Ace the SOFTWARE ENGINEERING INTERVIEW

An Interview Preparation Framework to Land the Job You Will Love

RYAN YLITALO

Ace the Software Engineering Interview

An Interview Preparation Framework to Land the Job You Will Love

Ryan Ylitalo

ACE THE SOFTWARE ENGINEERING INTERVIEW

DEDICATION

This book is dedicated to every great software engineer who wants to find the job that will be rewarding, fulfilling, and motivating.

Congratulations! You've taken the first step in the process of landing your dream job. This book is full of practical insights that will help you market your skills and experience in a way that will impress your interviewers.

In the 18+ years that I have been in the software engineering industry, I've been on both sides of the interview table numerous times. I have interviewed hundreds of software engineers and played key roles in improving the software engineer hiring and recruiting processes at some pretty large organizations.

I have seen countless blunders made and have noticed a serious lack of preparation on the part of numerous candidates. I wrote this book to provide a framework for interview preparation, so that you are better able to represent yourself when it is time to showcase your talents and skills.

> **The principles in this book should be immediately useful. You should be able to read and digest the contents in 120 minutes or less.**

This book is not meant to be a comprehensive interviewing guide containing every possible bit of information that applies to every unique situation. This book is not intended to be a technological manuscript for your favorite technology. It is intended to provide you with a framework for technical interview preparation.

This book is a summary of key interview thoughts that I have found important. While you should be able to read the entire contents in 120 minutes or less, my hope is that you will find valuable tips to help you in your career journey and job hunting experience.

FOREWORD

I didn't dismantle computers as a child, I didn't write clever programs in obscure programming languages, and I certainly didn't dream of becoming a software engineer. I barely knew the field existed. I first began my career journey as software engineer in the late 90's as a mechanical engineering student at the University of Minnesota. As part of my engineering curriculum, I was required to take a C programming course.

Taking this course opened my eyes to the possibilities. After completing my first programming course, I knew two things. My mechanical engineering aspirations were toast, I was going to be a software engineer. And secondly, I was 'ready' for a real software engineering position. In hindsight, I realized that one semester of C programming didn't make me a software engineer, but it was enough to land me a summer internship at a small software company.

My first job hunting experience was easy. I walked into a local software company, asked for an engineering manager, and got my first interview on the spot. We talked about my school experience; I coded up a string copy function in C, and walked out with a summer internship.

Almost twenty years later, reflecting back on that first job hunting experience, I realize that not all job hunts are created equal. Looking for a summer internship in the heart of the tech boom was a significantly different experience than looking for a permanent engineering job after the bursting of the tech bubble in the early 2000's. The ebb and flow of the tech industry has provided a wide ranging set of job hunting experiences. The proper preparation is a critical component of a successful job search.

I hope to pass on what I've learned through my own and other's mistakes, so that you can avoid them when searching for your next software engineering position.

CONTENTS

INTRODUCTION

Have you been contemplating a job change? Are you ready to begin the interview process? Is this your first interview experience? Perhaps you have been through this process multiple times. Are you wondering if there are things you could do that would help create a better impression and improve your chances of success?

During the course of this book, I will outline a framework that will help you understand the ideal time to change jobs, provide guidance on which organizations to seek out or avoid, and then guide you through the preparation and interview process.

I have intentionally kept the format brief so that you can get through the material quickly, and so that important points are not lost in pages of fluff. The book is ordered in a chronological fashion according to the typical interview process, but each section is self-contained and can be referenced in any order you prefer.

Let's get started!

1 - SIGNS YOU SHOULD CONSIDER SWITCHING JOBS

How do you know when it is time to switch jobs? There is not always an obvious answer to this question. There is opportunity cost associated with every decision you make. You certainly do not want to end up with a less desirable job than you already have. You also need a structured process for determining when you should switch.

Job switch timing depends on numerous factors that are unique to your situation. In the section below we will discuss some factors that will help you decide when it is the right time to change jobs. You may not find all of these factors relevant to your situation, but they should provide food for thought when you are considering a change. You should add factors to this list as you feel appropriate.

No Longer Challenged

This is one of the most important reasons that you should switch jobs. If you are no longer challenged, you will soon become bored. If you stay for too long in an unchallenging environment, you will become robotic and disengaged. Unless you are looking for a job to coast into retirement, you should seek out a position that is going to challenge you and keep you interested in your career choice.

Not Learning New Things

If you are not learning, then you are becoming obsolete. You do not want to wake up one day without a job, and a set of skills that nobody wants. Learning new things makes you more valuable, gives you negotiating leverage, and provides you with more security. Who wouldn't want that?

Not Surrounded by Elite Talent

You will find that you perform better, learn more, and are more motivated when working with people smarter than yourself. Your time and ability to learn is finitely constrained by numerous factors. You should place yourself in situations that maximize your learning potential.

Not Earning More

Have your earnings stagnated over a period of two or three years? This may be an indication that you are not growing professionally. Maybe you are not learning new things, or you are not taking on new responsibilities. Whatever the

reason, it is typically an indication that your career may have stalled, and it may be time to seek new opportunities.

Toxic Environment

If your boss is unfair, your co-workers are hostile, or you just do not enjoy going to work anymore, it may be a good time to move on. There are plenty of organizations that can provide you with a positive culture. While you may think that you are able to handle a toxic environment, you will eventually become unmotivated and your career will suffer as a result.

Software Engineers are Treated as Commodities

You do not want to work in an organization that does not value the true contributions of a great software engineer. These organizations typically do not hire top engineers. When they do have top engineers on staff, they will not stick around long. These companies are often laggards on the technology adoption curve, and their code base is not pretty. You will prefer to work in organizations where engineers take pride and ownership in what they do.

Culture Does Not Fit

Organizational culture is not always static. Sometimes your life situation changes in ways that make certain cultures less desirable. Maybe you no longer want to continually travel due to your family obligations. Maybe you have grown tired of large company politics, and would like to work at a small startup. If the shoe no longer fits, you should begin seeking an organization that is more culturally suited to your current life situation.

Not Using Strongest Skills

You might be doing your job well, but if you are not leveraging your strongest skills, you might want to consider a new position. You probably prefer to spend most of your time doing the things that you do best. If you are not put in situations that leverage your best skills, you may become frustrated by your performance and your situation.

However, you may be trying to transition out of an area that you excelled in previously and are developing new skills and competencies. In these situations, try to think of ways to incorporate your strongest skills to help you during your transition.

Receive a Better Offer

A better offer is not just about money. It could be a position that offers the opportunity to learn new skills, take on new responsibilities, or work on things

that you personally find more fulfilling. If you get an offer that is better overall than your current position, why wouldn't you take it?

Lack of Passion

If you are not excited to go to work, you should conduct a serious situation evaluation. Life is too short to spend significant time doing things that you do not enjoy. I was once asked during an interview what I would do if the job became boring. I told the interviewers that I would move on to a place that had more interesting challenges. I did not get an offer.

When discussing with the recruiter later, he said the company really would have liked to offer me the position. However, the hiring manager was not comfortable with my remark about moving on. Apparently they were looking for somebody who would be willing to stick around for a number of years once the new product was in maintenance mode. After hearing this feedback, I realized that the position wouldn't have been a good fit for me.

Summary

Develop your own list of things that are important to you. Use it as your job switch decision making framework. Don't be lulled into complacency when you realize it is time to move on. Staying at an organization that no longer fits you is much costlier than you may realize. Dust off your resume, and start searching. In the next chapter we will discuss the characteristics of organizations that you may want to avoid.

2 - SIGNS OF A POOR ENGINEERING ORGANIZATION

You have developed your list of indicators that will trigger your next job search. You know a number of organizations will not be a good fit for you. How do you identify these companies so that you can avoid them?

In this chapter we will discuss a number of signs that may indicate an organization should be avoided. None of these factors are fool-proof, but be aware of the risks when accepting a software engineering position at a company that exhibits any of the following characteristics.

Non-Technical Engineering Managers

It is not necessary for engineering manager to write code. However, a manager should have enough technical knowledge to understand the scope and scale of a particular task that an engineer is working on. They should be able to have intelligent conversations about technical tradeoffs because these types of decisions are regularly made on a development team.

Additionally, if your manager does not understand the most basic technical concepts, it is likely that your annual performance review will depend more on your marketing skills than your ability to code.

Technology Adoption Laggards

As the rate of technology adoption continues to increase, the line has blurred significantly between a traditional brick-and-mortar business and a technology business. When Amazon was first created, big-box retailers didn't even notice. The thought that an upstart online book retailer would become their primary competition certainly wasn't the prevailing concern.

Eventually these large retailers discovered that even brick-and-mortar businesses need to embrace technology to stay relevant. The traditional sales model had been disrupted by a number of technology-based organizations who found new and innovative ways to deliver a better customer experience. Companies that didn't embrace technology were overrun by the competition.

You should avoid any company that is not embracing technology as a way to serve customer needs. It will save you the hassle of having to look for another job when that company goes bankrupt.

It’s a Sinking Ship

If a company is consistently losing market share, you are better off not joining up. These environments are usually budget-constrained, overly stressful, and just generally toxic. You should find a healthy company in a healthy market segment. A company that is growing market share will provide you with many additional opportunities that will not be available in a company that is shrinking.

Lack of Employee Growth

When you take a job, it is not just about what you can do for that company. It is also about what that company can do for you. You need assurance that you will be able to grow your skills, knowledge, and expertise in a way that will be beneficial to both you and your company. If a company is not willing to invest in your growth, you should not be interested in investing your time with that company.

Hiring for Skills, Not for Competencies

Some companies will hire you based on what you know instead of what you can accomplish. You don't want to work at these places. If a company doesn't understand the distinction between knowledge and accomplishment, they will be unable to differentiate between a decent engineer and a great one.

While there is certainly a correlation between knowledge and productivity, the relationship is not transitive. I have worked with some incredibly knowledgeable engineers who weren't that productive. Usually, it was a competency issue. Some lacked motivation; others lacked problem-solving abilities.

You don't have to hide behind your skill set. You want to be recognized for your ability to accomplish great things and for adding significant value to your organization.

