

Interviewing at Amazon — Leadership Principles

 medium.com/@scarletinked/are-you-the-leader-were-looking-for-interviewing-at-amazon-8301d787815d

Disclaimer: I'm not representing Amazon in any way with my posts, opinions written here are strictly my own.

As a bar raiser at Amazon (a little googling will answer what that is if you're not aware), I've gotten the opportunity to interview a lot of people. I've interviewed both very senior and very junior folks. Bar raisers do interview across all job families, and I've interviewed people for some pretty crazy positions. However, most of my interviews are for technical positions. Jobs like technical program managers, software development managers, quality engineers/managers, and of course the highest volume: software development engineers.

There are plenty of web resources regarding how to pass the functional (job skills) part of interviews. In fact, when I have friends and relatives interviewing, I send them to the internet to figure out what types of questions to prepare for rather than spend my time explaining. This type of preparing is pretty straight forward. You need to be good at your job, and make certain that the functional knowledge about your job is pretty fresh in your head. Perhaps how Hashtables work, or a bit about K-means clustering.

On the other hand, there isn't much on the internet about the other side of the interview process. When we at Amazon interview candidates, we're looking for a combination of those functional skills (coding, design, program management, 3D modeling.. you name it), and leadership skills. This applies to every potential new hire at Amazon, from experienced vice presidents to brand new college graduates. We literally assign specific [Amazon leadership principles](#) to each interviewer in the process. Those interviewers attempt to get a picture of how likely you are to be the type of leader we're looking for.

This article was written for the specific purpose of preparing you (a little bit) for that half of the interview. It's my version of giving sample questions. I don't know what you'll be asked, but I know what we're

looking for. If you can be what we're looking for, we'd love to have you.

Amazon Leadership Culture

There are quite a few articles out there about what Amazon's leadership is like, how our leaders act, and what it's like to work at Amazon. There are mentions of continual innovation, cut-throat competition, and fast paced projects.

As an Amazonian, what I always tell candidates is that there is not much about working at Amazon which is consistent across groups. We don't do much the "Amazon way", because very little is centralized. We have the leadership principles which guide how we act. Otherwise, every group acts like a little startup. They establish their own processes and best practices. They build an organization and way of doing things uniquely their own, while still following our leadership principles.

Considering how little we have centralized, we use the bar raiser group as a type of glue across organizations. We select bar raisers from the pool of experienced folk at Amazon, not just those who can interview well, but more importantly—those who deeply understand our leadership principles. As bar raisers, we then try to hire people who can understand and act on our principles. Finally, we set them loose into the chaos which is Amazon, with an assumption and belief that hiring people who follow our leadership principles will lead to long term success.

Understanding the Leadership Principles

When I have friends or relatives (or friends of friends, or friends of friends of relatives) ask how to prepare for an interview, I always suggest they read the description of the [Amazon Leadership Principles](#), and think hard about each of them. More than any company I've worked with or heard about, we use those principles on a daily basis.

We obviously hire based on the principles. We give both positive and negative feedback which reference the principles. We are encouraged to be aware of our own successes and failures in relation to the leadership

principles. I know I've certainly referenced a leadership principle or two while talking about parenting techniques.

I've read many thousands of interview transcripts, and it's often glaringly obvious which candidates have really read and grokked what the leadership principles mean, and those who either neglected to prepare for their interview, or simply didn't understand.

Note for the below sections. The quotes I'm putting in are actual snippets of interviews I've had, with close to literal quotes. Yes, they're extreme examples. I'm using a blunt instrument to make certain you know what I'm talking about.

Another quick note before getting into specifics. We interviewers are spending our time talking to you—the candidate—in hopes that you'll be hired. It's a big investment of our time, we don't want you to fail. When we say something like *"Well, that's a good start, what else?"*, it's very rare that the right answer is *"Um.. nope, that's it"*. Please listen carefully to what your interviewer is saying. Again, we're here to help :)

Customer Obsession

Leaders start with the customer and work backwards. They work vigorously to earn and keep customer trust. Although leaders pay attention to competitors, they obsess over customers.

