Interview Process

End-to-end Process

1. Apply for a role (see applying section of series)
2. Email with Phone Screen or Online Assessment
   - **Phone Screen**: Candidates who meet the basic requirements may be invited to speak with a phone screener – depending on the company and its size, this may be the recruiter or one of the hiring team. The purpose of the phone screen is to determine the candidate's identity and previous experience, as well as their skill level, using some basic questions to evaluate competency before moving forward.
   - **Online Assessment**: Candidates who meet the basic requirements may be invited to complete an online assessment. The assessment is designed to measure a candidates skill level specific to the role. For engineering roles this is often two to three timed coding questions, which can be done through a platform like Hackerrank. The online assessments will drastically cut down the applicants before an interviewer is required to manually participate in the next stages.
3. Prebrief
   - The candidate receives an email with more information regarding the full interview loop and the role. There will be an online calendar to set times for the following stages (or reply via email), both technical and behavioral.

End-to-end Process

3. Technical Interview
   - There are typically 2 technical interviews, these will be carried out alongside an interviewer one-to-one (not necessarily someone in the direct hiring team). Can be in-person or a video call.
4. Behavioral Interview
   - During this section, the interviewer is looking to evaluate the individual's attitude and character to ensure they are a good fit for the company's culture. The hiring manager will typically be involved in this interview since the candidate will be directly working with this manager (if not they may schedule a separate interview).
5. Debrief
   - During each interview, the interview takes notes on the candidate, using a framework (in later slides). These are the strengths and weaknesses of the candidate, justified with example reasons.
   - For each interview, the interview is assessing the candidate against certain competencies, skills or company culture points. These are available online, for example one of Meta's principles is to Build connection and community.
   - The interviewer decides if they are positive or negative on hiring the candidate, and shares all this information with the rest of the interviewers and the hiring manager. At the end of the meeting the collective decides if the candidate will be given an offer.

Online Assessment & Phone Screen

Phone Screen/Online Assessment

- In the case your resume application is accepted, the first communication you will receive is an email with the next step details.
- For engineering roles the online assessment is more likely than the phone screen interview. However both are also still possible together. The email (or a follow-up email) will typically have a link to complete the online assessment – with a deadline date. Given there is a phone screen, there will be a link or the opportunity to reply, setting a date and time, and to provide your details.

👨‍💻

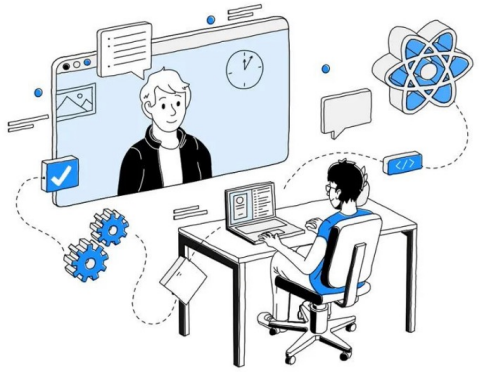
- The online assessment is the typically 1-2 hours- with a timer for the overall assessment or per question (e.g. 45 minutes per question), and generally covers around 3 coding challenges via a platform like HackerRank. The best way to prepare for this is practicing the coding questions portion of the material, revising your notes, trying new questions, and setting a timer on yourself to complete your solution.
- Reach out to friends, family, colleagues, or people who are willing to mock interview you if you can find anyone. There are also online communities, where friendly experts will mock interview (paid or free), or similar individuals practicing for interviews (Discord servers, Reddit, and other online communities are options to reach out).
- Some companies may offer “take-home” projects, where you are given some requirements, constraints and a deadline (days - week). Either online assessment or take-home project allows you to use the internet and online resources since it is not a live one-to-one, so take the opportunity.

