

Rock the Technical Interview

Eric Giguere

Co-Author, *Programming Interviews Exposed*

eric@piexposed.com

About this Talk

- Practical advice about preparing for the technical interview process at companies like Google, Microsoft, etc.
- These companies follow similar hiring processes and use similar technical interviews, but there may be differences.
 - Always a good idea to research each company!
- No book or talk will guarantee that you get a job. That's entirely up to you!

About Me

- Computer Science graduate from the University of Waterloo (BMath 1990, MMath 1996)
- Professional software developer
 - Computer Systems Group (UW) - 3 years
 - Watcom/Powersoft/Sybase - 17 years
 - Google - 5 months!
- Author or co-author of 5 books, including the 2nd edition of *Programming Interviews Exposed*
 - Search for me on Amazon!

What Do Companies Look For?

They look for **skilled software engineers**

who can **get things done**

and **work independently as part of a team.**

How Do They Find These People?

There are some initial indicators used to screen candidates:

- Do they have or are they working towards a relevant technical degree?
- Do they have good marks?
- Do they have relevant work experience?

By themselves, however, these indicators are **not enough!**

That's why they use **technical interviews** to assess a candidate's skills and capabilities.

What is a Technical Interview?

A **technical interview** (or **programming interview**) is a special type of job interview where a software engineer asks a series of programming questions in order to assess the candidate's skills and capabilities.

A technical interview has nothing to do with:

- Salary & benefits
- Career path
- Anything not related to software engineering skills

Technical interviews are all about determining if the person is a **skilled software engineer** who can **get things done** and **work independently as part of a team**.

Format of a Technical Interview

- Each interview is 30-60 minutes long
 - Expect at least 5, sometimes more
- 5 minutes of introduction/preparation
- 30-50 minutes of technical questioning by the interviewer
 - Candidate stands at whiteboard or can use paper
 - No books, laptops, just you and the interviewer
- 5-10 minutes to ask interviewer questions

What Kind of Questions Are Asked?

There are three broad categories of questions:

- Algorithm questions
 - *Can you figure out how to do something and how to do it efficiently? Can you measure its efficiency?*
- Coding questions
 - *Can you write working, readable code?*
- System design questions
 - *Can you design systems that work together and scale?*

Sample Questions

- Describe the difference between a breadth-first search and a depth-first search.
- Show me how you would code a depth-first search.
- How would you efficiently transfer a tree-based data structure from a mobile phone to a server?
- How would you design a web-based mail system?
- Describe how the UTF-8 encoding works. What are its advantages? What are its disadvantages?
- Write a function that computes the factorial of a number. Now write it again without using recursion.

How to Answer the Questions

- **Make sure you understand the question.** Do not start until you do! Ask for clarification if you're at all unsure.
- **Use the programming language you're most familiar with.** It doesn't matter if the interviewer is unfamiliar with that language. They can probably figure it out. They can consult colleagues after with any questions.
- **Verbalize your thought processes.** Don't just stand there at do nothing while you think about the problem. Talk about what you're thinking, discuss the pros and cons. Think out loud!
- **Aim for a general answer.** The question may be very specific, but if you can generalize the answer you'll get extra points. You may need to answer the specific question first to figure out how to generalize it.
- **If you don't know something, admit it.** The interviewer may give you a hint, or move on to another question.

How to Behave During the Interview

- **Be confident.** You know your stuff, right? You can be nervous about the interview but still be confident about your skills.
- **Never lie.** Don't claim things you know are not true.
- **Don't interrupt the interviewer.** Remember, the interviewer is also assessing how well you can work as part of a team. Being a good teammate means listening to what others have to say.
- **Don't disparage the interviewer, the company, other companies, other people.** Be positive and open about things.
- **Arrive on time and stay on time.** Time is precious!
- **Turn off your mobile devices.** Don't answer a call or text during an interview!

How to Prepare

- Study your computer science textbooks, particularly those relating to algorithms and system design.
 - If you don't have a good textbook, consider *The Algorithm Design Manual* by Steve Skiena. Pricey, but probably the most readable algorithm book available today.
 - Learn about design patterns. Understand sorting algorithms and basic data structures like trees and linked lists.
- Read about the company's culture.
 - For Google, try *In the Plex* by Steven Levy.
- Write some code.
 - Not just your school assignments, which are often simpler than real life coding.
- Do a mock interview.
 - Get someone to ask you some questions in front of a whiteboard.

If You Have Time...

- Update your LinkedIn profile and your resume.
 - Ensure they are complete.
 - Put a special focus on things you *accomplished*. If you can, try to express the *impact* of those accomplishments.
 - Also highlight *teamwork* and *independent work*.
 - Show *career progression*.
- Build expertise.
 - Answer questions (correctly!) on StackOverflow and other sites.
 - Write articles.
 - Start and contribute regularly to a blog.
 - Work on open source projects.
- Sanitize your online profile.
 - Get rid of "bad" stuff.
 - Change the privacy settings.

When the Interview Day Arrives...

- Don't cram. Either you're ready or you're not.
- Clear out your calendar. Don't do anything stressful immediately before or after the interviews.
- Get a good night's sleep. Get *several* nights of sleep!
- Dress appropriately. Suits are not necessary, but you still want to look good.
- If you are sick, try to reschedule the interview. You won't perform well and the interviewers really don't want to see you if you're sick.
- Arrive early. Go to the washroom. Relax.

After the Interview...

- It's over. Time to move on!
- Did you prepare enough? Be honest. If you didn't, do more preparation next time.
- If you don't get an offer, **it's not the end of the world.**
 - Maybe you were having a bad day.
 - Maybe you just can't code in front of a whiteboard.
 - Maybe one of your interviewers was having a bad day.
 - Maybe it's not for you.
 - You can try again later.

The technical interview system is not perfect. Some good candidates will not make it through. But it generally keeps the poor candidates out. Better to have false negatives than false positives!

For more tips visit

<http://www.piexposed.com>