Always one of the favorites to include in interview loops. I suspect as a principle it strikes so many candidates as obvious that they don't realize what we're really getting at. Every decision of consequence at Amazon involves the customer. Not just how they'll react, but what is actually best for them.

| What factors would you consider when deciding which offer should win the buy-box on the retail website?

| *"Maximize profit"*

| What else would you consider?

| *"Volume of sales, conversion rate, margin, etc, but it all comes back to profit."*

When interviewing research scientists for example (per the above), I've had some repeatedly insist that the right thing to do is to use algorithms to maximize profit. Even when I give a hint such as *"What unintended consequence might happen if you only optimized for profit?"*, they often miss the customer experience ramifications.

The thing we're looking for is that you consider and care about the customer. We've regularly made decisions at Amazon which lowered profit/sales, because it was the right thing to do for customers. We all understand that the company performs better if our customers (and our employees) believe it's acting in our best interest. We believe that's the right thing to do, we make those decisions all the time, and we reward people for it. We need to make sure new hires will do the same thing.

| For that feature, who was the customer?

| *"The product management team gave us requirements."*

| But who was the end customer?

| *"Well, the marketing team I suppose, because they wanted marketing slots. Oh! And the sales group too."*

| But who was the customer?

| *"I don't understand"*

It's somewhat shocking how often people buried in large companies don't mentally identify their customer as the final external customer. Their customer is their boss, sales, marketing, etc. They are so focused on doing what they're told, so focused on building what they're asked, without taking a big step back to understand who uses their product.

| *"I wrote the contact-us form for customer questions, but then I was watching the questions coming in (out of curiosity), and I saw so many of the questions had easy answers. I added a simple FAQ at the top of the contact form—after asking the product lead if she minded. It reduced usage of my form by a lot, which I'd call a win."*

We don't need you to exhibit our leadership principles at your current job (we understand that other companies are different), but we do expect you to be able to demonstrate customer obsession in your answers. Preferably you'll know who your customer is, their needs, what they really want from you (outside of your specific tasks), and think about solving their needs, not just tasks.

Ownership

Leaders are owners. They think long term and don't sacrifice long-term value for short-term results. They act on behalf of the entire company, beyond just their own team. They never say "that's not my job".

This is perhaps my favorite principle, because I think it is so core to how Amazon operates internally. You can be confident that you'll be asked questions to assess if you can act and think as an owner. This leadership principle is core to defining how our groups and company is organized. Slight digression to help you understand our model.

At Amazon, almost every organization / team / person can identify with something(s) they own. For example, I currently own the technology for Amazon FreeTime. By definition, that means all technology decisions related to the product end up in my court. As an owner, that means that I am responsible when something goes wrong, responsible when decisions are made, and I act in all ways as an "owner" of the product.

One of the managers in my group is the technology owner of the Amazon FreeTime Android application. This is still within my group, but he's expected to act as an owner, be responsible when things go right/wrong, and so on. You'll notice there are now two owners of the application. Of course I fully expect the engineers on the team to feel the same way.

And finally, if someone from AWS S3 comes to me and says they found a problem in my Android product, I'm going to be thrilled that they care enough. I'll invite them to my office, see what they found, and discuss how we might fix it. Because we're all owners and we all care, no one should ever say "*That's not my job*".

I've pushed for better HR policies. I've spent hours chasing down the owner for some random feature on the website to ask them to fix something. I fully expect everyone I work with (and hire) should demonstrate the same attitude.

When that happened and your whole application went down for a day, how'd you get it back up in the end?

"Well, we have an ops team which runs the website, so they eventually figured it out."

Did your developers help them figure out what had broken?

"No.. that's not our job."

I'd like to mention again, we don't expect your company to use the leadership principles. We don't expect you to have acted the way we'd expect someone at Amazon to act. However, let's take a step back. You're in an interview, and you know we care that people act like owners. What *would* you do as an owner in that case? Regardless of how you want to handle it in the interview, you need to make it clear to us that you know what the Amazonian thing to do would be.

