**Interviewer:**

So good evening. We are here at Startup 4. I would really love to hear a little bit more about the core service you have and maybe the team and so on.

**Interviewee:**

Where can I start? Where would you like me to start

**Interviewer:**

The core service that you offer?

**Interviewee:**

What do you mean the product offering?

**Interviewer:**

The product.

**Interviewee:**

So we offer a turbine inspection product right now. That was our original product offering for about two years ago. And so we come out as a service offering to our customers. We drop a drone on the ground. We hit go and we do campaigns of inspections, massive fleet-wide inspections for our customers so we'll do a couple thousand inspections every month. The entire time, what that means is what we're looking for is really a curated list of what people should be paying attention to in terms of the health of their fleet of turbines. So our end deliverable to our customers is these are the hundred damages across your fleet that should be paying very close attention to right now, maybe even turning off some of these towers. Going from there, helping them to design the exact strategy for how they go about repairing all of those and in what ways they do so.

**Interviewer:**

And maintainance. So you seem pretty big, but when were you founded?

**Interviewee:**

So we've been working on the core technology as a team for a decade at this point. We started the company back in 2012, but really started this form of the company. A couple pivots along the way, but this form of the company in 2017 is when we commercialized our product. Since then we've done 16, 17 or 18,000 inspections over the past two years. So a very rapid ramp up from two years ago when

**Interviewer:**

It seems you are growing fast.

**Interviewee:**

Yeah. Yeah. It's been exciting and hectic?

**Interviewer:**

Yeah, of course. How many employees do you have right now

**Interviewee:**

So in our Ann Harbor office, we have 30, I wanna say something like 35, I need to get an exact head count for

**Interviewer:**

Gender balance?

**Interviewee:**

It is a majority men we have. Six or seven women on the team right now. Okay. Out of 35 people. Honestly we've had a pretty hard time finding women who are interested in the roles. When we get applications, we have nine out of 10 people are male who are applying for these jobs. I see. So diversity is something that we definitely need to work on and are very interested in.

**Interviewer:**

What are your expect when you employee people like when you get this interviews, their profiles in development or skills?

**Interviewee:**

It's all merit based, it's entirely merit based our entire process. I would go so far as saying we have a meritocracy, but that would probably even undermine what we actually do here. Okay. It's a little bit jarring when people start because our thought process is that an idea speaks for itself and that we all just kind of represent ideas. So if let's say you and I are talking about some weird novel inspection technology. I would argue one side of it. You would argue the other side of it, just for the sake of representing both sides and then maybe halfway through the argument we'd switch. And it would just be like, it's almost just an exercise in finding the best idea. And so ideally the best idea just wins in all debates and arguments. It doesn't even really come down to credentials in that regard. Okay. Like it doesn't matter what your background is or anything like that. It's the idea that speak for themselves. Yeah.

**Interviewer:**

So if we can go a bit more specific to software engineering stuff. What kind of practices do you use to develop the systems?

**Interviewee:**

Really broad question for us because we do everything from, I mean, it's hard to, to separate hardware and software when you're talking about robotics, because they're so closely intertwined. For our physical drone product, we do the software development all the way down to the circuit board level. So we're writing the firmware that runs our boards that are doing highly precise timing and measuring of signals. So down from that point all the way up in terms of technology stack all the way up to front NUIs.

Interviewer:

Just to narrow this down a bit, when I meant practices, it's more like are you using sort of what agile practice, waterfall practice?

**Interviewee:**

I think the thing that it varies most based on is timeline of projects. So when you're talking about hardware projects, they can sometimes span six months. And Agile's a little hard to do in that regard because you need to be thinking six months ahead because that's how long it takes to design, make a circuit board, code it up, test it out in the field. Like it's a long range.

**Interviewer:**

When you say code it up, is that hardware level or is it firmware level?

**Interviewee:**

Both.

**Interviewer:**

Both. Okay.

**Interviewee:**

Both. Yeah. We do board design and routing, schematics, everything in house. So it's across that.

**Interviewer:**

Agile is not the best option.