Bad Company Reputation

A company's reputation is usually the result of an organizational leadership culture that originates from the top down. If a company has a bad reputation, it is usually not without merit. You can find groups that rise above a poor organizational reputation, but the positive culture in these areas often degrades as soon as one key leader leaves the group. There are plenty of companies with stellar reputations, so you should not waste your time with companies of ill repute.

The Perpetual Job Opening

There are usually two reasons that organizations keep a job requisition open perpetually. One is that they have been experiencing a sustained growth trend, and have been increasing staff for a significant period of time. This is an incredibly positive scenario. The other reason is high turnover within the organization. The company is continuously trying to keep up with employee attrition.

Keep in mind the size of the organization. Companies with a large engineering staff will often maintain an open requisition to keep the candidate pipeline full. Because of these factors, this indicator can be unreliable. However, if a position has been open for a long period of time, you should attempt to determine the reason before accepting an offer from this company.

Summary

While this is not an exhaustive list of reasons to avoid a company, you should consider the factors outlined above. In your quest to land a rewarding position, don't settle for less than the best. If you use these items as red-flag indicators, you will be well-positioned to avoid companies that are not going to enhance your professional growth.

3 - SIGNS OF A GREAT ENGINEERING ORGANIZATION

While it is good to know which companies should be avoided, it is even more important to have a set of criteria that will help you identify great companies. Some of the signs may vary based on your specific requirements, but the following are some generalizations that are helpful when trying to determine the level of engineering maturity within an organization.

High Bar of Entry

You should expect challenges during the interview process. You should also expect to be asked to code up a solution to some problem. A company that hires engineers without evaluating their coding skills is automatically suspect. You should be looking for companies that ask you to solve difficult problems during the interview process.

Be aware of the difference between hard problems and hard knowledge. Being asked about an obscure nuance of a particular language is not a hard problem, it is hard trivia. It is more important to be able to solve hard problems than it is to know answers to programming trivia. Generally, the more complete the evaluation during the interview process, the better you should feel about potentially accepting an offer.

Automate Everything

There is nothing quite as inefficient as repeatedly doing manual tasks that could easily be automated. When seeking job opportunities, you should seek organizations that have automated every possible aspect of development, because that means you will be spending less time doing boring and repetitive tasks that don't help you grow and learn. Instead, you will be solving challenging problems that will enhance your skills and expertise.

Thought leaders in Organization

Having industry experts and thought leaders in an organization is a great way to attract elite talent. These people can usually work at whatever organization they choose. By virtue of accepting employment at a particular company, they are providing that company with a tacit endorsement as a great workplace. Why wouldn't you want to get hired at a place where you can learn from the best in the industry?

Clear Testing Strategy

If a company does not have a clear testing strategy, it often indicates a lack of customer understanding or a lack of engineering maturity. A lack of customer understanding probably indicates lack of market share. A lack of market share probably indicates lack of funds. You don't want to work at company that doesn't understand its customer needs.

Additionally, companies without clear testing strategies are going to spend more time reactively fixing issues instead of proactively adding value with new customer features. You will find this type of environment mentally draining and demotivating over time.

Willing to Invest in Employee Development

As an engineer, you want to keep your skills up-to-date. That may mean attending a conference, using a new tool that helps you be productive, or taking an engineering course of some type. When you identify areas to improve your skills in a way that will benefit your company, you should expect your company to willingly invest in your growth and education.

Companies that don't realize the benefits of investing in employee development don't achieve that same level of success as companies that do. As such, you should restrict your job search efforts to organizations that are willing to invest time and money into your growth and development.

Appropriate Tools

When you are trying to get work done, you want to use the best tools available. Working at an optimal efficiency sometimes means investing money in engineering tools. If you are continually wasting time by using inefficient tools, opportunity cost is incurred by both you and your organization. You will not learn as much because you will be spending unnecessary time completing manual tasks. To be fully productive, you should spend almost no time on tasks that could be automated in some way.

Growing Customer Base

Life is always better with a growing customer base. There is a positive vibe, and employees feel a sense of security. Teams are energized and feel valued because they can see that their efforts are paying dividends. You will find that there is more money available for investment in employee growth or employee salaries. You will also find that there are more opportunities to take on additional responsibilities and enhance your skills as the company scales up to meet its growing customer demands.

Summary

Keep in mind the characteristics of great companies when looking for your next position. Use the ideas discussed in this chapter to help you create your own checklist of '*Great Company Characteristics*'. Add additional categories that you feel are appropriate indicators. Rank each company on a scale of one to ten in each of the areas on your list. This checklist will help you make relative comparisons when analyzing multiple companies.

4 - SIGNS OF A GREAT SOFTWARE ENGINEER

You have identified the company or group of companies that you would like to work for. You now need to develop a strategy for convincing one of these organizations that you are the best available candidate. This requires that you understand the common characteristics of great engineers. Figure out which of these traits you possess and then be prepared to sell yourself on these traits during your interviews.

In the following sections of this chapter, we will discuss a number of characteristics common to the top engineers within the industry. The more of these traits you possess, the more you will have to speak about during your interviews.

Expert Problem Solver

You not only solve the problems that you are tasked with solving; you solve every problem you encounter. You look for problems to solve because you love the challenge. Problems aren't something to be tolerated, but something to be embraced, cherished, and enjoyed. No problem is too small or too large. You don't think of problems as theoretical exercises, you will solve the problem within the real world constraints of budget, time, and resources.

Intrinsically Motivated

You don't need anything to keep you motivated. You develop and build software because you love what you do. Finding solutions to tough problems is reward enough for you. Innovation, creativity, and resourcefulness are common to you as you navigate environmental constraints to accomplish your objectives.

Design and Architecture Skills

You can design and architect a complex system or you can code up individual components. You enjoy doing either. You build performance, scalability, reliability, and security into your design and implementation because you realize that non-functional requirements are as important to the customer as features and functionality.

Pride in Your Work

You take pride in what you produce. You are not satisfied with '*It works!*" For you, each line of code is part of your personal brand, and you make sure you represent yourself well. However, you are still humble enough to accept input

and suggestions when others evaluate your work.

Great Self Awareness

A big part of what makes you great is your ability to know what you don't know. You know what your weaknesses are, and you try hard to improve them. You encourage and expect constructive criticism from your managers and peers. You learn how to delegate in areas that will never be your strengths. You know your strengths and you leverage them in every way possible.

Great Technical Passion and Awareness

There isn't a technology existing that you haven't heard of. You are constantly checking out new technologies, and trying to learn how they may be leveraged in your current environment. You code up proof-of-concepts, you read blogs and technology newsletters, and you attend conferences. You have an insatiable appetite and hunger for technologies and tools that can make your job easier or more efficient.

You love to code. You think in code. You write games, you build apps, you contribute to open-source. A day without coding is a day wasted.

Your motto - "I've never found a problem a little code won't solve."

You Get Along Well with Others

Gone are the days when an engineer could hole up in their cube and only come out for food. Collaboration is essential to be a modern engineer. Your ability to collaborate and get your team working well together is part of what makes you great. You are often sought out by your peers for advice and solutions. You cultivate relationships that will help your team be productive and effective.

You Get Stuff Done

Not only can you talk theory with your colleagues, you have a track record and history of accomplishing great things. You don't need to represent yourself in terms of skills (even though you've got a lot of them), you talk about accomplishments. Big accomplishments, game changers, things that have gotten you recognition within your organization. While you are known as a knowledgeable resource, you are recognized even more for your ability to get things done.

Summary

Analyze the list of great engineer traits above. Add other items to the list as you think of them. Evaluate yourself against each of these traits. Prepare talking points about each of the traits you possess. Practice talking through them so that

you are comfortable articulating during your interviews.

It is not enough to tell your interviewer that you possess these traits. Be prepared to provide specific examples of how you exhibit them. Use these examples as a way of demonstrating how you can provide value to an organization.

5 - THE SOFTWARE ENGINEER EVALUATION FRAMEWORK

It is important to understand the characteristics that are commonly exhibited by great engineers. You should be prepared to talk about the characteristics you possess during the interview.

However, this won't be the only criteria that you will be judged on. In this chapter, we will discuss two primary areas that provide the decision making framework for software companies looking to hire. Possessing this awareness will help you be more prepared and less likely to be surprised by the various types of questions you may encounter during your interview experience.

Keep in mind that this framework may vary widely between organizations, and between individual interviewers within an organization. However, if you remember that you can usually break down an assessment into skills and competencies, you will be better equipped to make a great impression.

Skills

Skills are the most commonly used dimension for candidate evaluation. They are usually the most heavily weighted during the interview process. Skills are the ability to do something based on your knowledge. Java programming, SQL, design patterns, algorithms, data structures, regular expressions, HTML, JavaScript, CSS, public speaking, etc. A typical job description contains long lists of required skills.

You will likely be asked to write code during the skills assessment portion of your interview. You will also likely be asked a litany of questions regarding various technologies, tools, and languages that are used within the engineering organization.

Don't be surprised if you encounter an experience in which you are drilled exclusively on skills for the entire course of the interview or interviews. While focusing exclusively on skills doesn't necessarily yield the candidate who is the best overall fit, organizations with less mature hiring practices may not delve into your competencies and characteristics. However, characteristics and competencies can help determine how well you will fit into the team and organization.

Competencies

You can expect that if you are interviewing with a company that has mature hiring practices, you will be expected to demonstrate a core set of competencies. These competencies vary by organization, and typically align with the company's mission statement and culture.

Startup companies may look for engineers who are flexible, adaptable, willing to work on diverse projects, and eager to learn new things. A large company looking to hire an engineer to maintain a legacy product line for a well-established customer base may be looking for an expert in a specific technology who can work in a well-defined, tightly-constrained environment that doesn't allow for a whole lot of creativity.

Competencies are typically more abstract, and harder to objectively measure. Competencies are often vetted in a behavioral style interview. They may include things such as teamwork, communication, influence, leadership, judgment, problem-solving, creativity, ingenuity, resourcefulness, passion, and integrity.