☎️

- The phone screen is typically 15-40 minutes. It may consist of a range of basic questions, confirming your identity, and technical questions regarding information that you may have provided on your resume, or information relevant to the role, for example “What are some common data structures and their principles?”, “Can you tell me what unit testing is?”, “What are some steps you take to debug code?”, “I read on your resume about this project XYZ, can you tell me more about it?”. To answer questions in this section – revise your software engineering concepts, and keep your answers concise yet informative – check the behavioral section for a framework on how to answer spoken questions.

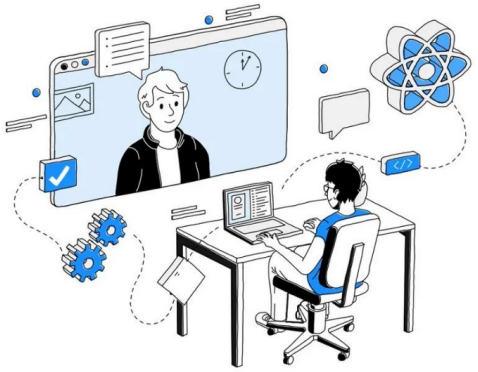
Technical Interview

Technical Interview: Overview



- The technical interviewers are very similar to the coding problems you have seen so far, or during the online assessment.
- The differences are that during these, you are coding live, one-to-one with an interviewer – a senior engineer from the company or the hiring team. The interview has to complete internal interviewing/hiring training and have shadowed others in live interviews.
- The interviewer typically has the freedom to choose the questions they ask – and they are not limited to coding questions. For example, they may select coding questions, or invent their own, which are more tailored to the role you are applying to e.g. web development or machine learning questions. They can also ask knowledge questions, again likely related to the role, for example "Can you tell me the difference between client and server-side programming?" for a web development role, or "Can you tell me how we can use the cloud to implement the solution we have built here?" for a software engineering role.
- Guides for the interviewer suggest good interview questions are typically "real-world" and not puzzle questions, since "real-world" is the environment the candidate will be operating and solving problems in. A good question will also have many layers, dynamic enough to make it easier or more difficult depending on how the candidate is responding, with multiple solutions – so the candidate can showcase their critical thinking skills.
- This interview is an opportunity for the candidate to showcase the depth of their technical knowledge and coding skills. The candidate should speak their thoughts openly, and we have gone through examples of this in the coding problems sections. This will be situational, however, for example the candidate can ask clarifying questions, "Will the input given to me always be a string?", "Should the output be sorted or is any order acceptable?". This is a noteworthy display of critical thinking skills.
- We have also covered this in the coding problems section, however, the candidate should also voice their thoughts, this includes approaches they are considering, trade-offs and the end-to-end solution, for example: "I am considering using a hash-map to store the frequency of each element in the array because it allows for O(1) – constant time complexity for look ups and insertion. However this approach will use additional space – O(N) for all N elements."
- Interviewers will often guide you in this section for multiple reasons, this is another benefit of voicing your thoughts. The help from the interviewer is not to be seen as a negative, they guide you because they may later give you an extension to the problem which a certain coding approach at the start is suited for. They will also guide the candidate since they are a junior engineer, and it is expected that they are not an expert; this provides an opportunity to see how quickly and efficiently the candidate can listen to help, or ask for help – since this is a skill in itself – especially for entry engineers.

Technical Interview: Overview guidelines given to interviewer



- Here are some guidelines for interviewers on what an entry level software engineer should be competent at – also take a look at the hiring guidelines document:

Overall Technical skills:
1. Coding (Logical & Maintainable)
2. Coding (Data Structures & Algorithms)
3. Coding (Problem Solving)
4. Component/System Design - This is used to evaluate SDE II (Senior) and III (Tech Lead) candidates