**Interviewee:**

So we stick to like almost a semi-agile. So we do OKRs here. I don't know if you're familiar with OKRs. It's Objectives and Key Results (OKRs). I think it's Intel founded originally. Google uses it. A lot of the big tech companies use it. We find it to be useful that we do these OKRs on a quarterly basis. So we will meet up as a team. Actually, we're gonna be starting our next one on Monday. We'll meet up as an entire company and decide on at a high level, what do we wanna get done over the next quarter? And then we kind of trickle it down to the various departments of saying, how does each department contribute towards getting this done? So it might be developing a new inspection technology or repair technology or workflow for a customer or a new cloud tool, whatever that might be. From a product development standpoint, it might be a new type of marketing campaign or thought leadership role or something like that from a sales and marketing perspective. Yeah. And then that trickles down to each department. And so each department then takes, it's usually two weeks at the end of each quarter that we do each department takes those two weeks and figures out what it means for that department to contribute towards that goal. And then also what that goal missed. So we like to say the expression goes, you don't hire smart people to tell them what to do. You hire smart people, so they tell you what to do.

**Interviewer:**

Was it Steve jobs that said this?

**Interviewee:**

I think that was either Bill gates or Steve gates jobs. I think it was a Gates quote. So for all of our engineers, they oftentimes know way better than any management, what needs to happen in terms of like testing procedures and art cloud architectures and new technologies that we should be using and all that kind of stuff. So we try and have a, a hybrid of bottom up top meets top down of where.

**Interviewer:**

Well, when it comes to testing can you mention some of testing practices for software? I wouldn't say specifically at low level, because low level testing for hardware, it's a bit sort of different when it comes to, but maybe you can mention integrated testing.

**Interviewee:**

So we do unit testing, integrated testing kind of sandbox testing. Massive entire system testing. Stress testing, all that kind of stuff. And it varies based on what we're talking about. So if you're talking about a cloud technology, we'll do everything from we don't do actually, we don't do like a heavy amount of unit testing. I'd say we probably have 60, 70% coverage. It's not like we're not targeting the 95% coverage. We more employ than just move quickly.

**Interviewer:**

Why is that?

**Interviewee:**

I would rather have our engineers working on fixing things that are actually impacting people. What we do with this company is we've made our software, the center of our own business. And so all of our employees are using it. And so, as they're finding bugs, that's what we'll pay attention to fixing. And as when you go to fix a bug, the first thing you do is you're write the test, you write the test that replicates the behavior and then you fix it and you make sure that that test now shows that it's working. And so using that mentality, it kind of hones your testing into the places that matter. Cause otherwise that means that you're gonna be writing, testing on stuff you don't even know if it's relevant yet. Okay. So you wind up just saying, we fired this bullet as they say, we tested out this theory and no one cares about it. So we're just gonna delete it.

**Interviewer:**

So you sort of go into the operations that are mostly happening for the software and those specific operations you want to test.

**Interviewee:**

Exactly. Yeah. Specific workflows. I mean we are playing the invention game almost where we are designing a tool to work with workflows that don't yet exist. Okay. So before we came on the scene, people didn't really do inspections of their wind farms of their entire fleets. It was one-offs here and there. It was PDF reports and emails and spreadsheets and old hard drives. And so we're trying to design, we're trying to both create products that actually address the workflows of the people who are using the software. But at the same time you have misses, right. When we'll try out a workflow, people will say, eh, that doesn't really achieve what I want. And we'll say, okay, check that, iterate it. Let's keep going from here. Sure. So having full coverage of testing over all the things that we don't know are relevant yet doesn't really matter for us.

Notes:

**Software Testing**

Unit testing 60% coverage

They test for meaningful operations

Integration testing of hw and software

Simulation testing

No metamorphic testing but similar to their simulation and making sense out of the data

**Technical debt**

We accept technical debt can happen, take responsibility for it, and this is all about trade-offs we need to make in achieving deadlines. Our team is highly deadline-driven, but fully committed to handling technical debt

**Artifacts protection**

The main protection is to keep the source code in private repositories. This has originally been done since the beginning.

IPs would have meaning if big companies want to invest and protect the software, which they don’t have the capacity to do so.

Use of open source licenses is welcomed as long as the ones reusing and updating the code can give back to the community.

**Documentation**

The code should be highly readable and peer reviewed.

We all agree in our team that the code should be highly readable and peer-reviewed if we want to manage technical debt

Only critical or complex code is thoroughly documented.

They divide teams based on their skillset. This helps in maintaining and understanding code.

**Goal or resource driven?**

Company is mainly goal driven or problem solving driven and especially prioritizes deadlines. This has been their success factor.

**Technical skills**

Looking for highly motivated and skilled engineers. The company employee background ranges from undergraduates to PhD.

**Soft skills**

White board approach where a new employee is teamed up and has to solve a real problem in a real scenario with the team. His ability to work as a team member by being not to shy not to much of a show off, is important in the selection.

Acceptance towards criticism is also deemed important.

Or have a short two/three week probation period.

Most are successful.