Improving the Interview Process for Software Engineering Roles

for

John Moore

CEO

JM Media

Vancouver, British Columbia

by

Nick Rutigliano

Software Engineer and Computer Science Student

November 24th, 2017

Table of Contents

[1 Introduction 3](#_Toc499321694)

[1.1 Summary of Standard Interview Practices 3](#_Toc499321695)

[1.2 Issues with the Current Practices 4](#_Toc499321696)

[1.2.1 Issues with the Pre-Screen Stage 4](#_Toc499321697)

[1.2.2 Issues with the On-Site Interview Stage 5](#_Toc499321698)

[1.2.3 Issues with the Offer or Rejection Stage 5](#_Toc499321699)

[1.3 Scope and Methods of the Report 6](#_Toc499321700)

[2 Data Section 7](#_Toc499321701)

[2.1 Interviews with Software Engineers and Interviewers 7](#_Toc499321702)

[2.1.1 Chris Zhu, Software Engineer, Airbnb 7](#_Toc499321703)

[2.1.2 Huzefa Fatakdawala, Software Engineer, CZI 8](#_Toc499321704)

[2.2 Survey Data Analysis 10](#_Toc499321705)

[3 Conclusion 15](#_Toc499321706)

[3.1 Summary of Findings & Recommendations 15](#_Toc499321707)

# Introduction

With the rise of the tech industry in the last two decades, especially as computers have become more and more ubiquitous, the demand for skilled workers in the field has increased year over year. Despite the rising enrollment in Computer Science and Information Technology education programs, JM Media and other tech companies are still having difficulties filling seats with quality employees that make it through the interview process to meet the demands of the fast-moving field. On the opposite side of the spectrum the game involved to pass through the process has led to freezing out people who would be unequivocally competent at the positions but cannot perform the arbitrary tasks involved with the interviews.

What follows in my report is the research of the issues in the interview process itself, followed by a survey with candidates from various technical backgrounds that take up Software Engineering positions, and two interviews with experienced industry engineers. The first interview is with Chris Zhu, a Software Engineer at Airbnb who conducts technical interviews while the second interview is with Huzefa Fatakdawala, a Software Engineer at the Chan-Zuckerberg Initiate who also conducts technical interviews.

## Summary of Standard Interview Practices

The current standard for the process of hiring Software Engineers is usually 3 parts if a company decides to proceed with and move forward following a candidates application.

The first stage is called the **pre-screen** and is used to determine whether the candidate has potential and is qualified for on an on-site (at the place work) interview. The pre-screen usually takes the form as any of the following: an online coding challenge (such as the ones on HackerRank.com), a technical question and answer phone screen, a non-technical question and answer phone screen, or a take home project for the candidate to work on in their own time that would then be discussed over a phone call.

The second stage is the **on-site interview** and takes place at the place of the position. This part is composed of anywhere from 2 to 8 interviews in a day, potentially multiple days, and including but not limited to: whiteboard questions, design questions, and non-technical interviews with both management and potential coworkers.

The third and final stage is the **offer or rejection** period and involves the company informing the candidate of their decision on the hire. This stage can involve feedback following the results of the on-site interview.

## Issues with the Current Practices

Currently there is issues in each stage of the process causing problems for you the employer and the potential candidates for your company.

### Issues with the Pre-Screen Stage

Following the resume and application review, should your company decide to advance a candidate into the pre-screening phase, there is issues with some of the options in place. Online coding challenges have the problems of often being archaic and esoteric questions that have very little reference to the real work of the condition. Also, given the nature of the platforms for these challenges, the candidate will find themselves constrained on resources and time, while these are not always considered a problem, they are not accurate representations of working situations within your company.

Technical phone screens face similar issues as the coding challenges in terms of limited resources and times for the candidates to solve the questions at hand. The questions themselves are at the discretion of the interviewer, and as such can be poorly correlated to the core competencies of the position or the qualifications it requires.

Take-home projects face a different problem in that they can sometimes be strenuous and take a relatively longer time to complete instead of an hour long coding challenge or phone screen. If your candidates have busy schedules then this option may prevent them from advancing the interview process itself due to their limited time to complete the challenge depending on the deadline given for the project itself.

### Issues with the On-Site Interview Stage

After you have approved a candidate to have passed through the pre-screen stage is then that they are invited to the company’s office for the on-site interview.

Issues arise during this for various reasons. When trying to test and affirm a candidates qualifications with whiteboard interviews the problems are exacerbated by the often little to zero reference materials at hand. If the problem given itself is obscure the difficulties are multiplied and the candidate stands little chance of passing the interview unless they have prepared for these arcane questions. This itself is the brunt of the problem for this stage in that this candidate would have now had to forego preparation for the position to instead prepare for the interview in hopes of passing your interview. In solving these questions the candidate has not proven their ability and qualifications for the position at hand, simply for the interview.