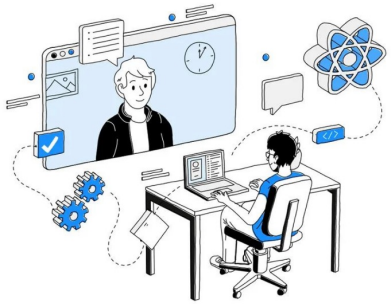
Computer Science Fundamentals SDE I-III (all levels):
• Understand underlying implementation
• Understand different data structures and algorithms (and can they make good trade-offs)
• Runtime and Space complexity

Coding SDE I (entry-level):
• Is able to identify ideas, and convert them into code
• Implements working solution in reasonable amount of time with fluency or mastery
• For Logical & Maintainable competency - write code that is readable and maintainable
• Mostly correct with only trivial syntax correctness; Appropriate error-handling and self-testing of solution

Problem Solving SDE I (entry-level):
• Thinking through the problem
• Identifying alternatives, weighing and choosing
    - The ability to identify and recommend trade-offs in solutions is critical in university candidates. For example, providing a brute force solution is acceptable if the candidate can identify an option to optimize and/or the trade-offs and constraints of their solution
• Seeing potential roadblocks/challenges with chosen solution

SDE I:
• Design is not necessary for SDE I interview (i.e. not a reason to reject in debrief)
• Strong design skills cannot compensate for weakness in coding
• Logical and Maintainable coding has some facets of Basic Design, and university/academic candidates are not expected to be bar-raising in code maintainability, as it is expected it can be taught with industry experience

Technical Interview: Internal coding-specific guidelines given to interviewer



- Here is an example of an interviewer's evaluation guidelines for a candidate's technical interview performance. These are questions are an internal company resource, or in some cases provided directly from the hiring manager, and they will provide the justification for the positive or negative response the interview gives in the debrief:

Evaluating Responses - Coding

Did the Candidate:

Find most bugs (depending on level)? Consider whether they found the bugs on their own or after hinting, verified that the code worked, and corrected any defects.

Deal with edge cases? Assess if the code generally worked or if they missed in multiple areas.

Test their own code? They may or may not do this on their own. What you are looking for is someone who gets to a solution, even if it is not optimal. Then when you get to the solution, can you think about the edge cases with this solution.

Use language idioms correctly? If they say JavaScript is their best language and they use it incorrectly then that is a red flag. Again, it doesn't matter what language they use, as long as it is correct.

Is the Code:

Reasonably simple, given time constraints? We want to write syntactically correct code, but at the same time if they miss a semi-colon, it is okay. We use software now that would flag us if we misspell or leave a semi-colon out; therefore, if the candidate does make this mistake while writing on a whiteboard (or in an editor that does not check) and under pressure, we should be forgiving.

Correct? Does it work? If the candidate gets to the end of writing code and it does not work, then you have a big red flag. This is again assuming that you asked them a good question that could be answered in the time given. Using hints, you can also help push the candidate to get to the right answer.

Efficient? If they get to a quick solution, but it is not an efficient way to solve the problem then help direct them to look for another solution.

Behavioral Interview

Overview of Behavioral Interview

- **Behavioral Interview**: interview to assess a candidate's past experiences and actions to predict future behavior and suitability for the role and company culture.
- Often overlooked by candidates, behavioral interviews are very important. The behavioral interview is the opportunity to separate candidates, especially if the technical interviews are all fairly equal.
- Behavioral interviews are just as important as the technical (if not more) because it’s well known that a candidate with the correct behavior/attitude is more than capable of learning the technical aspects they may lack. Whereas a candidate with very impressive technical skills but poor behavior or weak team-working/social skills is more likely to be rejected.