> **If you've never participated in a behavior interview, be prepared to give examples of how you behaved in specific scenarios in a way that demonstrates how you exhibit a certain competency.**

Since competencies are somewhat more abstract in nature, they are also more difficult to assess. Additionally, not every organization has formally defined what core competencies are important or required. For these reasons, it is possible that you may not end up being assessed on your competencies.

Summary

It is very important that you understand what specific competencies and characteristics a company is looking for when you interview for a particular position. Even if you possess the skills required of the position, be wary of accepting a position that you feel is not going to be a good fit for you personally.

6 - THE INTERVIEW PREPARATION CHECKLIST

Often when I am conducting interviews, I have been surprised by the candidate's lack of preparation. Preparing for an interview consists of more than just filling out an application for a particular position. Being fully prepared will allow you to concentrate on what is happening during the interview instead of thinking about the answers to the interview questions.

You will find that interviews are going to be time-consuming, intense, and stressful. However, the more you prepare, the less stressful you will find the interview experience. In this chapter, we will discuss a number of steps that will help you be more prepared and help keep your next interview as stress-free as possible.

Dress Appropriately for the Position and Company Culture

Be aware that dressing appropriately depends on the culture of the company. Conventional wisdom will tell you to dress up. While this is typically good advice, I once was rejected as a candidate for a technology startup early in my career because I wore a suit coat to my interview. Immediately upon entering the interview, I realized I was way over-dressed. Make sure you consult with your recruiter when trying to determine what level of dress will be the most appropriate for the position and the company.

Research the Company

How will you know if this is your dream job if you don't know anything about the company and the culture? You should not go into the interview expecting that the interviewer will be able to answer every question you have. Do not ask questions that could have been easily answered by perusing the company website or with a simple Google search. This will make you appear as though you are unprepared and lack interest.

If you have any contacts within the organization, reach out to them and get as much information as possible. Search the various website and forums that provide company reviews. See how the company interacts with customers on various social media sites. These interactions can tell you volumes about a company’s culture.

Remember to research your interviewers prior to the interview. A quick

search on LinkedIn can give you a feel for an interviewer's background. You can use this information to anticipate questions you may receive. It can also help an interviewer relate to you if you focus on a few technology areas that you have in common.

Know What Is On Your Resume

While I should not have to mention this one, I continually encounter candidates who have added skills that they don't possess. It makes for an awkward moment when I ask them to talk about one of these skills. Make sure you are prepared to speak to anything that is on your resume. Don't be tempted to add skills which you really don't have. Any fabrication on your resume will result in automatic disqualification.

Be Prepared to Articulate How Your Skills Provide Value

It's good to enumerate your skills and expertise during an interview. However, this usually is not going to be enough to land your dream job. You need to be prepared to talk about examples of how you have used your skills to accomplish specific notable achievements. You will need to demonstrate how those achievements added value to your employer. You will also need to be able to articulate how those skills can translate into value in the position you are interviewing for. It's not enough to talk about skills and tasks; you need to know how you can contribute to the company's annual profits.

Know your Strengths and Weakness

I have often been surprised by how many candidates aren't able to speak well about their strengths and weaknesses. You should know yourself better than anyone. If you aren't able to speak about your strengths and weaknesses, it speaks to your lack of preparation.

Prior to the interview, you need to list three or four strengths and weakness. Think about how you are going to speak to those things when asked. Make sure you talk about legitimate weaknesses, but be prepared to explain how you've been able to overcome them. Be prepared to articulate how valuable your strengths are, and how your strengths set you apart from other candidates.

Make Sure You Know What You Are Looking for in Your Next Position

If you think that you are just looking for more money, then you need to reconsider. More money is not going to keep you motivated or excited about a new job for very long. You need to understand what things intrinsically motivate you, and then look for positions that allow you to do these things. Make

sure you ask enough questions during the interview to determine whether the position is going to be a good fit for you.

Practice Makes Perfect

You need to practice prior to the interview. Conduct mock interviews with your friends or family. Conduct mock interviews with yourself. You will be surprised by how silly you look asking and answering questions to yourself in the mirror, but you will be more surprised by how much better prepared you will be.

I have sometimes interviewed at companies when I have not been actively seeking a new position. This has allowed me to investigate a position without being too emotionally attached to the company. Having an active profile on LinkedIn will allow you to passively investigate opportunities at your convenience.

If you choose to interview as a passive candidate after being contacted by a recruiter, you will find that these interviewing experiences are great practice. They are not as stressful when you know that you don't need a job. And it is possible that you might find an opportunity that is better than the one you currently have.

Summary

Preparation is one of the key success factors when you are trying to land your dream job. Make sure you know what type of position you are seeking. Be prepared to talk about how you can provide value in the position that you are interviewing for.

Make sure you practice prior to the interview. This will help you become more comfortable talking about your skills, experience, and competencies. Having a plan prior to the interview will make you more relaxed during the interview process and will help you perform better during the interview.

7 - THE INTERVIEW PROCESS OVERVIEW

So now that you have taken a number of pre-interview preparation steps, let's discuss the various interview types and formats. While the interview process is going to vary by company, most interviews fall into one of a few different categories. In this chapter, we will discuss the most common interview types, and what you should expect to experience in each.

Initial Technical Screen

The initial technical screen will typically take place over the phone. You will likely be asked questions that are focused on your experience with specific technologies and frameworks. You may also get questions on software engineering concepts such as algorithms, data structures, design patterns, architecture, database technologies, etc. You may also be asked to code up a solution to some engineering problem using one of the many interactive web-based collaboration tools that are available.

These interviews typically range in length from thirty minutes to one hour. You need to be prepared to talk about your experience, but you also may be asked a number of skills based questions that give the interviewer a very basic understanding of your skill set. It is unlikely that you will be asked an in-depth problem-solving question as there usually won't be enough time during the initial technical screen.

Onsite Interviews

Onsite interviews vary in style and length, but they usually fall into one of three main types. Typically, you will have a number of different onsite interviews where each fits into one of the formats described below.

Job Fit or Skills Based Interview

Generically speaking, you may find that this interview tends to be most similar in nature to the initial technical screen. You will probably experience a number of questions around various technologies that help the interviewer get a feel for your knowledge in relevant areas. Depending on how formally the company tracks and records questions asked of candidates, you may end up answering questions that are similar to the ones you faced during the technical screen.

You may be asked to review a code sample, or code up a solution to some problem on a whiteboard. You should be prepared to talk about how your

knowledge and experience will add value in the current position. Make sure that you know everything that is on your resume, as you may be expected to speak to anything that is on it.

Behavioral Interview

Behavioral interviews are typically used to assess candidate competencies. You should be prepared to provide specific examples of how you have demonstrated a number of different competencies in your past positions.

You may be presented with a number of hypothetical situations and then asked how you would respond. You will often be asked to provide examples of how you behaved in similar situations. The topic areas are often related to competencies such as teamwork, employee relationships, motivation, judgment, character, integrity, and other areas that are outlined in ***Chapter 10 - Essential Software Engineer Competencies.***

Case or Problem-Solving Interview

A case is a real-world problem that you will be expected to solve. There are a couple of ways that case interviews are conducted, but both methods will require you to write code. Typically, a single case will take the entire allotted interview time. You won't be asked questions about your resume. You probably aren't going to be asked about specific technologies. The interviewers are attempting to assess your problem-solving abilities. You will need to demonstrate how you are able to apply your technology skills to come up with a solution to a real-world problem.

Some companies will present you with a case with instructions to code up the solution prior to the interview. In these situations, the case interview will involve the dissection of your particular coded solution. This will give the interviewer insight into your thought process, your coding skills, and your software design and architecture abilities. Be prepared to walk them through any algorithms that you have used and how you were able to arrive at your solution.

Other companies may present you with a case at the beginning of the interview. In these types of scenarios, be prepared to verbalize your solution as you are thinking through it. The interviewers are not going to expect you to come up with a solution immediately. Make sure you ask any clarifying questions when any of the requirements are ambiguous. When you start crafting up a solution, talk through each component. This will help the interviewer understand your thought process. A good approach is to start with a naive or

brute force solution, and then iteratively optimize the solution until it is in a state that you feel comfortable presenting.

Case interviews are one of the best ways to get insight into your problem-solving thought process, so be sure to provide as much information as you possibly can. Remember to verbalize everything. The interview may also contain significant dialog between you and the interviewer. If you need time to think, let the interviewer know that you need of couple of minutes. The interviewer will also be assessing your collaboration skills. You will need to keep them engaged as you work through the problem and solution.

Summary

While the style of interviews will vary, the main reason for each interview is to help the company determine whether you are going to be a good fit for the organization. Remember that the assessment goes beyond just skills, and that each interview type will be assessing different dimensions of your overall abilities and cultural fit for the position.

8 - HIRING MANAGER PERSPECTIVES

Have you ever been interviewed by two people for a single position, and then wondered if somehow there were two positions open? You will find that this experience is not uncommon, even in organizations with mature hiring practices.

These inconsistent experiences often are the result of managers with two differing outlooks when hiring engineers. One group will evaluate you on your ability to be immediately productive working within the confines of the current technology context.

The other group will take more of a long-term outlook. They will try determining if you have the ability to quickly learn the necessary skills required of the current and future technology context. You can usually tell a manager's approach based on the type of questions they are asking you. In the sections below, we will discuss some strategies for dealing with each of these types of hiring managers.

Near-Term Outlook

If you interview with a manager who is looking for immediate productivity, they will be looking to assess how your skills match their specific needs. These types of managers will often quiz you on all your various technology experiences. They often don't conduct as thorough cultural fit assessments. They may not spend time looking into competencies and characteristics that will help the team and the company long-term.

You will have to convince these managers that your technology-specific skills are going to be sufficient to be productive immediately. There won't be any harm in speaking to your competencies and characteristics, but skills will be the focus of your conversation. Make sure to demonstrate how well your current experience and skills match up with the job description.

Long-Term Outlook

Hiring managers who are looking to take a long-term view on what you can bring to the team and the company will have a different focus. While these managers will still be looking to assess your experience with relevant technologies, they will spend more time trying to identify other traits that are indicators of long-term success in the position.

Long term-outlook managers will be very interested in your ability to solve

problems, the pace at which you learn new technologies, your ability to make sound judgment, as well as collaboration and communication skills.

