoretical Questions
This pattern emerged from the following requests:
- "It would be cool to have more questions about design patterns and theory. I feel completely unprepared for these categories but am feeling good about my algorithms."
- "More CPSC310 exam-type questions please! I've already forgotten about most of those slides…"
- "I want to know how I might be tested about specific languages. Do interviewers just ask you to code in a language that you've put on your resume or do they ask about how the language works?"

The reference to CPSC 310 questions is in regard to the UBC CPSC 310 software engineering course. Many theoretical concepts were covered and tested in this course. They include: abstraction, async, automation, testing (white box and black box), design patterns, design principles and many, many more. Because of this, this comment was filtered as theoretical. Knowing the intricate details of language operation could be filtered in both practical and theoretical categories but because of the excerpt "how the language works" suggests language fundamentals. For these reasons, they were sorted as part of the above categories.

Category 2: Company-based knowledge

This pattern emerged from the following responses:
- "I'd like to know which companies are LGBTQ+ friendly."
- "It'd be nice to have testimonials from co-op students and recent grads who've worked at various tech companies. It's not always an indicator but it's nice to know how workers feel about it."
- "I don't know most local tech companies, I see them on the co-op boards and, apart from the giants, I have no idea what they do and what their workplace is like."
- "It'd be nice to be informed of Networking events once we graduate and don't receive the UBC e-mails anymore".
- "I'd like to see what companies asked what coding questions/"

The above responses can be broken down into company-based knowledge and cover the following facets: company culture, mandates , networking and interview practices.

Two outliers that were not able to be categorized as cleanly as the above were reported.
- "People need to respect the time limit in the video calls. Sometimes it goes on too long because of the questions, when they could be addressed via chat afterwards."
- "We should get access to funding so we can expand."

Both of these are outside the scope of this project. The first because it relates directly to management of existing protocols rather than the expansion of resources to fulfill its mandate. The second because

financial resources are not the subject of this investigation. Although funding is instrumental in accessibility to resources, a separate budgeting proposal could better address this topic.

Documentation

Questions 2-4 outlined in the membership requests introduction pertained to prospective shared documentation accessible by the members of the TIP.
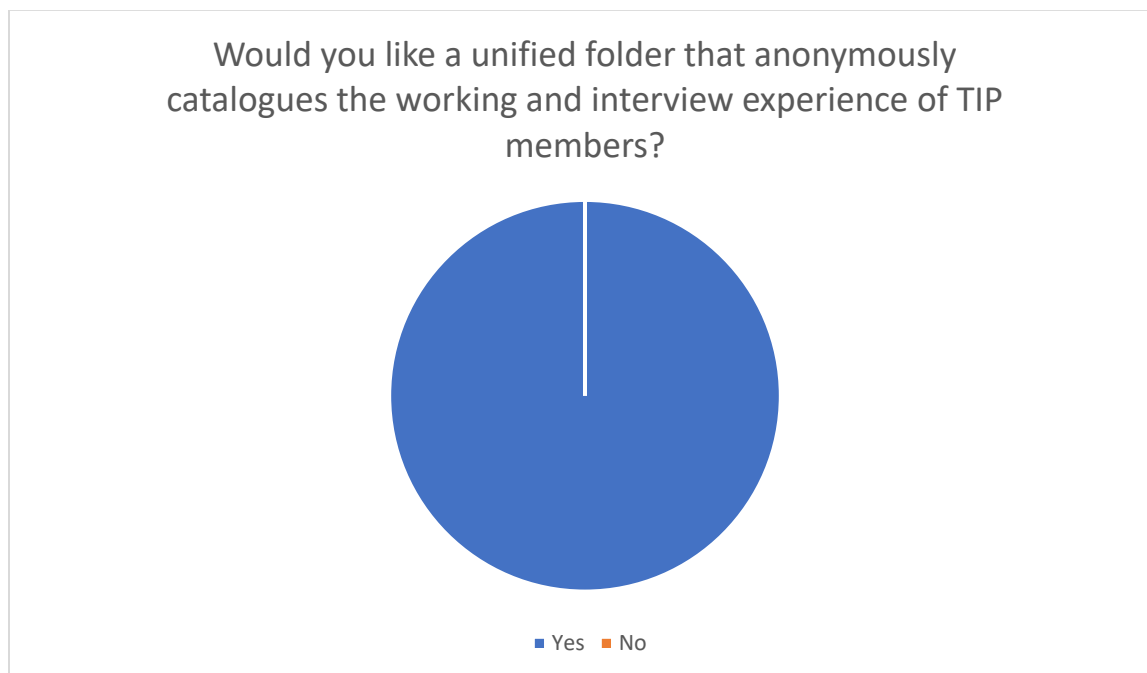
Would you like a unified folder that anonymously catalogues the working and interview experience of TIP members?