*"I was their QA and they shouldn't trust me too far with the code because I barely know what I'm doing *laugh* but when they had a few bad weeks and were getting behind on the bug queues, I felt I had to help. I started spending a few hours a day fixing bugs instead of testing. They were pretty simple bugs, but I felt it was a better use of my time, rather than finding bugs no one had time to fix. I just did my best to document things really carefully to make sure my help was a net benefit."*

I remember once someone suggesting that we shouldn't write to candidates explaining our leadership principles, because they'd be able to prepare and fool us. My response was that an interview candidate who understood what we were looking for so much that they were able to trick us, was exactly the type of candidate we'd love to have at Amazon.

Invent and Simplify

Leaders expect and require innovation and invention from their teams and always find ways to simplify. They are externally aware, look for new ideas from everywhere, and are not limited by “not invented here”. As we do new things, we accept that we may be misunderstood for long periods of time.

Amazon is all about innovation. We are regularly doing things people have never done before. New scale, new products, new platforms. One of the most frustrating responses from anyone who has an open question is “I can’t think of another solution”, because there is **always another solution**. It’s just possible we’ll need to invent one.

The other half of this principle is the idea of simplification. The KISS principle is enshrined in our principles, because we all firmly believe that the simplest solution is the one you can maintain, the one you can iterate on, the cheapest one to build, and so on.

Minimal viable/lovable product is a concept firmly entrenched in how we do things, because so much about our ability to innovate, pivot, change our mind, build new things, is about doing things as simply as possible.

| Ah, so you use wishlists. If you were in charge, how would you improve the wishlists feature on Amazon?

| *“Oh, I’d fix the drop down to sort the lists in most recently used order.”*

| Ok good, what else?

| *<Long pause> “Hm.. no, that’s about it.”*

We can never be done with possible answers. **We all have the capacity for infinite ideas**. We need people with a capacity for infinite ideas, because frequently the first 7 ideas we had won’t work, and we’re going to need an 8th. We need people with the capability to look at other teams in Amazon who had similar problems, solutions in the industry, solutions used in other industries, books on the topic, etc. We need the question of “*What else?*” to be met by such a long stream of answers that it would take us forever to try them all.

| When you realized the project review meeting wasn’t useful, how did you fix it?

“I canceled the whole thing. I realized the status we covered was already in our ticketing system, and I’d rather everyone spend that hour working. When in doubt, I always remove processes rather than fix them.”

Experienced software engineers know that the best engineers often remove code when solving problems rather than adding code. Removal is always a better option, when it comes to code, systems, and processes. Simplicity allows for faster and cheaper innovation, and that’s why they’re married together in this principle.

Are Right, A Lot

Leaders are right a lot. They have strong judgment and good instincts. They seek diverse perspectives and work to disconfirm their beliefs.

It’s certainly not about being faultless. I’ve heard people quip that being good at this principle is the difference between being right 55% of the time instead of 50%. While we’re obviously listening for someone to be intelligent while we interview them, that’s not the core of this principle.

We give our leaders (in other words, everyone) a scary/amazing amount of authority to independently get things done. *“Get forgiveness instead of permission”* is a popular quote, because we need people to just move. In return, we need them to both have good judgment/instincts, as well as question their own decisions and be open to counter opinions.

| When she disagreed with your design, how did you know yours was right?

| *“She was very junior, she didn’t have as much experience.”*

| Sure, but how did you know she was wrong?

| *“My instincts are really good.”*

We like people with good instincts. But we absolutely require and love people to be open to being wrong. Right and wrong are a matter of knowing what you are trying to accomplish, what the options are, and how to compare the merit of the options. Right and wrong are never a matter of who is more senior or who talks the loudest. We need our

leaders to recognize that **every perspective and opinion needs to be valued**. And someone disagreeing with you is wonderful. It gives you a chance to reexamine your viewpoint.

Learn and Be Curious

Leaders are never done learning and always seek to improve themselves. They are curious about new possibilities and act to explore them.