With these types of managers, you will still have a good chance to land an offer even if your experience is a bit light. You will need to demonstrate that you have an accelerated learning curve compared to your peers. You should be prepared to give specific examples of different accomplishments that will help them learn about your various competencies.

Summary

Remember that short-term outlook managers will be focused on skills. Long-term outlook managers will still conduct skills assessments, but they will also be very interested in your competencies and characteristics that make you a good fit for the position. You should remember to tailor your approach accordingly based on the hiring manager type.

9 - ESSENTIAL SOFTWARE ENGINEERING SKILLS

Your interviewer's focus on skills will depend on their approach to productivity. Are they more interested in what you can accomplish immediately? Or are they more concerned with what you can accomplish long-term? Regardless, you will need to demonstrate that you possess a set of core engineering skills and knowledge.

Not every engineering position requires that same set of skills. However, you should be able to speak intelligently about most of the topics outlined below. You should use these topics as a guideline to help you prepare for your next interview.

We will not discuss the basics of the topics below, as there are countless books and articles available on each of them. This section is to help you understand if you have deficiencies in your core engineering skill set. If you feel that you are lacking in any of the areas below, you should review these topics prior to your interviews.

Data Structures

You should have learned this in college, and you likely have a good understanding of this topic. However, I am continually surprised by candidates who don't have even a basic understanding of some of the simplest data structures. You should be prepared to talk about stacks, queues, heaps, maps, and trees. You should also be prepared to explain how you would implement any of these data structures in the language of your choosing.

Big 0 Notation

If you are not familiar with runtime-complexity, brush up on this topic. Some positions may not require this knowledge. However, if the position involves sorting and searching large amounts of data, this knowledge is critical. You may not end up getting any runtime-complexity questions, but if you are asked about it, you'll be expected to know what it is.

Searching and Sorting Algorithms

Similar to Big 0 Notation, you may not be asked about searching and sorting algorithms. However, if you are asked, you will be expected to know common algorithms and their implementations. If the position does not involve much

searching and sorting of large data sets, a basic knowledge is probably going to be sufficient. If the position does require searching and sorting of large data sets, you should have an expert level of algorithm knowledge.

Object-Oriented Design

You will need to understand the following five object oriented principles as explained by Robert C. Martin. Single-responsibility principle, open-closed principle, Liskov substitution principle, interface segregation principle, and dependency inversion principle. If you don't understand what these are, go research them and understand why they are important.

Design Patterns

When I interview engineers with numerous years of experience, and they don't understand the most basic design patterns, it's a red flag for me. Most managers will be looking to hire software engineers, not coders. Almost anyone can make a piece of code work.

Design patterns are fundamental to good software design. If you have not taken the initiative to learn them after numerous years of software experience, your motivation and interest in your career choice will be questioned.

Some managers may cut you slack if you have less than five years of experience, because the level of abstraction can make it difficult for junior engineers to grasp the concepts. However, if you been an engineer for numerous years, you should be a design patterns expert.

Recursion

While recursion is a concept that you probably learned in college, it is not always easy for junior engineers to understand where, when, and how recursion should be used. However, if you are a senior engineer, you need to be very comfortable coding up a recursive solution to a problem. If you have not used recursion in some time, you should work through a code sample to review this prior to your interview.

Dynamic Languages

You should understand the differences and similarities of dynamic and statically typed languages. You should understand what closures are and how they can be used. Even if you don't have experience with dynamic languages, you should have a basic understanding of their advantages and disadvantages. There are numerous dynamic languages that are used frequently in modern development, and it is important to know how they are beneficial.

Database Technologies

You need to possess a solid understanding of relational database concepts and SQL. You should be familiar with query performance optimization techniques. You need to understand the benefits and drawbacks of database indexes. You also should possess basic understanding of the various types of NoSQL database technologies; key-value stores, column-family stores, document databases, and graph databases.

Client-side Frameworks

While the client-side frameworks available are numerous and constantly changing, it's important to know the key concepts of the "*client-side framework of the day*". Even if you don't do a lot of client-side development, you should understand the basic features, advantages, and disadvantages of the most common frameworks. There are numerous popular JavaScript frameworks available, and each has a distinct set of pros and cons.

Thread Concepts

Even if you don't have experience with multi-threaded applications, you need to understand some basic threading concepts. A number of languages have rather extensive threading libraries. You should learn these libraries and be able to speak to them. You also need to understand the basics of synchronization, deadlock, and race conditions.

Application Security

With the ubiquity of cloud-based systems, almost every company is vulnerable to a data breach. You should remember "*defense-in-depth*". Be prepared to speak conceptually on the various protections that can be added to make data more secure through each layer of the application.

You don't need to be a security expert, but you may be asked a number of conceptual questions on this topic. You may not need to know all the details of system architecture security. However, you should be able to speak to basic application security concepts and mitigation techniques. This includes cross-site scripting, cross-site request forgery, injection vulnerabilities, and insecure direct object references. Research the latest OWASP top-ten list, and remember how each of these vulnerabilities can be mitigated.

Software Architecture

The more senior the position, the more you will be expected to know about software architecture. You should understand the basics of common application

frameworks. You should understand the components of various integration strategies. The topic is vast, and is often going to be specific to the technologies used by a particular application.

Web Services

You should know and understand the latest technology trends in the area of web services. You should be able to clearly articulate the advantages, disadvantages, and differences between SOAP and REST service architectures. You should understand the fundamentals of the micro-service movement. You should understand the integration benefits of a public web service API.

System Architecture and Scalability

There are not a lot of self-contained applications these days. Most applications interact with numerous disparate systems, and it's important to understand conceptually how these systems interact. You should know how to build a system that scales-up as the system load increases. You should understand how cloud-based architectures can be leveraged to accomplish resiliency and scalability within an application.

Big Data

Entire careers are built around big data technology. Even if you're not going to be a big-data scientist, you should have some idea about the trends within this industry. You may not be asked much about big data, but you should be able to speak to general concepts and understand why it is important.

Reflection

While not every language supports reflection, you should be aware of use-cases where it might be useful to extract class meta-data. Many common frameworks make extensive use of reflection. You should be able to speak to advantages, disadvantages, and the appropriate contexts where reflection might be a good fit.

Performance Diagnostics

There are numerous technology-specific tools that can be used for performance profiling. At a minimum, you should know which tools can be used for performance diagnostics in your particular technology. You should also have a basic familiarity with the features of the most common performance profiling tools available to your industry.

Test Automation

Manual testing is labor-intensive and costly. You should understand the various test automation frameworks and strategies used within your technology stack.

This includes unit, integration, and client-side testing frameworks. Most technologies contain fairly advanced testing frameworks, and it is critical that you know how to leverage them.

Technology Specific Skills

While the list above is focused on generic concepts, there will be numerous frameworks and technologies that you will be expected to know and understand. This will vary by your technology stack, and will be constantly changing. You will be expected to go deep into the technologies you have experience with. You should also be able to speak at a high-level to related trending technologies, even if you don't have work-related experience with that particular technology.

Summary

While the list outlined above is not exhaustive, use it as a baseline for determining which areas you need to review prior to interviewing. You will need to add the frameworks and languages specific to your technology stack. You should rank your level of skill in each of the areas outlined above. If you encounter areas of weakness, spend some time researching those areas. This will help you market yourself as a well-rounded and knowledgeable engineer.

10 - ESSENTIAL SOFTWARE ENGINEERING COMPETENCIES

Possessing essential skills is necessary to land your dream job. However, skills alone are not going to be enough. There are also a number of competencies that you will be rated on during the course of an interview. In the following sections of this chapter, we will discuss a variety of competencies that are important for software engineers.

You should use the sections below to help you identify your areas of strength and weakness. Rank yourself on a scale of one to ten in each of the areas outlined. Be prepared to speak about how you have been able to apply your strengths as a software engineer. You should also be prepared to articulate how you are able to overcome your weaknesses by leveraging your areas of strength.

Experience, Education, and Job Specific Skills

How does your experience line up with the requirements outlined in the job description? Are you familiar with the languages, frameworks, and technologies used within this organization? What is your ability to design, code, test, and deploy production-ready applications? What kind of education do you have? This includes college degrees, conferences, self-study, and other venues that help you gain industry-related skills and expertise.

Be prepared to speak about your skills, experience, and education in such a way that will let your interviewer know that you have the technical background required to do the job.

Judgment, Decision Making, and Pragmatism

Exercising sound judgment consists of the ability to incorporate your past experiences, stakeholder needs, priorities, facts, and constraints to arrive at a decision that is going to be the most beneficial for the organization. It requires the ability to deal with situations in a realistic and sensible way, even if theoretical wisdom would advise taking a differing solution. You need to be able to deal with ambiguities to help figure out which items are truly important.

Communication

Your ability to communicate both in verbal and written form is critical as a software engineer. You may be creating design documentation or trying to convince those within your organization why they should adopt some new technology. Either way, the ability to clearly express your thoughts is

necessary. Modern software engineering is extremely collaborative, and the ability to communicate well is required to influence, explain, collaborate, and inform.

Customer Focus

For every task you complete, there is a customer waiting on the result. The customer may be internal or external. Regardless, your work is ultimately consumed by some customer at some point. Do you attempt to think like the customer? Do you try to anticipate the features and functionality that might be requested before the request is actually made? Your ability to anticipate change before it happens will make your customers extremely satisfied, and will help brand you as a visionary.

Teamwork

How well are you able to work within a team? Do you work collaboratively with your team members to help achieve common group goals and objectives? Do you enable your team to work more efficiently by providing value beyond completion of your daily tasks? Engineering positions that do not require teamwork skills are rare. If your teamwork skills are lacking, you will need to work on improving them.

Coaching and Mentoring

Your ability to coach and mentor will allow you to leverage your skills in ways that help others grow and develop. Do not become deceived into thinking that you are more indispensible by keeping your knowledge a secret. The ability to help others grow is immensely more valuable than being the sole owner of a key piece of technology information. Technologies become obsolete, but the ability to mentor is a lasting skill that can be used forever.