■ Yes ■ No

Figure 3. Pie Chart of Cataloguing Question

As demonstrated in figure 3, the ten members who filled out the survey answered yes to the cataloguing.

If so, where would you like to see this information?

■ Google Drive   ■ Dropbox   ■ OneDrive   ■ N/A
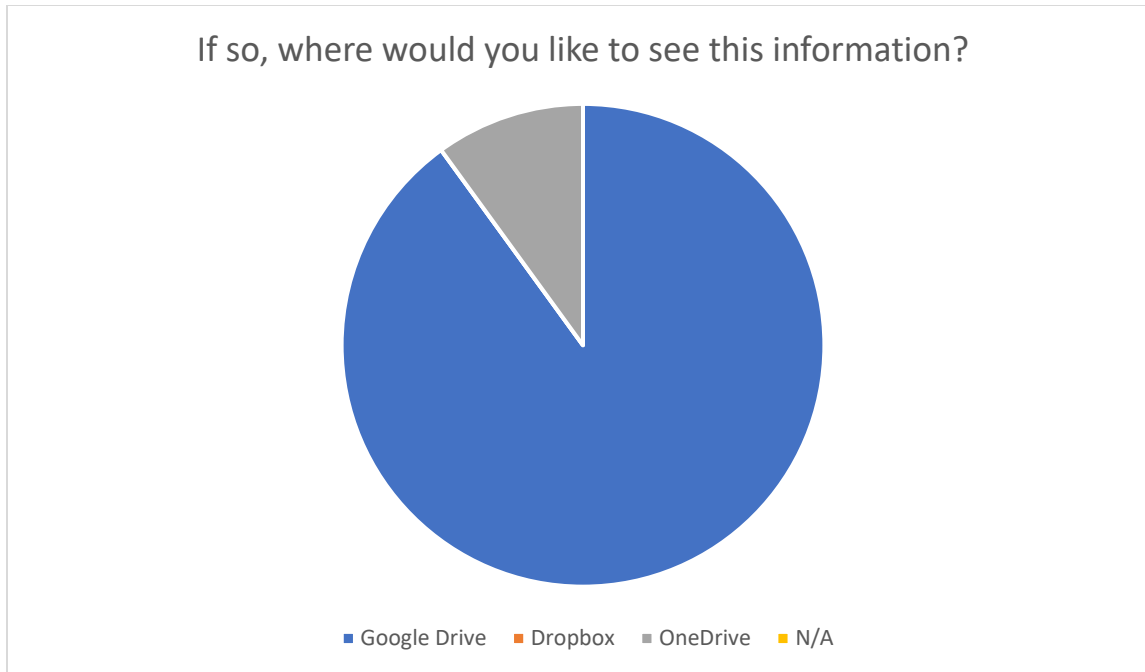
Figure 4. Pie Chart of Resource Sharing Medium

A large majority (9/10) indicated Google Drive as their preferred resource sharing medium.

Question 4 was a long-answer text question to see what the membership felt would be important to have in the body of shared documentation. Here are the answers that were submitted:

- "Company reviews, anonymous and accessible only to the membership."
- "I'd love to see what corporations asked what questions during interviews."
- "I mentioned this earlier, but maybe some mini-profiles on companies or something along those lines."
- "Technical problems that people had to solve in their interviews."
- "Experiences LGBTQ+ folks have had working in tech"
- "Tips that worked for people during interviews. Like what to do if you're blanking."
- "Maybe a list of recruiters to add on LinkedIn?"
- "What people's experiences were with different companies. A mini-GlassDoor situation"
- "Anything useful really."
- "Technical problem list and Youtube explanations of solutions."

