Purpose of this doc

This is a guide for Engineering Management interviews at FAANG like companies. Note that EMs also have various levels and across levels, your previous experience, and the teams managed / impact delivered will be an important factor.

- This is to only be treated as a directional input. This is NOT a cheatsheet to interviews.
- This is only applicable for FAANG like companies (Google, Amazon, Microsoft, Salesforce, Adobe, Walmart, etc.). At other companies, guidelines might vary depending on the scale of company and how mature their processes are.
- This is only applicable to Engineering Management interviews. Not applicable to IC rounds
- In all discussions, you are expected to drive discussions.

Process definition

Phone Interview:

- 2 Phone screening rounds. Things being judged:
    - People management skills + Previous experience
    - System Design

Onsite interviews:

- 1 Problem Solving round:
    - 2-3 DSA questions - easy questions most of the time.
- Hiring Manager round:
    - People Management drill down
    - ***Competencies checked:*** Hiring, growing, motivating, firing, succession, measuring right things, being scientific in approach.
- System design round

    - ***Competencies checked:*** Data modelling, architecture, fault tolerance, back of the envelope calculations, API design, scale up systems.
- TPM core round
    - Technical proficiency
    - Project management skills.
    - ***Competencies checked:*** SDLC, Priorities, Milestones, Stakeholders, timeline estimations, tech skills including DSA.
- PM core round
    - Vision + roadmap planning given a case.
    - Planning of customer features
    - Planning around team size / funding.
    - ***Competencies checked:*** Roadmap, funding, Customer research, features, vision, metrics & instrumentation.
- Bar Raiser
    - ***Competencies checked:*** Troubleshooting, risk mgmt, preparedness, event response, root cause, organizational learning
    - Communication, Documentation skills.
    - OKR and prioritization - Why did we build X? What business goal does it solve? What business or technical assumptions were made?) You provide your team with the necessary support to take responsibility for their systems end-to-end (design, code quality, system health). You are mindful about the consequences of short-term solutions and make sure their impact on the long-term architecture is properly assessed.

Sample rounds with questions

People Management Round:

How big is your team? Can you describe the structure of it? Have you grown it or inherited it?

Q1: Tell me about a time when you invested in an employee's development.

What did you invest in and why? What was the outcome? Can you share an example where investing in an employee's development didn't work out? Were they a star performer? Why did you decide to develop them? What career path did they want? What did you do to enable them? How did you decide the areas that needed developing? Where are they now? Did they get promoted? Are they closer to their career goals? What im- pact did this development have on the business?

Q2: Tell me about a time when you had a low performing individual on your team.

How did you deliver feedback? Did their performance improve? What was the initial problem? How was this identified? What specifically did you do to help? Did you understand what was causing the poor performance? How was this tracked? Did you succeed in turning it around? What's happened to that person now? How are you monitoring the situation?

Q3: Have you managed someone out?

How did you become aware of the problem? Did you engage HR? How did you manage the process? (dev plan then PIP, what documentation) How did you reduce risk to the team during the process? Would you have done anything differently before or during the "managing out" process?

Q4. How did you measure productivity of your team? What steps did you take to fix?

Did you have clear metrics to indicate impact driven by your team? At what frequency - per quarter, per 6 months, per year - why? What was the outcome of the steps you took? How did that affect the team's happiness?

Evaluation Checks:

You independently manage a team of SDEs. You hire the right mix of SDEs to accomplish team goals. You are able to as- sess SDE performance. You have experience managing both high and low SDE performers and take effective action addressing employee concerns. You are able to hire, develop and promote SDEs. You delegate appropriately, assigning the right level of tasks based on the strength and development objectives of the individuals on your team.

Blitz questions:

- How do you draw new thinking and innovation out of your team? Give an example of how your approach led to a specific innovation.
- Have you ever tried to do a job for which you were unqualified for or in over your head? How did you react at that time? Were you able to accomplish your task?
- What makes a good software manager?
- What are your north star metrics currently at work?

System Design round

Example Questions:

1. A service that allows runners or cyclists to record their performance and compare to other athletes (this could be like Strava, MapMyRide or a gym tracker)

2. A food ordering service like Swiggy or Zomato.
3. A delivery or returns service for e-commerce orders (like Amazon locker)
4. Google Search typeahead.
5. A streaming service like Hotstar.
6. A high concurrency booking website like IRCTC / BookMyShow
7. A payment service.
8. A distributed crawler.

Example question sequence:

Let's design <one of the examples above>. First of all, what basic components would you use to build this service at scale?

A typical candidate will ask what the scope of the problem is and the expected customer demand. They will then identify the client platform, mobile device or web platform and describe a web-service in the cloud. Did they make a back-of-envelope calculation? Probe with follow-up questions to see if they consider a global service, deploying on multiple regions. Leave the scope open and see if they ask clarifying questions and narrow down as necessary. Do they separate static and dynamic con- tent? Did they think about using a CDN? Did they describe a 3-tier architecture? What are the layer choices? What compute platform choices did they make? Did they consider a load balancer? Did they consider an elastic compute? If they selected a serverless solution, why is it a good choice? What is the communication between the client and the service?