Amazon has grown in incredible ways. What used to be considered a high volume service is now a laughably small one. The large organizations of years ago are literally 20x the size they used to be, or more. From those who have been at Amazon longer, you'll hear quotes like *"When I was first hired, this group was at 40 people! Now it's at 700!"*

| *"So for the team considering me, what language is that service written in?"*

| I believe it's Java.

| *"Well I only know C++, is there a C++ team I could work for?"*

We've all decided that what worked yesterday is not going to work today. Your code will need to be replaced, the design will change, the customers will demand more. Our services written before in Perl moved to C, and many of our C services moved to Java, and our Java services will continue to be re-written in whatever works best next. Everything changes, and so we need to as well. We place a large emphasis on everyone being open to learning new things, inventing new things, opening new doors, and being interested and inquisitive.

| Tell me something interesting you've learned recently.

| *"Rust! Have you ever programmed in Rust before? It's so interesting. So here's the differences between Rust and .."*

We don't **need** you to be obsessed with the latest language or newest tools or the hot new technologies in your field. We certainly don't need you to know the language we're programming in today. What we do need is for you to be open and interested in learning new things. Our leaders need to be open to new things, because that's always the path forward.

Hire and Develop the Best

Leaders raise the performance bar with every hire and promotion. They recognize exceptional talent, and willingly move them throughout the organization. Leaders develop leaders and take seriously their role in coaching others. We work on behalf of our people to invent mechanisms for development like Career Choice.

As a bar raiser, I'm heavily involved in this leadership principle. Hiring people who are interested in growing, and then helping them learn is a huge aspect to the leadership job. Managers hire for their teams, individual contributors interview people, and everyone is responsible for making certain we're letting in the right folks. That's the whole point of this article right? Making sure the right folks have the necessary information to get in :)

Each aspect of this principle is critical. For example, can you recognize strong performers?

| How do you manage your top performers differently?

| *"I don't. I think everyone deserves equal treatment. I think everyone is great in their own way."*

It's a lovely sentiment, but it's something we don't believe as a high performance management culture. We believe that the top performers need our attention and guidance to ensure that they have the opportunity to provide their very best at Amazon. Spending time on our top performers is the best use of a leaders time. We do recognize exceptional talent, and believe it's critical that our exceptional talent receives the attention they need.

| So since she was so successful, what'd you have her work on next?

| *"Honestly, I sent her to another group. They needed a new project lead, and she was awesome, so I offered her to them. As great as it was to have her on the team, this was much better for her."*

Internal transferring at Amazon is an amazing and open process. We freely move around the company, and are able to self select our next job when we feel we need the growth. The very best leaders encourage others to always be stretched. If you're comfortable, you're not growing. Because doing things which challenge and scare you is the best way to learn. And so in the interview process, ensuring the growth of those who work with/for you is a critical talent we listen for.

It sounds like you're more senior than all your co-workers. When they made all those mistakes, how did you help them?

"I was available if they had questions. If they didn't want to ask for help, it's their own fault."

We believe it's the responsibility of **every single person** to help others grow. When our college hire software development engineers join, they may spend an awkward first few months figuring out how things work. However, 3 months later, that engineer is often the one who helps ramp up the next hire. Because they are the ones with the most fresh memory of the process (and pain) of joining and figuring things out. I love that we give the uncomfortable and new job of teaching to everyone immediately, because it's a critical skill that **everyone** needs to learn.

Insist on the Highest Standards

Leaders have relentlessly high standards—many people may think these standards are unreasonably high. Leaders are continually raising the bar and driving their teams to deliver high quality products, services and processes. Leaders ensure that defects do not get sent down the line and that problems are fixed so they stay fixed.

This principle is tied heavily to the ownership principle. If you're an owner, do you want things to be low quality? If you're an owner, would you be embarrassed or proud of the quality you're delivering? In the same way, we expect our leaders to be proud owners of the things they own, and as such, insist on standards which everyone (including themselves) should struggle to meet.

It certainly doesn't mean we expect long hours, and "*driving*" doesn't mean beating people up. What it means is that we should never be satisfied with what we have. Look at the industries and companies which stopped innovating. You innovate when you know you're not done, and we insist that everyone has such high standards that we're **never done**.

| How did you know what you had delivered was what the customer wanted?

| *"The product team accepted the feature, which meant we delivered their requirements."*

| But was it the right features? Was it the quality you wanted?

| *"Well, it was good enough to be accepted, that was our goal."*

Our goal is never to deliver to requirements—to only meet our goals. At all times—once you look past our official project goals—our goal is to improve. Our goal is to never accept that something is broken, to never feel that anything less than a perfect product is acceptable.

| Then how did you fix the overcharging bug?

| *"We didn't. It's annoying I guess, but no one noticed, and we make more money. It's a feature not a bug right?! Hah!"*

(I know some quotes are a bit over the top, but so are some answers in interviews.)