Behavioral Interview: Format

- The interview is typically 40 mins – 1 hour. A senior member of the hiring team or the hiring manager themselves will likely be present since this is the direct team that the candidate will be working with.
- One behavioral interview may consist of several one-to-one interviews.
- Each company will have their own style but the overall approach is similar. The interviewer is given certain company principles or character traits to evaluate the candidate on. There a question sheets with common questions an interviewer can use, or they can invent their own.
- The questions will be related to the role the candidate is applying for and the candidate's background, looking at their previous experiences. The question will not be problem-based or require a solution like the technical interview.
- The questions asked will typically begin by touching on a past experience, with the opportunity to lead onto hypotheticals. Interviewers will be probing your story, so make sure it is a valid answer and a personal experience. For example if you were part of a group project and you state that you solved some problem - you will be expected to have deep knowledge of this problem e.g. "Why did you go with this choice?", "What were the alternative options?", "What was the biggest obstacle?".

Behavioral Interview: Questions

- Interviewers will be given a table or note taking framework similar to this and ask 2-3 questions regarding per principle:

Company/Behavioral Principle			
Strong Negative	Mild Negative	Mild Positive	Strong Positive

Behavioral Interview: Questions Example

- Interviewer at Google example – Fast is better than slow
- Questions selected:
  - **Tell me about a time you took a calculated risk where speed was important**. What was the situation and how did you handle it? Did you take any steps to reduce the risk? Would you have done anything differently in hindsight?
  - **Tell me about a time you worked against strict, short deadlines and did not have the opportunity to collect all the data before making a decision.** What was your approach? Did you meet the deadline? What did you learn?
  - **Tell me about a time you had an problem impacting your team and took a proactive approach to solve it?** What was the problem? Specifically what did you do and what was the outcome? What did you take away from the situation?

Company Principle: Fast is better than slow			
Strong Negative	Mild Negative	Mild Positive	Strong Positive
• Slow to make decisions that have no significant impact. • Fails to meet deadlines.	• Nervous to tackle tasks that are new to them. • Consults others too often, even on tasks that are insignificant.	• Moves quickly on a project, even when information is missing. • Able to determine and prioritize what is an urgent task and the tasks that can be delayed.	• Understands when to contact others and when it's not necessary. • Handles issue promptly so as to unblock others.

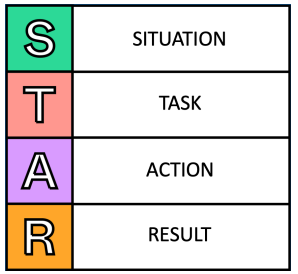
Behavioral Interview: Questions Example

- Interviewer at Meta example – Serve Everyone
- Questions selected:
  - **Describe a difficult experience with a customer (or team member if no customer examples).** How did you handle it? What was the final result? What would you have done differently?
  - **Can you give me an example of a time you went beyond your defined responsibilities for a customer.** What did the customer want? What did you do for the customer? Why did you do it? How did they react? What was the result?
  - **It's possible for customers to make irrational demands, has this happened to you and if so tell me how you pushed back.** What was their request? How did you turn down the request? Why did you push back? What was the final outcome?

Company Principle: Serve Everyone

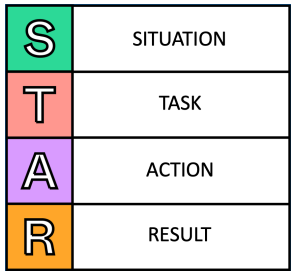
Strong Negative	Mild Negative	Mild Positive	Strong Positive
• Unable to identify the customer. • Fails to fulfil company promises to customer and prioritizes low-impact work over customer deliverables.	• Fails to collect or listen to customer feedback and make adjustments. • Focuses on competition and trends over direct customer requirements.	• Takes suitable actions based on customer requests. • Listens to customer feedback, and takes action to collect feedback.	• Goes beyond required scope to please customers by exceeding promises. • Not afraid to push back when the customer requests goes against company values