Experience

We asked members who interviewed in the tech industry. We received eight yes responses and two no responses. Among the "yes" responses, two members said they experienced between 1-5 interviews, three members responded that they had experience 6-10 interviews, two members stated 11-15 interviews and one member said 21+.
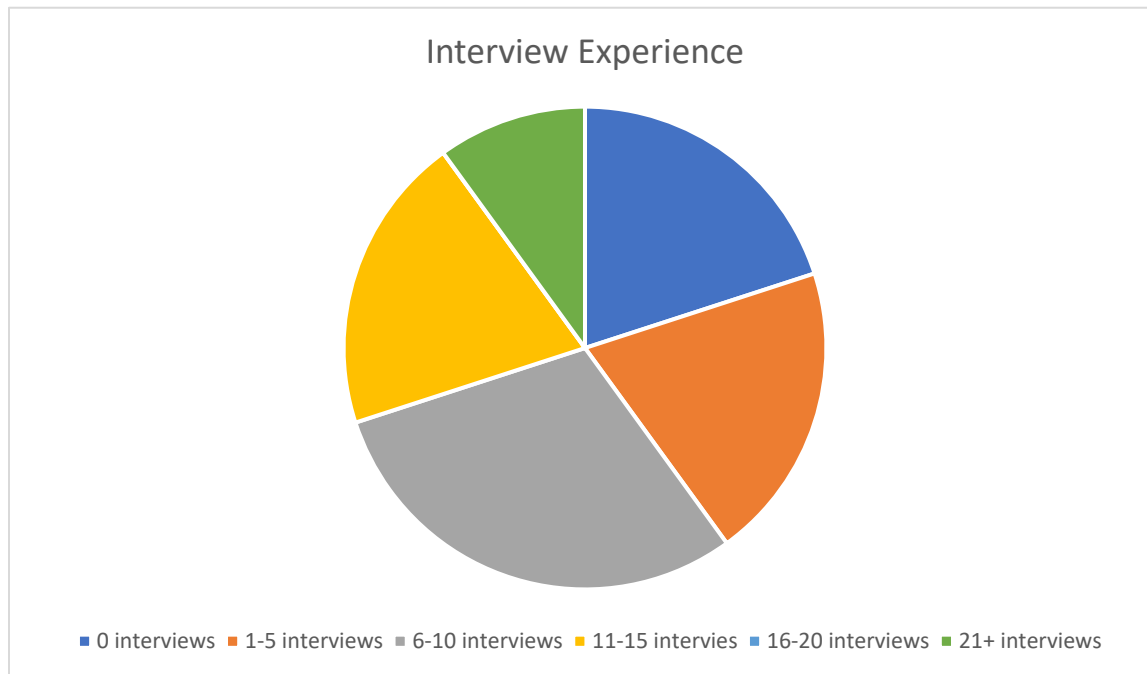
Figure 5. Summary of Interview Experience

We also inquired as to whether subjects had work experience. We received six "yes" answers and 4 "no" answers.

Figure 6. Summary of Work Experience

The survey extracted information about interview and work experience. In the interview segment we were collecting information about what questions were asked, while in the work experience, we were gauging a feel for work cultures.

Interview Experience

The core interviewing question was "Please describe any questions that you remember from the interviews. If able, let us know which company the question was asked by."

Here are some of the responses we got:
- "SAP asked about the Fibonacci sequence"
- "Samsung asked me to write an algorithm that locates all the numbers that have only the same digit (i.e. "1", "22", "888") within a range of n numbers"
- "AppNeta asked me to explain what a React component was and to code a component"
- "Blackbird gave me a take-home challenge. I forget what it was now, but it was three questions."

- "Google asked me this Leetcode question https://leetcode.com/problems/climbing-stairs/ "
- "It's all a blur TBH."
- "I interviewed for Hootsuite, they asked me about the 3-color graph problem."

Work Experience

Subjects with work experience were asked which companies they had worked for and to give them a rating. In this instance, no two candidates worked at the same company. Therefore, we have collected the rating results as follows.
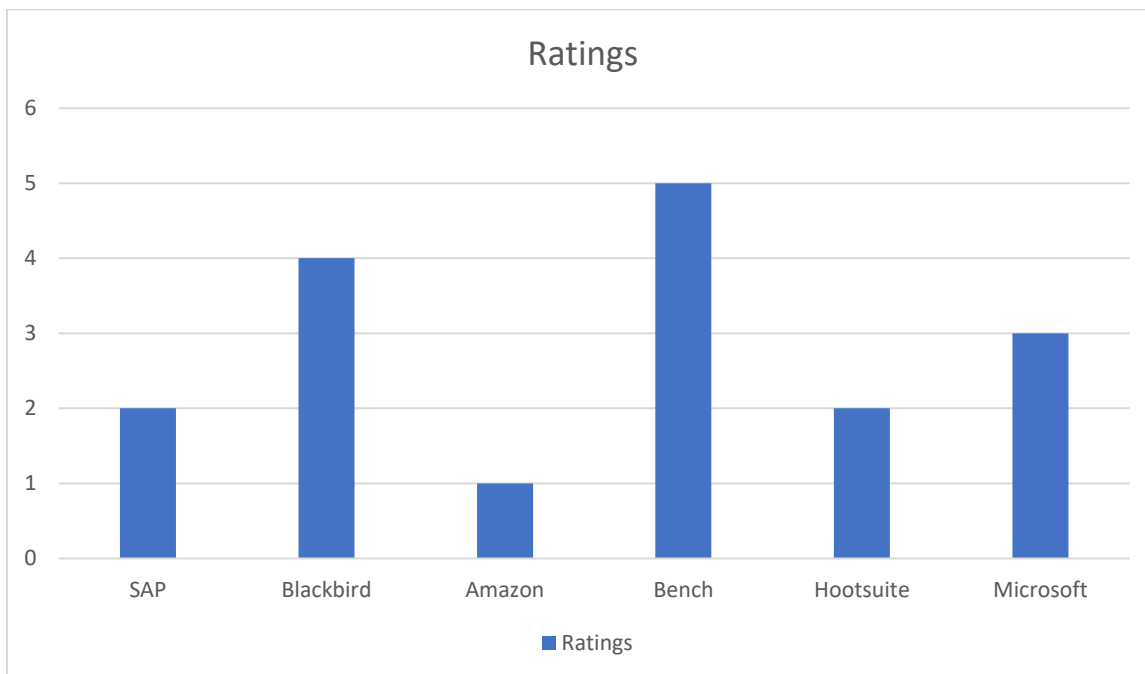