We care about fixing the root causes of problems (not just hiding or patching them). We care that you are always thinking "*I know we can do better than this*".

Think Big

Thinking small is a self-fulfilling prophecy. Leaders create and communicate a bold direction that inspires results. They think differently and look around corners for ways to serve customers.

Imagine the first person at Amazon who said "You know, I bet we could make a store where everything is checked out automatically by just picking up an item." I love that thinking big is such a part of the Amazon

DNA that we have it written in our leadership principles.

| If you owned marketing for our kids tablets, what new things would you look into?

| *“Well, I’d look at our creatives and A/B test which images work best.”*

| Sure, but what other ways do you think you could market our product?

| Anything else?

| *“No, that would probably work.”*

Leaders here never accept that our vision is big enough, innovative enough, or that we have high enough standards. If we are asking you in an interview for an idea, you can’t say the first small idea which pops in your head, and believe you’ve satisfied the requirement. When you’re asked for ideas, you should err on the side of being asked to stop. We’re always looking for bold vision (perhaps grounded in a small amount of reality), and need to know that our leaders are open to the idea of thinking bigger than the thing in front of their face.

| What other ideas do you have to improve the product detail pages on Amazon’s retail website?

| *“Ok, going a bit more experimental, what if we eliminated detail pages? You could just input your requirements to Amazon somehow—like you want a TV by Tuesday that’s at least 60” with some certain resolution and a certain maximum price, and then Amazon would pick it out for you and just ship it. Maybe we can eliminate customers picking out items completely.”*

When I’m interviewing someone, I prefer to think that someone’s idea is somewhat impossible, because explaining constraints to a co-worker is easier than explaining how to invent and think.

Bias for Action

Speed matters in business. Many decisions and actions are reversible and do not need extensive study. We value calculated risk taking.

We strongly feel that making some decision is much better than making no decision. We assume that we hire smart people who are right a lot. We want those smart leaders to make quick decisions and move forward. You learn more from doing things and measuring rather than surveying or testing or projecting results.

Since the site was down and you thought it was the cache, did you try clearing it?

“Well, I guess I could have, but I didn’t want to risk being blamed for breaking something. It was much safer to wait for my manager to make the decision, it was above my pay grade.”

I know at your company it’s possible that you’re encouraged to not take action unless asked, but we find that unacceptable. You should know what we’re looking for and expecting. We want leaders who take action, are bold, are willing to take risks, because they know it’s the right thing to do.

For software engineers, it’s a bit of a running joke about how many months it takes someone to completely break the service/system their team owns. Because it’s a part of being handed dangerous tools and responsibilities. For almost every major human-caused accident at Amazon, there’s a sheepish person who raises their hand and says *“Yeah.. that was me”*. And they’re still here, because we value calculated risk taking.

How did you decide between the two technologies?

“The other engineer and I discussed it for a couple hours, figured out what we disagreed on, and I suggested that neither answer was necessarily better. She agreed. I said that unless she had any new info, I’d rather we just pick mine, and we could always come back to the decision later if we learned something new.”

We value this type of risk taking, because we need to move fast in areas where we can’t possibly know the right answer. Option A or Option B? Pick and move on.

Frugality

Accomplish more with less. Constraints breed resourcefulness, self-sufficiency and invention. There are no extra points for growing headcount, budget size or fixed expense.

This leadership principle is perhaps the most joked about within Amazon. There's always a risk that frugality (taken to its extreme) ends up feeling a bit "frupid". However, once you get past the fun of joking about the principle, it's also a core element of how Amazon has managed to be successful.