### Issues with the Offer or Rejection Stage

Should a candidate have made it this far you are now responsible with deciding whether or not to extend an offer to work for your company or a rejection letter informing them they are not what you are looking for. If the candidate was successful then the interview process is over.

However, should a candidate have not passed, the rejection is now imminent and you must inform them of your decision. Depending on the legality of the interview within your company you may not be obligated or even capable of providing reasoning or feedback as to why the candidate has failed. This presents an unfortunate situation for the applicant as they can be at a loss for what they need to improve upon to land themselves a comparable position.

## Scope and Methods of the Report

This report is intended to improve the overall quality of the interview process for software engineering roles in hopes to both increase the quantity and quality of hires for JM Media as well as create a more positive experience for new potential applicants to the company in the future.

One hundred and ten software engineers from various backgrounds responded to the survey regarding various questions involving the three stages of the process and areas for its potential improvement. In addition I also conducted two interviews with discussions and recommendations from experienced industry experts to find practical and feasible solutions.

# Data Section

## Interviews with Software Engineers and Interviewers

### Chris Zhu, Software Engineer, Airbnb

*Chris Zhu is a Software Engineer at Airbnb in San Francisco, California. He acquired his Bachelor of Science at the University of British Columbia with a Major in Computer Science. He has been interviewed at many top tech companies including Google, Facebook, Quora and now does some technical interviewing alongside his engineering work.*

NR: How do you feel the current processes tests the skills and qualifications of the candidates?

CZ: I think its pretty decent. It could be improved.

NR: Is it interview skills or job skills?

CZ: I think its favoured towards the applicable jobs skills but it depends on the question really.

NR: What kind of questions do you find are good at evaluating candidates?

CZ: I think design questions are pretty good. They force the candidate to kind of draw on all sorts of knowledge and job ability to create something. They aren’t black and white and you can see how they build something from the ground up.

NR: What kind of questions do you find are poor at evaluating candidates?

CZ: Tricky algorithmic problems. They don’t really test the kind of thing you do at work, kind of just academia problems.

NR: What kind of adjustments to the process could improve the quality of assessment for candidates?

CZ: Probably more pair programming work.

NR: How feasible is that?

CZ: Should be feasible for startups and smaller companies, not massive companies.

NR: Is the length of the process sufficient in assessing candidates?

CZ: I’d say its about right.

NR: Are the individual interviews to long/short?

CZ: Probably a little too short.

NR: Do think there is an issue with the current rate of false positives/negatives from hiring?

CZ: It’s hard to say.

NR: Do you think there is any other issues with the process from an interviewer perspective?

CZ: As an interviewer its pretty tedious and boring.

### Huzefa Fatakdawala, Software Engineer, CZI

*Huzefa Fatakdawala is a Software Engineer at the Chan-Zuckerberg Initiative in Palo Alto, California. He acquired his Bachelor of Applied Sciences at the University of Waterloo with a Major in Electrical Engineering. He has been interviewed and worked at many tech companies including Microsoft, Amazon, Arista and Facebook.*

NR: How do you feel the current processes tests the skills and qualifications of the candidates?

HF: It’s alright. It could definitely use some improvements.

NR: Is it interview skills or job skills?

HF: Leaning towards interview skills for sure.

NR: What kind of questions do you find are good at evaluating candidates?

HF: I like practical questions for the people I interview. I don’t care about solving random problems that don’t really matter in our real work.

NR: What kind of questions do you find are poor at evaluating candidates?

HF: One trick questions. The question should be multi-part so giving a hint or solution to one part doesn’t give away the entire thing.

NR: What kind of adjustments to the process could improve the quality of assessment for candidates?

HF: 3 hour pair programming session.

NR: How feasible is that?

HF: Not very since it takes up too many resources. I know Cloudflare does this though.

NR: Is the length of the process sufficient in assessing candidates?

HF: I’d say it’s a little short.

NR: Are the individual interviews to long/short?

HF: Too short as well.

NR: Do think there is an issue with the current rate of false positives/negatives from hiring?

HF: False positives are bad. The cost of making a mistake hire is much higher than not hiring a good person every so often.

NR: Do you think there is any other issues with the process from an interviewer perspective?

HF: Elitism. Many people need to lose it during the interview. You gotta approach the interview as a team effort to come to a solution.

## Survey Data Analysis

Figure 2.2.1 shows the breakdown of the surveyees educational backgrounds.