Behavioral Interview: Answering questions



- Your objective as the candidate is give the interviewer every opportunity to fill in all the strengths in their notes, and none of the weaknesses. You should give them no reason to reject you and every reason to hire you. This is very possible since you are the one answering the questions and have all the control in this situation.
- Answer questions with the following format: **STAR**
- **S – Situation:** Describe the situation you were faced with. Think of the context, What? Who? Why? Where? When? They do not need to know every detail, do not bore them with the peripheral details, just enough information so they understand the context.
- **T – Task:** This is the opportunity to dive deeper on the specific task at hand. Some extra details on the task will be helpful – Why was this important? What were the constraints? What was the scope of the problem? What were the risks?
- **A – Action:** This is the section of your answer where you highlight your value. What actions did you take? What part of the project did you own vs the team? How did you do it? What was the most significant obstacle and how did you approach the problem? Did you have to work with others or influence decision making- if so how? How did you define the priorities or tasks? What technical actions did you take?
- **R – Result:** Back up your actions with the results. What did you achieve? Use hard data if possible – revenue generation, cost savings, customer acquisitions, time savings, user session times. Remember from the resume section – Accomplished X by doing Y as measured by Z. Why did you use this measurement of result? Showcase critical thinking skills by mentioning any trade-offs e.g. greater immediate cost but more users and potentially greater lifetime value.

Behavioral Interview: Answering questions



- A good rule of thumb is 1 minute 30 seconds to answering behavioral questions.
- To prepare look over your resume – which should be ready by this point. Bullet point the 5 most substantial experiences – some common, solid examples include a major university project or final research project, personal projects, any with real end-users/customers is a major positive, any professional or industry experience. If you lack examples directly related to engineering or with some technical aspect – don't hesitate to include other experiences - retail jobs, voluntary experience etc.
- For each experience go through the STAR format and note down bullet points for each STAR point. Since you will not know exactly what question will be asked in the interview, these notes should serve as talking points that you can adapt to the question.

Example Experience: **Creating a website for the university Chess Society.**

Situation:
- Only person with technical experience
- Never created a website without a website builder

Task:
- 2 weeks to create for a university-wide chess tournament
- Limited resources due to small society funds

Action:
- Created the website with HTML/CSS and JavaScript – the most significant feature being email sign ups.
- I reached out to Computer Science teaching assistant to get idea of the scope and useful libraries before coding.

Result:
- 20 people signed up to the chess tournament using my website with 1000 page visits.
- This improved my full-stack skills which led me on to teach myself React and show others in the society how to build a single page static website.

Interviewer Debrief and Offer

Debrief

- The debrief is a meeting the candidate has no involvement with, however some information about the meeting may help influence your decision making and thought-process during the interview loop.
- Ultimately each interviewer is answering the question: "Based on the information you collected on the candidate – should we hire them?"
- Some rough guidelines in big tech companies:
  - Is this candidate better than 50% of the current employees in the same role and level?
  - Does this candidate have the potential to grow and take on more responsibility in the next 2-3 years?
  - Will this candidate bring unique skills or perspectives that we currently lack in our team?
  - Has this candidate shown a track record of consistent performance and reliability?
  - Is this candidate adaptable and capable of learning new technologies or methodologies quickly? Can we overcome any current weakness areas with mentoring?
  - Will this candidate help us achieve our short-term and long-term business objectives?
  - Does this candidate uphold high ethical standards and integrity in their work?
  - Is this candidate receptive to feedback and willing to make improvements based on it?
- Companies and interviewers often believe it's better to occasionally make the wrong decision by not hiring a good candidate than by hiring the wrong candidate. So they will err on the side of caution – this is why the candidate has to give the interviewers every reason to hire them at this stage.

Extra Resources

Extra Resources

- Companies will offer an abundance of free resources. Contrary to popular beliefs there is a shortage of (good) engineers. So companies will provide material and opportunities to find, train and hire these engineers:
  - Amazon SDE course (can skip system design if going for entry role)
  - AWS Certifications
  - Google Courses
  - Google Certificates
  - Meta front-end and back-end courses
  - Apple apps course

- These are a few of the courses and certifications direct from big tech companies. Depending on the specific role and level you are interested in, you can find different opportunities.