Look at the nature of how Amazon does business. AWS constantly drops prices to ensure it's competitive (and almost always cheaper) than doing things on your own. Amazon devices (such as our tablets) are shockingly cheap compared to the value they provide. Amazon retail is well known for valuing low prices and pursues efficiency aggressively. This principle is all about ensuring that we can provide as much value for as little cost as possible.

| Why did you get into an argument with him?

| *"Because he didn't give us enough people for the project. I told him it needed 5 people, and he said he'd give me 2. I said things cost what they cost, and if he won't pay, we won't do the work."*

Things never cost what they cost, because we can always do less, do it cheaper, do it slower, do it with lower quality. We always have options. We almost never have the resources we feel we "*need*" at Amazon. And that does amazing things for the products we build. I know the process can frustrate people, but pay attention to the language in the principle. We're **inventing** to solve problems in a cheaper way. Which often is faster, easier to maintain, easier to extend, etc.

| Then how did you fix the lead generation tool?

| *"Actually we really didn't have the time to maintain it or rebuild it, but they really needed a tool to track their work. I realized that our bug tracking system had most of the features they needed. Certainly wasn't perfect, but I explained how they could use it, and it worked out fine. One less system to maintain."*

When you're interviewing, we're looking for you to understand that having not enough time or resources is a fine and expected thing. Having too much to do, and too little time is perfectly ok. Not doing everything you want on a project is healthy, because doing everything you ever wanted to do is inefficient. The last thing on your stack rank will never be done, and we should all be happy with that.

Earn Trust

Leaders listen attentively, speak candidly, and treat others respectfully. They are vocally self-critical, even when doing so is awkward or embarrassing. Leaders do not believe their or their team's body odor smells of perfume. They benchmark themselves and their teams against the best.

The most commonly missed aspect of this principle is the “*vocally self-critical*” aspect. Probably because when you're interviewing and telling us how awesome you are, you might feel challenged to explain that you're not awesome all the time. However, with big power comes big responsibility. It's very important for a leader to be able to recognize when they've made a mistake, so that they can correct it quickly. This “*earns trust*” with co-workers, as well as ensures that future mistakes can be corrected quickly.

| Tell me about a time you made a serious mistake at work.

| “*Hmm.. I can't think of anything.*”

What's great about that quote is that I've heard it dozens of times. However, it might not be as instructional as I need.

| Tell me how that project you were leading failed.

| “*Well, the product manager had given us the wrong requirements, and then one of my engineers screwed up the metrics gathering. Then when we launched, marketing didn't notice it was under-performing, and kept advertising the broken product.*”

We expect leaders to always recognize that they have the power to improve things. It was someone else's decision? *Then you failed to influence them.* It was someone else's error? *You failed to notice the mistake fast enough.* When you're explaining a mistake, first recognize your own, before explaining other misses.

| How did the marketing team send the emails to the wrong people?

| *"Well, first we didn't give them the necessary tools, so they were forced to rely on manual mail merges. It wasn't fair to expect them to avoid errors. My team should have warned them about the risks, and we could have built better tools.. in fact, we're in the process of building them now."*

The best leaders look dispassionately at the details of a situation and address it as a leader at Amazon. It's not about pointing blame, it's about fixing processes. Mechanisms for the long term rather than being focused on the short term incident.

As an interviewer, think carefully about how you deal with mistakes, both yours and others. We care about long term, not short. We care about fixing problems, not blaming. We care about honesty, not a story. We care about respecting those we work with, because we're going to work together for a long time. Partially because we're so decentralized, it's critical that we trust those who we work with, because each of us have a lot of power and responsibility.

Dive Deep

Leaders operate at all levels, stay connected to the details, audit frequently, and are skeptical when metrics and anecdote differ. No task is beneath them.

"Trust yet verify" is a favorite phrase at Amazon. We care deeply that leaders keep a careful eye on what they own, and know ways to audit their space. If something doesn't make sense, our leaders need to have the ability (and interest) to dive in and figure out what's going on.

| Your team reduced latency on the website? How?

| *"We removed bottlenecks."*

| What bottlenecks?

| *“Code ones.. I’m not really sure, it was one of the engineers on my team.”*

I’m not suggesting that you claim knowledge over every detail of what you or your teams have worked on, but I will say that you shouldn’t tell a story where you don’t know the most obvious of details. If you increased traffic, you better know what you did. If you dropped error rates, you should know what the errors were. We care that people know details, because it means they cared enough to question things. We need curious and skeptical leaders.

| How did you reduce the memory footprint?

| *“It’s funny, we had to restart our system every few months because of a memory leak. I’d just joined the team, and the mystery annoyed me. So I spent a few hours reading very carefully through our memory management code, and realized that we incorrectly double-parsing in one of our helper functions. Not only did it save on memory, but CPU as well. It’d been there for years.”*

I love when I ask questions of people, and they can go 4 or 5 levels deep, and keep getting more excited because the details are actually interesting to them.