Mentoring can be done formally or informally. Maybe you have shown others on your team how to navigate new functionality. Maybe you have given a presentation on design patterns or some relevant technology. Be aware of coaching opportunities, and take them when they become available. Coaching and mentoring is also a great way to reinforce your knowledge on a particular topic, and will also help depict you as the authority in a particular area.

Self Awareness

Self-awareness is critically important for software engineers. Unfortunately, there seems to be a large number of engineers who don't possess this. They are not able to recognize their strengths and weaknesses and subsequently aren't able to realize their full potential. If you want to improve your self-awareness, you

need to be open to constructive criticism.

You should actively seek out feedback from your managers and peers. Use this feedback as a roadmap for your self-improvement. Doing so will cause you to realize immense growth and will provide you with numerous additional opportunities, both personally and professionally.

Organization, Goal Setting, and Stress Management

What is your ability to organize your work appropriately so that you can complete what is expected of you? What is your ability to anticipate and set goals that will benefits both you and your organization? It is important that you have a logical process for managing your work load. You should be able to prioritize tasks and goals based on a number of different factors. If you have a prioritized task list, you will be able to remain productive and calm even when the situation is chaotic.

You should create two task lists. One list is for your short-term tasks, and another list is for your long-term objectives. Every day you should reprioritize your short term task list and work only the items from the top of the list. You should revisit your long-term objectives monthly to reflect on what has been accomplished and to reprioritize as necessary.

Political Savvy and Influence

Political savvy and influence is an area of weakness for many engineers. You may not be interested in navigating the organizational politics to get work done. However, you should realize that the ability to persuade and lead other will make it easier for you to get your work done. These skills may not come naturally to you, but you should still develop them as much as you can. You will find these skills valuable in ways which you had not imagined.

You should take every opportunity you can to present information to differing groups of people. These informational sessions will provide others with the opportunity to learn who you are, will help brand you as an authority, and will give you credibility. You will find that this credibility will make it much easier to get others to support your thoughts and ideas.

Strategic Skills

Do you have the ability to deal with problems in a holistic way that goes beyond tactical measures to achieve a comprehensive solution? Are you able to distinguish between strategic and tactical measures? The ability to think strategically will be required if you plan on pursuing any leadership roles. You will need the ability to set a vision, and then set the course of action required to

fulfill that vision.

Creativity, Resourcefulness, Initiative, Tenacity

Do you have the ability to keep trying, even when the odds seem to be against you? When dealing with problems that have numerous constraints, are you able to come up with creative solutions by thinking unconventionally? Do you seek out solutions to problems that you encounter without being asked?

These are valuable traits when trying to orchestrate change and get things accomplished within environments that are tightly constrained. You will find that regardless of your company size or culture, there will be times when you need to do some outside-of-the-box thinking to get something done. Your ability to accomplish objectives in spite of difficult constraints will brand you as somebody who "*gets the job done*!"

Adaptability

How well do you deal with an ever-changing environment? Technology changes incredibly fast. Competitors are continually bringing new products to market. This requires that you are adaptable and willing to embrace change. Companies need to stay flexible to survive. You will be required to continually morph to meet your company's ever-changing needs.

Intelligence, Problem Solving, and Analytical Skills

What is your ability to learn new things? Do you have the capacity for logic and abstract thought? Do you possess the ability to think through a problem and come to a logical solution? Are you able to draw meaningful conclusions by analyzing data?

These are all critical skills for your engineering career. If your skills in this area are superior to your peers, talk about them during your interview. Make sure that you provide specific examples. This is one the most important competencies for software engineers, and is one of the best indicators of superior performance. Are your problem-solving skills a bit weak? You will need to provide some pretty good examples of how you have been able to overcome your deficiencies in this area by leveraging other competencies of strength.

Motivation, Ambition, Passion and Energy

Do you love your job? Are you excited by the prospect of a new challenge? Are you energized by problems that you encounter in your daily work? If you aren't, this may be a signal that you are in the wrong profession, the wrong position, or have the wrong boss. Think about what you would rather do, and then seek out a

career that aligns more closely with your interests.

You will find that the best engineers have all been extremely passionate and energized by their work. If you want to do your best work, you need to be sure that your career is aligned with your interests and passion.

Summary

I would recommend scoring yourself in each of the competencies outlined above prior to your interviews. Use **Appendix B: Software Engineer Competency Rating Checklist** to help you discover your strengths and weaknesses. You can then devise talking points to emphasis competencies that are your strengths, as well as talking about the ways in which you compensate for weaknesses.

Preparing before the interview will make it much easier to talk through these areas when asked during the interview and make it much less likely that you will be surprised when asked about them.

Pay attention to the competencies that the interviewer is focusing on during the course of the interview. If you don't possess certain competencies of importance, it may indicate that this position is not going to be a good fit for you. You don't want to be put in a situation where you are not going to be motivated, or are unable to overcome the challenges that you will be facing in your new position.

11 - INTERVIEW FOR SUCCESS

It is important to practice prior to your interview. However, there are also a number of things you should do during the interview to create an impression of competence, likeability, and trust. Having an awareness of these principles can significantly improve your chances of leaving a good impression.

In its simplest form, an interview consists of the following three components: questions, responses, and response ratings. Both you and the interviewer are looking for the proper fit. Both of you will have the opportunity to drive discussion and ask questions. As the candidate, you need to make sure you ask any questions required to help you understand how the position will fit you. In the following sections, we will discuss a few ideas that will help maximize your interview experience.

Questions

It will be primarily the interviewer asking you questions, but remember that you are trying to determine whether the position is a match for you also. Prepare a list of questions prior to the interview that will help you determine whether this particular role is going to be the perfect fit for you. When asking questions, keep the following things in mind: • **Focus on Opportunity** - by focusing on what the opportunities of the position will provide you, you will appear engaged and interested in the position for the right reasons. You will come off as motivated and driven.

- **Don't Ask Questions That You Could Have Answered Yourself** - do not ask any questions that you could have easily answered by browsing the company website, the job description, or a quick Google search. This will give the impression that you are not organized, not prepared, lazy, and not that interested in the position.
- **Ask the Interviewer to Describe Real Job Needs** - this will help you determine whether the ideal candidate characteristics line up with your own. Remember that you are searching for a role that fits you as much as the company is searching for a candidate that fits the role. It will also help you understand which of your skills and competencies you should focus on during the interview.
- **Ask about the Next Steps** - this shows that you are engaged in the interview process, and interested in the possibility of accepting this position if offered. If you do not show interest in the position, the

company isn't going to be interested in making you an offer.

- **Watch for Repeat Questions** - if the interviewer asks you the same questions, rephrased in a slightly different manner, think clearly about what your response should be. This typically means that the initial answer you gave them was not satisfying in some way. Try to figure out why the interviewer was not fully satisfied with your previous response. Tailor your subsequent answers in a way that will appease their concern.

- **Do Not Focus on Material Factors** - do not ask any questions that focus on compensation, commute, flexibility, etc. These things can often be negotiated if you receive an offer. It sends the message that you are interested in the job for the wrong reasons if brought up during the interview.

- **Don't Ask Irrelevant Questions** - don't ask the interviewer any personal questions, or any other questions that are not germane to helping you determine whether the role is a good fit for you.

Responses

While it's important to have good questions prepared, it's equally important that you have a structured format for responding to questions from your interviewer. First of all, you need to know what is on your resume. You may be questioned about anything that is on it.

Make sure you've completed all the steps in the pre-interview checklist so that you aren't fumbling for your answers. Interview questions are designed to identify whether you are a good fit for the role from a skills and competencies perspective. There are a few things that you should keep in mind when answering questions.

- **Don't Feel Compelled to Answer the Second You Receive the Question** - an interview is meant to be more of a dialog than a quiz. While you will be expected to provide answers, it's a good idea to ask follow-up questions if you are not clear on what the interviewer is looking for.

- **Think Through Your Answer for a Few Seconds Before You Start Talking** - this will help you focus on what the question is and what your answer should be. This will also create the impression that you are thoughtful and in-control.

- **Align Your Experience With the Position** - make sure to

accentuate your experiences that most closely align with the position. While your experiences may not be an exact match, make sure you mention any experiences that you feel would correlate to a similar set of skills.

- **Be Prepared to Display Your Problem Solving Process** - interviewers are often as interested in the process of arriving at a solution as they are in the solution itself. You should be prepared to articulate your thought process as you work through various problems during the interview.

- **Be Confident** - exuding confidence will give the interviewer the impression that you know what you are talking about. It will also give the impression that you are comfortable and can handle working with a wide variety of individuals.

- **Be Prepared to Back Up Your Words With Examples and Proof** - if you tell the interviewer that you learn quickly, you will be asked to provide specific examples of how you learn more quickly than your peers. If you tell the interviewer that you are a great problem solver, you will be asked to provide details and proof.

- **Never Bad-Mouth a Previous Boss** - there is no situation in which bad-mouthing a previous boss or employer will make you a more desirable candidate. Don't do it, ever.

Response Ratings

The interviewer will either consciously or subconsciously judge the quality of your responses. There are a number of ways in which you can influence how your responses are scored that have nothing to do with the quality of your answer.

- **Be Engaged** - make sure you give the appearance of being interested in the problem you are being asked to solve. Providing solutions to problems that you clearly aren't interested in will significantly lower your score, even if the solution you provide is technically accurate.

- **Be Concise** - speak long enough to provide the complete solution, but don't ramble on. Being unable to concisely articulate the solution gives the appearance of not knowing your stuff.

A Few Things to Avoid

The following list contains a number of common pitfalls that trap many quality candidates. Usually it's just a lack of awareness on the part of the candidate. Remember the following during your next interview.

- **Don't Be Arrogant** - there is sometimes a fine line between arrogance and confidence. You don't want to give the impression that you are too smart to learn from others. A proper amount of humbleness to go along with your confidence will leave a lasting good impression.
- **Do Not Look At Your Resume During The Interview** - this can create the impression that you may have added things that are not totally truthful.
- **Do Not Beg For The Job** - I repeat, do not beg for the job! This makes you sound desperate and insecure.

Summary

Remember that there are a number of factors that go into making a great impression that aren't specifically are related to your knowledge and expertise. Your ability to interact with your interviewer in a socially appropriate and engaging way will leave a significant impression. You should do your best to make sure you are likeable, friendly, and are truly interested in the position that you are interviewing for.