Question: What solution would you use to store the data? How would you model it in that store?

Candidates will typically choose a NoSQL Key/Value store or relational database. Probe if they make sensible choices for primary keys and rela- tionships if they use a relational database. Ask how they would construct the Key for NoSQL databases. If your question lends itself to a graph- based solution probe why did they make that selection? If they store customer data see if they consider GDPR (General Data Protection Regulation) [Only relevant when appearing for EU companies].

Question: How would you scale out the data store?

Candidates may sidestep the question and rely on the underlying technology. Ask how that works and the methods they would use.Probe if they understand what technology options are available, MongoDB, Cassandra or DDB are good NoSQL options. Ask how they would scale that solution regionally, replicate data or handle disaster recovery. For relational systems probe how they would shard or partition the data. Poke around the edges to see if they consider read-replicas and eventual consistency.

Question: How does this solution scale?

Candidates often say it will not scale without understanding why. Most examples expect higher levels of read vs. write traffic. A good candidate will recognize this and will flex their design to pre-process incoming data and use caching to improve the read performance. Probe what data they can prepare ahead to reduce request demands. How would they accommodate that in their design, does it require pre-processing and queuing? What kind of asynchronous or message bus solution do they use? What data do they cache and what are the compromises of this? Do they pre- process all the data? In the social media example, it's the network effect from the number of relationships. Ask how they would cost-effectively prepare the solution for peak/surge demand.

Question: How will you show results to the customer?

This part of the question relies on the example requiring collating multi- ple results and providing them to the customer with an acceptable user- experience. Candidates will usually approach it with a simple approach to generate results on request. Probe how they approach the query. Ask if it can be run in parallel. Do they need to search the full data-set each time? Does the solution require merging, table joins, sorting and pagination? Use this question to probe if they consider how their design choices consider customer usability.

Question: What metrics would you track for this solution?

Depending on experience candidates may identify operational metrics, code development or customer experience metrics. Probe for all. Ask what they consider are the most important metrics to measure. We're looking for tickets, customer experience/business, traffic, errors, latency and host monitoring. Ask how they would identify issues.

Evaluation Checks:

Strength behaviours: (1) candidate understands system design concepts at scale (2) approaches ambiguous technical problems methodically (3) clarifies assumptions (4) can express and support opinions and trade- offs. (5) The system design question is difficult as a phone-screen and will challenge candidates to express their ideas without use of a white-board.

Concern behaviours: (1) not asking clarifying questions, diving immediately into problem solving (2) making assumptions on scale or performance that are unrealistic (3) mapping to a problem they want to solve, rather than the one you are asking (4) making tech choices without asserting why or considering trade-offs (5) word-salad throwing out tech buzzwords (6) not identifying bottlenecks (7) not flexing their design with new information.

TPM Rounds

Example Questions:

0. You're joining a new team; here are the burndown charts for the last three iterations. You're joining just in time for the retrospective for iteration #3, which is later today. What questions do you need to ask, and of whom, to understand what's going on with this team?
1. The site is slow. Why?
2. Come up with a model for an auction site.
3. Reducing waiting time at airport's security line by 10% and later 50%. You can take this question in many directions, but I like the PM'ing aspects that you can add to it. Most people fail to understand constrains to the problem: budget, resources, time, etc.
4. Designing customer address book for Amazon multiple lines of businesses, e.g., there are some interesting follow up questions about notifying of address changes and storing prior addresses
5. Suppose you have 4 interviewers for a college recruiting trip. You need to interview every candidate once in the first round, and 3 interviewers (the 3 that did not interview the candidate in the first round) need to interview all candidates in the second round. How would you manage the scheduling?

6. You are given two separate web access log files, say from Apache or some similar web server. The first log file is from day 1, the second one is from day 2. The web site is set up in such a way that every access logged will contain a unique identifier for that customer somewhere in the request line. The website is a fairly high traffic, so these log files are very large. Question - How can you find a unique list of customers who visited on day 1 and then came back for a visit on day 2?

7. Design a system for storing log entries for a distributed web application

8. How does a browser talk to a server?

9. What kind of architecture could allow for flexibility where there are many cascading services each with hardened APIs and similar data passed from one service to another?

10. how would you define project management? Describe a project that was difficult to manage and tell me how you worked through the difficulties.

11. Find the largest numeric palindrome made from the product of two 3-digit numbers

12. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?

(a) Use one class, Car, with three instance variables: int numDoors, boolean hasAir, and double milesPerGallon. (b) Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon. (c) Use a class Car with three subclasses: Doors, AirConditioning, and MilesPerGallon. (d) Use a class Car, with a subclass Doors, with a subclass AirConditioning,with a subclass MilesPerGallon. (e) Use three classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car.