Figure 7. Company Ratings of TIP Membership with Work Experience

A central question to this section was "Describe the work culture" of the respective company the subject rated. Since this was a long-answer text question, the responses were varied:

- "SAP was weird. It felt very cliquey. And if you didn't like the team you were on, you were SOL."
- "I liked my time at Blackbird. At times, the work felt above my skillset but everyone was real nice about it."

- "Amazon was the worst. Bad ethics and they expected you to work insane hours."
- "Bench was great! I really felt like everyone was there to support each other, and I was able to directly contribute to product features."
- "Hootsuite had a super bro-y vibe. Not surprising though when you think about how employees would ice each other just a few years ago."
- "Microsoft is okay. Some nice perks. Will probably continue working there until I get tired."

Conclusion

The data collected provides a tangible screenshot into the requirements for successful interviews as well as the wishes and experience of the TIP membership.

Summary and Interpretation of Data

The Laakmann McDowell research provided us with a framework of understanding software engineering interviews in the tech sector. From this, we could ascertain that the most interview aspects were analytical skills, coding skills, technical knowledge, experience, and culture fit. The TIP Club in its current iteration tests for analytical skills and coding skills. With open-ended questions in the survey, it was easy to assess whether the membership felt if it was lacking in other aspects. This was reflected by the theoretical and company-based knowledge categories that emerged from the data.

This project also investigated whether the TIP Club would be into a shared pool of resources. With a unanimous yes (Figure 3), this can be established as a definite interest for the existing membership. A majority opted for GoogleDrive as a shared medium.

From the assessing of resource collection, it follows that the subjects were polled on their experience. From this, we were able to ascertain interview questions posed in the industry, as well as subjective ratings (Figure7) and comments. The results were unique and varied.

Proposed Solutions and Recommendations

Based on the data, the following recommendation ensues. A shared google drive folder should be created to host the lacking resources of the club. The sub-folder hierarchy should be based on the Laakman McDowell model in the collected data. Below is an illustrative mock-up:

Figure 8. Prospective Google Drive

I recommend a slight modification of the proposed model from culture fit to company knowledge. I did this based on the company-based data category that emerged from the question "What could the TIP Club improve on?" The data collected from these questions highlighted a desire for information about the culture but also on more detailed knowledge like interview practices. The analytical skills and coding skills questions that the TIP already cover could be archived within these sections. The theoretical knowledge questions that emerged as part of the ask could be filed under Technical Knowledge. The working experiences, company ratings and interview practices such as the ones collected within the data could be stored in company knowledge. No data surfaced from the surveys about the 'Experience' model of Lakmann McDowell, which related to technical projects, hackathons and other such items one might find under a interviewee's resume. I recommend leaving the folder despite the lack of data supporting a want for this because of the limitations.

Limitations

A few key limitations are at work in this study. The membership is small and the quantity of members who responded to the survey is even smaller. These results provide us with very specific membership wishes at a very specific moment in time. With the high turnover in this sort of club, it is possible that

those wishes might change. It is also why we chose to model the archive with the Laakmann McDowell model. This model is an industry standard so it makes sense to center the wishes of a changing population around a recognizable model. Although there may be many other interesting ways to prioritize the information, this study provides a launching spot to begin resource collection that is requested by the TIP membership and should be able to elevate the caliber of the individual interviewees.

References

McDowell, Gayle Laakmann. *Cracking The Coding Interview : 150 Programming Questions and Solutions.* Palo Alto, CA :CareerCup LLC, 2011.