12 - GOOD QUESTIONS TO ASK AN INTERVIEWER

While you have probably spent significant time preparing for questions that will be asked of you, do not forget to prepare questions for your interviewers. You need to determine if the job is going to be a good fit for you. You will be trying to impress the interviewer with your skills, experience, and long list of redeeming characteristics. However, understanding how this job is going to benefit your career long-term is of utmost importance. Keeping that in mind, you are going to need to ask a few questions.

While the questions you ask will be tailored to the things you deem important, I will provide some examples of questions that I like to ask potential employers, and the reasons that I like to ask them. Look at the questions in the following sections of this chapter. Then use these questions to help generate your own list of questions for your interviewers.

TELL ME ABOUT THE CURRENT TECHNOLOGY STACK. WHAT POTENTIAL CHANGES DO YOU ANTICIPATE IN THE NEXT YEAR?

Purpose

You will find that this is a good opening question. The answer to this question gives you an early indication of where the company is on the technology adoption curve. Geoffrey Moore's Technology Adoption Lifecycle curve contains innovators, early adopters, early majority, late majority, and laggards. You should try finding an organization that is similar to you on the technology adoption curve.

Good Response

By knowing the current and near future technology stack, you get a decent understanding of where the company is on the technology adoption curve. As a rule of thumb, you typically would want to work at a company that is either at the same adoption level that you prefer, or one level to the left or right.

Bad Response

Any time a company is at least two levels from where you personally like to be on the adoption curve, they are probably not a good fit for you. If a company adopts new technology at a significantly faster rate than you prefer, you will likely end up feeling lost and uncomfortable. If they adopt technology at a much slower rate than you prefer, you will likely feel bored and unchallenged. In either case, you will not be in a good position to succeed.

WHAT ARE MY GROWTH OPPORTUNITIES IN THIS POSITION?

Purpose

What you are trying to find out is "*what's in it for me*?" What is the value proposition? Every position that you take should help you grow your skills and knowledge. You should be seeking positions that provide you with the type of opportunities that you are looking for.

Make sure that you have created a list of things you are looking for in your next opportunity. This should be done before you start the job search. This will help you narrow your search to specific organizations. Once you have this list, you can use it as a guideline to rate interviewer's responses to the "*growth opportunities*" question.

Good Response

A good response is any response that contains opportunities that you find appealing. You are not going to make a decision based strictly on the answer to this one question. However, it does serve as a good starting point for a conversation focused on the opportunities that would be provided to you.

Bad Response

If the interviewer is not able to articulate the opportunities, or the opportunities mentioned don't line up with your long-term career goals, it might be a sign that this isn't going to be the best position for you.

WHY IS THIS POSITION OPEN?

Purpose

There are a couple of reasons for asking this question. You are trying to determine if the company is growing. You are also trying to figure out if there is some underlying cultural issue that is causing people to leave the company.

Good Response

If the position is a newly created position, that's a positive sign. It indicates growth within the company. A growing company will provide you with more opportunities for personal and professional growth than a company that is not growing. If the position is open because the previous person left for another opportunity, make sure you find out what those reasons were.

Bad Response

If your interviewer doesn't provide a straight answer, proceed with caution. It may indicate that there are some underlying issues. Maybe the previous employee didn't like their boss, didn't fit the culture, or were not provided enough opportunities in the position. Maybe they just wanted a change. In any case, the interviewer will put a positive spin on the reason. Pay attention to the signs and rely on your instincts when determining how to interpret this response.

WHAT ARE THE BEST AND WORST THINGS ABOUT WORKING AT THIS COMPANY?

Purpose

You are trying to find out both the good and the bad of this company. You will find the answers to this question are filled with little nuggets of honesty. Interviewers realize that the position needs to fit both from the candidate and company perspective. While interviewers will temper down the negatives, you can usually get a feel for what kind of issues employees commonly deal with.

Good Response

If the interviewer is willing to talk about all sorts of positives, that is a good sign. If they feel comfortable giving some honest feedback on the negative side, that is also a good sign. You want to be well-informed so that you can make a sound decision if you are offered a position.

Bad Response

You should pay close attention to what is said for both the good and the bad. If the interviewer isn't able to articulate many positives, that's a bad sign.

If it seems like they are avoiding the question about the negatives, it could also be an indication that there are issues lurking that they do not want to talk about. If the negatives are issues that you would rather not deal with, you will be better off seeking other opportunities.

WHAT IS THE APPETITE FOR SENDING ENGINEERS TO CONFERENCES AND TRAINING?

Purpose

You are trying to figure out two things with this question. What is the financial health of the organization? While this isn't exactly telling the full state of a company's financial health, if money is short, the budget for conferences is probably pretty light.

Is the company willing to invest in employee development? If they are not committed to employee growth, you might not want to work there.

Good Response

You want to hear that conferences and additional training course will be made available to engineers. You may ask if there is a specific amount budgeted annually for training purposes, and how much that amount is.

Bad Response

If a company doesn't invest in its employees, they probably have a high turnover rate. Exercise caution when taking a position at one of these companies. This is often a cultural issue and quite likely an indicator of other issues that you will find unappealing.

It is possible that the company would like to send employees for additional training but they don't have the funds. They can't afford to invest for the future because they have cash flow issues in the present. Think about what that means before accepting an offer.

WHAT IS THE TECHNOLOGICAL APTITUDE OF A TYPICAL ENGINEERING MANAGER?

Purpose

There has been something of a mini-movement among technology companies to hire managers that have strong technical backgrounds. While a technical manager does not need to have a software engineering background, they should be able to speak intelligently on broad category of technical topics.

The need to bring products to market incredibly fast has made is necessary to remove extra layers of management abstraction. Additionally, if your manager doesn't have strong technical skills, your performance review is going to depend more on your marketing skills than your engineering ability.

Good or Bad Response

You will have to make up your own mind on what you think a good or bad response consists of. Managing and leading a team of engineers does require a different skill set than developing software.

Regardless of your opinion on this issue, you should make sure that your engineering manager is at least knowledgeable enough to understand the scope of your accomplishments. Remember that you will also rely on your manager for constructive feedback that will help you grow and mature as part of your career development. Make sure that you feel comfortable with your new manager playing this role.

WHAT IS YOUR TESTING STRATEGY?

Purpose

A company without a solid testing strategy is a company with an unreliable product. A company with an unreliable product is a company with unhappy customers. A company with unhappy customers is a company in trouble.

You are trying to understand how this company tests their software. This includes manual and automated tests, integration and unit tests, functional and non-functional (security, performance, reliability, scalability, etc).

Good Response

The interviewer is able to articulate a well-planned testing strategy that involves significant automated unit and integration tests, as well as manual functional testing efforts where required. They should also have a comprehensive plan for testing non-functional requirements such as security, performance, scalability, and reliability.

Bad Response

You will recognize a poor answer when you hear one, so I'm not going to enumerate all the possible permutations. Just remember, if the testing is sub-standard, you will find that there are residual side effects that will cause numerous other operational issues.

DO YOU HAVE ANY ENGINEERS IN YOUR ORGANIZATION THAT ARE WIDELY KNOWN IN THE INDUSTRY?

Purpose

Does the company employ any industry thought leaders? Industry experts can pick where they want to work. A company that attracts these people is probably a pretty solid company.

Good Response

If a company employs people who are widely regarded as experts in an industry, they probably have a number of good things going on. Industry experts can work just about any place they would like. If they already work at the company you are trying to get hired at, you are provided with some level of validation by virtue of social proof.

Bad Response

Since the percentage of engineers that are widely recognized within the industry is small, most organization won't employ an industry thought leader. So it's not necessary a bad sign if a company doesn't have any of these people.

However, it is a bad sign if they are continually losing good engineers. Make sure you check your network to see if you have connections already working within the company. Make sure to reach out to these people and learn everything you can about the company. Even if they aren't engineers, they will provide great insight into the company culture.

WHAT TOOLS DOES THE TYPICAL ENGINEER ENVIRONMENT CONSIST OF?

Purpose

How productive can you be in a particular environment? It depends on how many tools you have at your disposal. Are you going to spend a significant portion of your day undertaking manual tasks that could have been handled automatically? I have worked on a few projects in which I wasn't able to use tools that I typically would have. I experienced significant frustration because it felt like my efforts were being wasted. I was wasting time doing things that could have been handled by the appropriate tool. You should always seek to maximize your productivity. If tools are what it takes to be productive, tools are what you should have.

Good Response

What you really want to hear is that the company will purchase any tool you need. If this is not realistic, you will want to know that they can get you most of the tools you need. You don't want to spend your days in complete frustration working around issues that could be easily fixed with the appropriate tool.

Bad Response

If your company is not willing to invest in the proper engineering tools, it is likely that they tolerate operational inefficiencies in many other areas of the business. You will be better off finding a company that will allow you to maximize your productivity. You will find productive days much more rewarding and satisfying.

HOW OFTEN DO YOU DEPLOY TO PRODUCTION?

Purpose

This question will help determine how nimble an engineering organization is. Can they get products to market quickly? The more often they deploy to production, the faster they are able to respond to changing market conditions.

Good Response

It is good if the time between production deploys is measured in hours, days, or possibly weeks. A company that deploys frequently can react very quickly to market opportunities. It also indicates that they probably have a solid release and deployment strategy.

Bad Response

If the time between production deploys is measured in months or years, it is a bad sign. In the software engineering industry, the world can change in a matter of months. Organizations that can't adapt quickly soon become obsolete. The feedback cycle needs to be extremely short to consistently react to the changing market.

13 - QUESTIONS THAT YOU MAY BE ASKED

You should now have a prepared list of questions that you will be asking your interviewers. In this section, we will prepare you for questions that you may be asked. We will not cover any technology specific questions, as those will depend on your technology stack. You can refer to ***Chapter 9: Essential Software Engineering Skills*** for specific technical areas to focus for interview preparation.

In the sections below, we will go over a number of behavioral style questions that will help the interviewer learn some of your key characteristics. Understanding the logic behind these questions will help you be more prepared when you receive similar questions during your next interview.