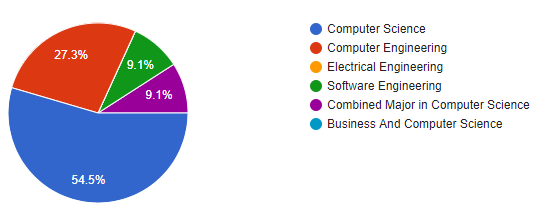


Figure 2.2.1 Education Backgrounds for Software Engineers

The engineers were then asked a variety of questions with interesting results.

To start they were asked a few questions about the pre-screen stage.

Figure 2.2.2 - In regard to the screening methods before on-site interviews, which of the following did you feel best tested your relevant skills and qualifications?

No clear favourite which would lend itself to seem that it is largely preferential which one they thought best showcased them to potential employers.

Figure 2.2.3 - Is there a method that you find worse than the others?

This result was far more interesting than the previous. Despite no consensus on what result best showcased their ability, candidates seemingly agree against online coding challenges.

Figure 2.2.4 – Is there a method that you find better than the others?

This result is also interesting as it shows candidates greatly prefer quality technical phone screens and take home projects to the alternatives for pre-screening methods.

The engineers were then asked questions about the on-site interview stage.

To start they were asked the following “with the current standard process of whiteboard or resource-less coding questions, how did you feel these work at assessing your skills and qualifications as a potential hire?”. The results mildly leaned towards a positive indication that the process did an alright job at assessing their abilities and qualifications. This can be seen broken down in Figure 2.2.5

The next question was open ended: “What kind of questions or approaches align the interview more with the job?”. This came with a myriad of responses including:

* Depends on job. Asking algorithm questions for a front end job doesn't work.
* the same process but with different, more relevant questions.
* Domain specific questions
* Discuss real problems
* General open ended questions
* "Which of these things do you know", "Make me a smaller verison of our product", "what have you done before like our product"
* Rather than general knowledge of coding, asking more domain specific questions. Knowledge of the language and best practices. Understanding codebase and structures.
* Software deisng questions, rather than algorithms and data structures
* Questions with implications of being relevant to the job or a problem rather than a very abstract problem with no meaning explained.
* Related questions to the job
* Scenario questions, real-world programming questions

When asked whether “the current process rewards preparation for the interview or for the job?” the candidates were once again very split between moderately interview leaning and moderately job leaning with no real consensus.

Finally the engineers were asked about the offer or rejection stage.

When asked “How do you feel about the current feedback systems in place for the interview process?” the responses are also interesting.

* Non existent due to legal reasons. Very poor feedback experience.
* nonexistent by policy
* There are none
* Could be better
* There's barely any, very opaque
* Extremely poor
* There isn't really much of a feedback system. Often times you won't hear back any results from a company, or if you do you'll only hear accepted/rejected, not why.
* no real feedback is given
* Poor. Of the feedback systems, I am not even sure people use them or read the responses.
* Terrible, usually an auto-generated response that doesn't go into more detail than "We thought you did great but there were even better people applying"

This question was then followed up with “Is there something you felt was missing in the communication following the interview process, with the company and/or the recruitment team?”.

* Disconnect between recruitment and engineering.
* would definitely like more feedback
* Sometimes no feedback at all
* Real feedback about performance
* I feel like there's not enough information on cultural fit
* asdf
* What the team liked about me, what in the process lead from my resume being accepted/rejected, to my interview going well or not
* Big companies don't really ask about portfolio, projects, or past experiences
* I feel like more direct and timely communication of success or failure is important rather than lack of contact when 'failing' an interview.
* Lack of updates if I'm still being considered

# Conclusion

## Summary of Findings & Recommendations

There is no clear consensus for all things considered in this report. However, there is certainly some strong favourites and potential improvements to each stage of the process for your company to implement.

The pre-screen stage should largely avoid online coding challenges if possible. The questions are often low quality and irrelevant to the position. Candidates much prefer phone screens. Should phone screens present an operational difficulty in terms of human resources, take-home projects may be a viable alternative to adequately assess talents before advancing to on-site interviews.

On-site interviews as well as pre-screen questions should avoid esoteric, one-trick, or heavy algorithmic questions. Design and real-world scenario based questions draw more general and quality knowledge from the applicant and allow them to show more facets of their abilities instead of one-off interview knowledge. If possible pair-programming could also allow the candidate to program with one of your employees testing both their problem solving ability alongside their teamwork.

If at all possible, some feedback on the candidate should be they rejected would be nice. Candidates are eager to improve and often without any feedback are anxious to re-apply to companies given they are unsure of what they previously did wrong in interviews.