Have Backbone; Disagree and Commit

Leaders are obligated to respectfully challenge decisions when they disagree, even when doing so is uncomfortable or exhausting. Leaders have conviction and are tenacious. They do not compromise for the sake of social cohesion. Once a decision is determined, they commit wholly.

I particularly love this principle because I feel it’s such a clear departure from the natural culture at other companies. I believe this is true, because in interviews I’ve been in, one of the most common quotes is *“Because my boss said so”*. We never find it acceptable to do something because your boss said to. We do things because it’s the decision we’ve made, and with the data we’ve gathered, we’ve committed to that plan.

“My boss had insisted that React Native wasn’t responsive enough for our interface, so I made a quick prototype of our UI and showed him how it was just as responsive as our existing product.”

Data based disagreements are great. We use them all the time at Amazon, and examples of when you gathered data to make an argument are great. What you have to be careful about is that people end up in the realm of disagree & disagree (instead of the commit part).

“I showed my boss that the design wouldn’t work, she refused to listen. I explained to the other engineers that it wasn’t my fault if things break, which of course they did later.”

As a leader, we can never feel (and certainly not express) glee in the failure of others. Cheering at others failures suggests that you’re playing a zero sum game, and the game is not zero sum within Amazon. We believe that we’ll all succeed if we’re all on the same team. “I told you so” is never acceptable. We give data, make decisions, and live with the consequences as a team.

“I showed my boss the data, explained why option B was better, she pushed back, I suggested that we could try both options, she still disagreed, and asked us to move forward. I told my team we were building option A, and made sure to explain why it was the right choice. Because my team needs to know that we’re all on the same page.”

We need leaders who are able to recognize when disagreements are needed, and when they’re not. What’s worth arguing about, and what’s not. What’s worth escalating, and when it’s time to move on and start delivering. And we need to believe that you’re that type of person.

Deliver Results

Leaders focus on the key inputs for their business and deliver them with the right quality and in a timely fashion. Despite setbacks, they rise to the occasion and never settle.

In the very end, Amazon employees need to deliver value. That one is pretty obvious. Take a look at a few key words, which are the ones we often dive into in interviews. “*Key Inputs*” is a good one. Do you know your key inputs?

| How did you know if your registration form was working?

| *“If it registered people properly.”*

| How might you go about measuring the success of a registration form?

| *“If it doesn’t have errors it’s successful.”*

We need people to recognize that they’re not accomplishing tasks, they’re focusing on inputs for their business. They’re not delivering registration forms, they’re delivering pipelines for new customers, which have conversion rates and perhaps customer contacts. It is much harder (and much better) to focus on the business itself, rather than completing tasks.

Also, *timely fashion, setbacks, and never settling*. We’re somewhat allergic to people being casual about date slips and deadlines, because per our earlier discussion around high standards, it’s critical that leaders always expect better. While it’s reasonable to miss dates (we all do, regularly.. certainly more regularly than hitting dates), we also can’t be comfortable with it.

| What’d you do after you realized you couldn’t hit the date?

| *“I looked at the features, and picked out a few I felt we could cut. The product manager agreed on a couple. Then, as bad as it sounds, I convinced QA to cut a week off the testing schedule. It was risky, but it was so important to hit the holidays. I figured we could test everything important, and it was the only way to get out in time.”*

You’ll notice our principle doesn’t say with “*perfect quality*”, it says “*right quality*”. Right quality is often less quality than we’d prefer. Going back to our frugality principle, we never have the time or resources to accomplish things the way we **want**, so we accomplish things the way we

need. So before you get yourself too wrapped up insisting on high standards, keep in mind that we're extremely pragmatic. And we're all about delivering the right results.

Final summary

I have to say that this ended up longer than I'd intended. I started with the intention of writing a quick note (in response to a couple particularly bad interviews), but I ended up with something a bit more extreme. I hope this proves useful to someone. And I'm hiring ;)