Review the questions in the sections below. Spend some time thinking about what you would deem critical characteristics of a software engineer. Then create a list of questions that you would ask this person to determine if they possess those characteristics. Use your list of questions to help prepare you for your own interviews.

TELL ME THE KIND OF THINGS YOU DO TO STAY CURRENT WITHIN TECHNOLOGY?

Purpose

The pace of change in technology is ever accelerating. If you are not actively investing time in learning new technologies, you will soon become obsolete. To keep up with an ever changing context, you need to be constantly engaged in your education. This may be through formal or informal means. You might attend conferences, conduct self-study on new technologies, enroll in formal education, or participate in a variety of other training courses. Whatever your preference, you need to make sure that you are doing things that are continually growing your skills and knowledge.

Good Response

- "*I try to attend at least one or two major developer conferences a year. Last year I attended **AWS re:Invent**, and this year I plan on attending **SpringOne**.*"
- "*I'm continually researching and learning new technologies or skills. Currently I've been studying the Gang of Four design patterns and their application. Prior to that I was investigating Docker as a dev ops solution.*"

Bad Response

- "*I read and follow a number of blogs.*" (This is the answer I hear most frequently if a candidate doesn't actively do much to keep up-to-date).
- "*I read lots of books on technology.*" (Usually a follow-up question about which specific books doesn't yield much of an answer).

Positioning Possible Negatives

It's usually a bad sign if you are not actively investing in your technology education. However, you may be able to temper the negative reaction a bit if you commonly contribute to open source projects or if you conduct other coding exercises in your spare time. Unfortunately, if you are not actively investing in your education, you probably are not coding in your spare time either.

DO YOU CONTRIBUTE TO OPEN SOURCE PROJECTS, OR HAVE ANY HOBBY PROJECT THAT YOU WORK ON?

Variations

Do you contribute to developer forums?

Purpose

There are a couple of reasons to ask about open source and developer forum contributions.

If you contribute to open source or have hobby development projects, you most likely exhibit the following traits. You love coding (you will code for free), you are motivated by the challenge of solving a problem, and you are ambitious (what's not ambitious about providing solutions for people you've never met, when you could be watching TV instead). Engineers who love to code are almost always better engineers than those that don't.

If you are a significant contributor to open source projects, you likely know other engineers who contribute to open source. That is important for managers looking to add engineers to their team. They would rather ask their current team members for referrals than posting another ad to any number of job boards.

Good Response

- "*I'm a frequent contributor to open-source Project X. You can feel free to browse my code contributions on Github. I've also developed a fairly decent reputation on StackOverflow through my forum contributions.*"
- "*I've been building an online dating site with a buddy of mine. We recently released a beta version to production. If you're interested in seeing what we have, you can check it out at http://<www.someurl.com>.*"

Bad Response

- "*I've never had the opportunity to contribute to open source or forums. I don't really have time to code outside of work hours as I've got other commitments.*"

Positioning Possible Negatives

If you don't have any hobby projects and you don't contribute to open source, be prepared to present alternative information. You are trying to convince the interviewer that you are motivated and interested by your career choice. You could talk about conferences that you've attended, and what you've learned. Make sure you mention any self-study exercises or proof-of-concept demos that you've completed to learn about new technologies.

TALK ABOUT THE MOST REWARDING PROJECT YOU HAVE WORKED ON? WHY WAS IT SO REWARDING?

Purpose

This question is usually asked to help determine whether you have traits that are desirable for somebody in this role. It can also provide insight into your personality and help determine whether you are a cultural fit for both the company and the team.

Good Response

There really isn't a right or wrong answer with this one. Just be aware that you should try to position your answer so that it aligns with what you think might be desirable traits for this particular position. For example, you could describe a situation where you were motivated by a challenge, and then created a solution.

- "*I once worked on a web application that was leaking database resources. Typically we'd discover the problem when the application performance began to degrade. Eventually the application server would become unresponsive. Then after 30 minutes or so, when the resources had timed out and been returned back the resource pool, the application would start functioning normally again. Initially we employed a brute force approach. I and another developer would comb through our application code and make sure that every database resource that was opened was then subsequently closed appropriately. This was a very labor-intensive process. Alternatively, I developed a solution using the Decorator pattern in which we were able to close database resources automatically, and log error stack traces indicating the location of the offending code. This saved us hours of time, and fixed this nagging problem forever.*"

Bad Response

- Any response that indicates that you are motivated by material factors will be viewed negatively. You want to demonstrate that you are motivated intrinsically by the challenges that you'd typically face in this particular position.

TELL ME WHAT YOU KNOW ABOUT <CUTTING-EDGE TECHNOLOGY X>.

Purpose

This question is usually asked to help determine what kind of technology awareness you have. Do you know what is trending in the industry? The interviewer won't necessary expect that you will have experience with this technology. However, they would like you to have a general awareness of what the technology is, what the appropriate use-cases and context are, and a basic understanding of how it works. You may get a number of these types of questions on differing technologies, even if those technologies aren't listed on your resume.

Good Response

- "*While I don't have specific experience with this technology, I've done a little research on it. I know that it's typically used for these types of scenarios and use cases... [Enumeration of appropriate use cases]. I also created a proof-of-concept just to get a basic understanding of how this technology could be leveraged at my current company.*"

Bad Response

- "*I haven't gotten the opportunity to work with this technology, and so I really don't know anything about it.*" [While it's ok to admit that you don't have knowledge on a certain technology, it's not ok to say you haven't gotten the opportunity to work with it. Motivated individuals will make their own opportunities to learn about various technologies.]

WHAT KIND OF THINGS HAVE YOU DONE TO PROMOTE TECHNOLOGICAL CHANGE WITHIN YOUR ORGANIZATION?

Purpose

There are a couple of things that an interviewer is looking for with this question. What is your ability to influence? What is your propensity to look for better solutions than the status quo? Being able to recognize when a change will improve operational efficiencies and then being able to influence others to change will make you incredibly valuable to an organization. If you possess this talent, be sure to mention it, even if you are not asked about it.

Good Response

- "*When we started down the path of micro-service architecture, I realized that our dev ops deployment model was not going to scale well. I researched how we could leverage Docker to improve the speed and efficiency of our deployments. I created a proof-of-concept, and demonstrated its functionality to the dev ops team. Once we adopted Docker, we were able to go from weekly deployments in development to multiple daily deployments.*"

Bad Response

- "*Every team that I've ever worked on has always run pretty smoothly. I can't think of any situations that required any changes.*"

Positioning Possible Negatives

It may not be a deal-breaker if you aren't able to provide specific examples of promoting change. However, you should be able to demonstrate that you have done some related things on a smaller scale. Maybe you refactored a particularly troublesome piece of system functionality to make it more maintainable and easier to understand. Maybe you promoted some change within your team to improve how it functioned. You should mention any examples that demonstrate initiative or influence.

IF YOU COULD CHANGE ONE THING IN THE ARCHITECTURE OF YOUR CURRENT SYSTEM, WHAT WOULD IT BE?

Purpose

There are a couple of reasons an interviewer may ask this question. One is to determine your level of understanding of the current architectural landscape. The other is to determine your level of context awareness. Are you able to realize when there might be a better way to do something?

Good Response

- A good response will indicate that you have a solid understanding of the current architecture. You should also have an idea of something significant that could be changed to make the architecture better. If you have specific examples of how you acted on an idea to implement an architectural improvement, be sure to mention it.

Bad Response

- A poor response is one that is so trivial that it really doesn't provide much added value. This is usually an indication that you haven't reached an engineering maturity to think at an architectural level. You likely would struggle with tasks that involve some level of design and architecture.

TELL ME ABOUT THE ACCOMPLISHMENT IN YOUR CURRENT POSITION THAT AFFECTED THE COMPANY MOST SIGNIFICANTLY?

Purpose

The interviewer is trying to get a feel for what the scope of work is that you typically take on. The more significant the accomplishment you describe, the larger the scope of the work you will be able to take on. You may also be asked to provide examples from a few different positions that you've been in. In these cases, the interviewer is looking for a pattern of increased responsibilities, and increased level of accomplishment. Your greatest accomplishments should be from your most recent positions.

Good Response

- A good response will consist of an example accomplishment that is representative of someone who has your level of experience or better. Additionally, if you are asked about accomplishments from previous positions, your accomplishments should be increasingly more significant over time.

Bad Response

- If you are unable to provide any meaningful examples that would be representative of someone with your level of experience, the interview is not going to go well for you.

WHAT ARE YOUR GREATEST STRENGTHS AS A SOFTWARE ENGINEER?

Purpose

An interviewer will ask this question as a means of determining your level of self-awareness. You will be expected to provide a number of specific examples where you exhibited this strength. Make sure you talk about why this particular strength is so valuable to a software engineer.

Good Response

- A good response is one in which you are able to provide numerous specific examples of how you possess the strength that you've chosen to talk about. Make sure the strength you have chosen is relevant to the position you are applying for.

Bad Response

- If you are not able to provide specific examples demonstrating how you possess the strength you mentioned, the interviewer will assume you don't actually possess it. Interview talk is cheap, and you will be expected to back up all assertions with supporting facts.

WHAT HAVE YOU DONE TO IMPROVE YOUR DEPARTMENT'S OPERATIONAL COSTS?

Purpose

Some software engineers do exactly as they are instructed. Some of them are incredibly gifted. If they were told to make the build process faster, they would do it. If they were told that the web application response times needed to be cut in half, they would probably figure out a way to do that also. However, they wouldn't have the foresight to think of doing these things on their own. There is certainly a place for gifted engineers who aren't always dreaming up new ways to make things more efficient. However, if you are an engineer who has the insight to see inefficiencies, and then figure out ways to fix them without being prompted, you are in a league of your own! Employers are clamoring for engineers like you. You will often justify your wage several times over in operation efficiency improvements alone.

Good Response

- Make sure you mention anything you've done that has made your department more efficient. Maybe you've made the build and deploy process more automated. Perhaps you've built a clever tool that made some repetitive chore easier. Maybe you've introduced some software that made project or task management more efficient. Whatever the case, think of things that have improved the operational efficiencies from a macro perspective.

Bad Response

- While you may not have come up with any great ideas on improving operational efficiencies, the chances are that you've worked on projects that have resulted in them. Even if the idea wasn't your own, let your interviewers know that you have worked on projects that improved operational costs. You should at least demonstrate that you understand the concept. Being unable to mention anything related to operation cost improvements will create the impression that you are not able to think strategically.

WHAT HAS BEEN YOUR MOST CREATIVE ACHIEVEMENT AT WORK?

Purpose

There are often problems encountered by software engineers that aren't going to have an obvious solution. They may require some unconventional thinking. The solution to the problem may not even be a technology solution. It might require a change to a business process or possibly thinking about the business context in an entirely different way. The interviewer is attempting to gauge how you will be able to handle these types of situations when you encounter them. How well are you able to step back and evaluate the big picture when you encounter these types of challenges?

Good Response

- *"We once encountered a situation where we were asked to build a very complex workflow into our system that our current architecture really didn't support without an extensive refactoring effort. I went back to the business to better understand their needs and look for alternative solutions. After further discussions about their business goal, I was able to propose a change to their business process that supported their business objectives without any changes to our software."*

Bad Response

- If you aren't able to think of any situations where you came up with a creative solution to a problem; your interviewer is going to think that you are going to need hand-holding every time you encounter a difficult problem. You can counteract this impression by giving a specific example of how you solved a troublesome issue using some other competency such as resourcefulness or tenacity.

WHAT IS THE BIGGEST MISCONCEPTION PEOPLE HAVE ABOUT YOU?

Purpose

This question is more behavioral than technical. However, it is important to mention. If you get this question, beware! You are entering the danger zone. If there are things that you feel are misconceptions about you, it is quite likely that they aren't misconceptions. You just haven't been able to recognize this characteristic in yourself. Even if it is truly a misconception, your interviewer is going to assume the misconception to be true rather than otherwise. This is often just a subtle way of asking '*Could you please articulate your faults and other reasons why I may not want to hire you?'*

Good Response

- You may want to just avoid this question by saying that you really can't think of anything or that you generally feel like people have an accurate representation of who you are. If you feel compelled to come up with something, make sure it's either so benign that it won't leave much of an impression on your interviewer or it is something that could be viewed as a positive characteristic. Don't go over-the-top in trying to spin it into a positive. You will sound disingenuous.

Bad Response

- Any response in which you mention anything that could be thought of as a negative trait, regardless of whether it is true or not, will reduce your desirability as a candidate. Remember that your goal on this question is just to break even. Don't spend a lot of time talking about how you see yourself in a different way than everyone else does. You will have lots of other opportunities to impress them, so work around this question gracefully if you receive it.

14 - ROLE-BASED KNOWLEDGE EXPECTATIONS

So now that you are prepared for a number of interview questions, we will outline some basic requirements for engineers at each level of experience. While every position is unique, and the organizational needs may vary widely, there are generic levels of competence that should be attained by each engineer depending on their experience.

Entry Level Software Engineer

An entry level engineer is not going to have much experience developing software. If you are just starting your career, you will be measured differently than an engineer with significant experience. Your interviewer will measure your problem-solving skills, motivation, and passion for coding. You may see a number of problem-solving questions that you will be required to solve in the programming language of your choosing.

You will likely face a number of questions about things that you were expected to learn in your Computer Science curriculum. Data structures, search algorithms, runtime complexity, and other theoretical concepts. While an interviewer may not ask you about your GPA, they will use it as an indicator of how you performed relative to your peers.

You will likely be asked to explain any development projects that you completed as part of your degree curriculum. You will also have to walk the interviewer through your thought process. While you may not be grilled about all the latest technologies, you will need to demonstrate that you have some idea of what might be trending within technology.

If you have any projects that you have worked on outside of your school curriculum, be sure to mention those. Whether it's an open-source project or just something that you are working on for your own purposes, it is definitely worth sharing. This will help demonstrate your passion for engineering, as well as your level of motivation and ambition.

Senior Software Engineer

A senior software engineer can encompass a wide range of experience, but there are still a number of concepts that you should understand.

You should have a fairly solid grasp of design patterns and object oriented principles. Design patterns are a fundamental element of solid software design, and also indicate a level of engineering maturity. You should also have a solid understanding of regular expressions. Engineers who have a solid knowledge of regular expressions are typically better able to comprehend abstract thought.

With the ubiquitous nature of cloud-based applications, application security is becoming an increasingly important area of expertise. While you don't need to be a security expert, you should have a solid understanding of basic security concepts. This includes both application and system security.

Performance and scalability is another important area of knowledge for senior engineers. Even the most basic applications encounter performance issues. You should be familiar with the tools and techniques that are commonly employed to debug performance issues within your specific technology stack. You should also be familiar with multi-threading architectures and thread concepts.

You should be fairly comfortable with SQL. You should also understand the basics of indexing and query optimizations. Additionally, you should also understand the different types of NoSQL databases, and when their use would be appropriate.

You should also be able to write and navigate recursive functions, and understand which use cases are appropriate. You should understand static and dynamic-typed languages and the pros and cons of each.

You should also understand the basics of both SOAP and REST web services, and know the benefits and drawbacks of each. You should be familiar with various client-side frameworks, and some of the key points of each, even if you don't have experience with front-end development.

Team Lead

In addition to possessing the knowledge of a senior software engineer, a team lead should also have great software architecture skills. You should understand enough architecture to make solid design decisions as you lead and guide your team. You should also have a solid understanding of system architecture and scalability, and be able to make recommendations on integration strategies. You should know how cloud-based architectures can be used in both production and pre-production environments. You should also know how big data technologies can be leveraged by your applications to provide business value.

In addition to your technology skills, you should be adept at navigating organizational politics. You will leverage these skills to help keep your team productive. You should also have a thorough understanding of the business domain as you will be relied on to translate business needs into working application functionality.

15 – WRAP UP

Thank you once again for taking this journey with me. I hope you have found value in the material as I have presented it. Your career as an elite software engineer is well started. I hope that you find your profession as rewarding and fulfilling as I have.

In the quest to grow your knowledge and skills, continue to have the intellectual curiosity that got you started in this profession. Continually ask yourself and others how you can improve your knowledge and skills.

Remember that preparation is the key to interview success! Use the information in this book as a framework for your interview preparation. Create your own preparation checklist. Remember to focus on your strengths, but be prepared to explain how you mitigate your weaknesses.

The purpose of this book has been to provide you with a framework to begin your interview preparation. This book was not meant to provide you with all the answers. I'm sure you have already thought of many things that I've missed that would have made this book more valuable. Use the principles in this book to craft a plan for landing your dream job!

Good luck on your job hunting journey!

APPENDIX A: GREAT SOFTWARE ENGINEER CHARACTERISTIC CHECKLIST

	Great Software Engineer Characteristics
Self-Awareness	Seeks out constructive criticism and ways to improve. Has solid understanding of strengths and weaknesses. Continually strives for improvement.
Tech Awareness	Constantly investigates the latest technologies and tools. Always knows what's trending within the industry, and has a basic understanding of how these technologies may be leveraged.
Design Considerations	Iteratively works through solutions to arrive at the best designs. Understands what constitutes solid design at a class, framework, application, and system level.
Problem Solving	Incredible knack for determining root cause on any issue. No problem is too big or too small, and each will be solved permanently. Has the ability to think through problems from both a strategic and tactical perspective.
Motivation	These engineers are intrinsically motivated. The sense of accomplishment is enough reward. They are less likely to be affected by office politics and often don't need lots of positive reinforcement.
Passion for Technology	New tools and technologies are toys for these types. They are continually checking out the latest and greatest technology. They like to live on the bleeding edge of the technology adoption curve.
Passion for Coding	Programming is a hobby. Will program things that have no practical value in the name of entertainment. If an API exists, chances are they have tried it out.

APPENDIX B: SOFTWARE ENGINEER COMPETENCY RATING CHECKLIST

	Superb	Adequate	Needs Work
Communication			
Integrity			
Intelligence			
Likeability			
Listening Skills			
Motivation & Ambition			
Problem Solving & Analytical Skills			
Teamwork			
Adaptability			
Assertiveness			
Creativity			
Coaching Skills			
Customer Focus			
Education			
Experience			
Influence			
Job Specific Skills			
Judgment & Decision Making			
Organization Skill			
Pragmatism			
Political Savvy			
Resourcefulness & Initiative			
Strategic Skills			
Stress Management			
Goal Setting			
Self-Awareness			

Tenacity			

APPENDIX C: SOFTWARE ENGINEER MATURITY MATRIX

	Entry Level	Senior Engineer	Team Lead
Data Structures	x	x	x
Runtime Complexity	x	x	x
Search & Sorting Algorithms	x	x	x
OO Design Principles	x	x	x
Build Automation		x	x
Application Security		x	x
Big Data Awareness		x	x
Client Side Frameworks		x	x
Database Technologies		x	x
Design Patterns		x	x
Dynamic Languages		x	x
Performance Diagnostics		x	x
Recursion		x	x
Reflection		x	x
Regular Expression		x	x
Thread Concepts		x	x
Test Automation		x	x
Web Services		x	x
Cloud Based Architectures			x
System Reliability			x
System Scalability			x
Software Architecture			x
System Architecture			x

AFTERWORD

I want to thank you again for reading this book and taking the first step in your goal of landing the job you will love.

I wrote this book because I have witnessed many candidates self-destruct during the interview process. Many of these software engineers would have been worthy candidates for the position. I have self-destructed on more than one occasion. I would have loved to have this book available to me, especially early in my career when my interviewing skills were quite raw.

My hope is that you have found concepts of value in this book that will aid you in your search for the ideal job.

PLEASE leave a review on Amazon.com. I welcome your feedback! Reviews will help other readers understand the value that this book provides. They also help me understand where questions and frustrations exist, so that I can provide additional clarification in future versions of the book.

Best of wishes on a rewarding and fulfilling career!

Ryan Ylitalo

PS: Don't forget to check out www.byteperceptions.com and sign up to receive update notifications on any revisions to this book or other software engineering related training materials that I will be coming out with in the future.