**Interviewer:**

Yeah. Excellent. So the first question, this is just a checklist. Actually, the first part I need to understand is the core product that Startup 7 is producing just a few words. You don't have to talk so much.

**Interviewee:**

We make a platform for providing digital services for the aquaculture industry. Yeah, so we work with two main focus areas. We provide logistical planning and execution tools for transport at sea. And we make an information-gathering tool for fish health and fish journal control of the fish. Yeah.

**Interviewer:**

Yeah. Sounds quite exciting.

**Interviewee:**

And it's provided as a software as a service, licensed fee per month. Fixed prices usually,

**Interviewer:**

Is it like Microsoft cloud or Google cloud?

**Interviewee:**

It would; it could appear like that. So it's a platform you log into, and you have a map with all the resources, and then you have the functionality for actually doing a lot of stuff. Yeah.

**Interviewer:**

And when was the company established?

**Interviewee:**

I think that Startup 7 was originally founded in December 2012. Okay.

**Interviewer:**

Maybe this information is also on the website. You are located in Trondheim, I think. Yes.

**Interviewee:**

Well actually the head office is in Trondheim

**Interviewer:**

Yeah. It's

**Interviewee:**

But, but we also have people in Oslo.

**Interviewer:**

Okay. And what is your role in the company?

**Interviewee:**

I'm the CEO. Okay. But I'm also the tech. So at least for now, I'm also the CTO sort of.

**Interviewer:**

That's nice. So you deal also with the technical stuff.

**Interviewee:**

I actually develop stuff myself as well.

**Interviewer:**

Okay. No that's cool. It's nice to meet people that are also CEO and develop things. So, yeah, it's quite nice actually. What type of ecosystem? I mean, I imagine you have offices, so it's kind of that it doesn't,

**Interviewee:**

So we are located here in Trondheim and in Bergen, we are situated at co-working places. So I'm here at digs intron, and we are also having offices in Bergenworks in Bergen, which is also co-working.

**Interviewer:**

I think that's very good also because you share some knowledge with others, maybe. Yeah. How many employees are there in Ontario?

**Interviewee:**

So we've grown. So let me see today, so basically we are nine now.

**Interviewer:**

Out of these nine, how many is like development people

**Interviewee:**

Six or five and a half. I don't have time to spend all my time on development.

**Interviewer:**

But you have the competence to do to follow up with development

**Interviewee:**

And I do develop stuff every day, so.

**Interviewer:**

Do you sort of participating in, yeah, I think you said this, you do development, but of course, you may test what you develop and et cetera? Yeah. So it's a natural thing during the development process, so that's very nice. One quick checklist here. Are you all men, or is it like men and women?

**Interviewee:**

For now, we are only men, but out of the nine, there's one girl.

**Interviewer:**

That's good. This part is more a bit focused on the software engineering practices. So one question is like what software development practices tools are you using? And I think you started describing the software as a service.

**Interviewee:**

Yeah. So, but when it comes to the development we use, I mean, if you look at the actual tools that we use for development, I would say the main tools are GitHub. We use GitHub for both, of course the code and the revisions of the code, but also we use GitHub for planning. So we use the board that GitHub provides so that we have development pipelines for the different products that we have. Yeah. And since we have a pretty broad tech stack that we develop on. We basically develop in most tech stacks probably except Microsoft. So we are pretty heavy on JavaScript use frameworks, like view leaflet, Mapbox, and also on the backend side, we have some nodes, some Python. Yeah. We have Java backends. We have PHP backends, we use databases like CouchDB Postgre ES SQL Maria DB posts for map and geographical data. We operate most of our platform on Google cloud. So we use Google cloud we also utilize Kubernetes and Docker for operation and also development.

**Interviewer:**

Really cool. Is the team like an agile team?

**Interviewee:**

It's very agile, but we don't, we haven't married to any specific agility, or we do. We do something that would seem to be similar to scrum, but since we're all scattered around the country, we use a lot of online meetings, both Skype, Hang out, stuff like that. And slack. It's all about breaking down all the development tasks into small enough tasks, and then all the developers have to pick new tasks from the top of the stack. So I usually do the prioritizing for all the products that we do. And then, the other guys will just pick the next task to complete, and we do constant rollouts of all functionalities. So once it's complete or at least complete and roughly tested, we put it into production. Okay.

**Interviewer:**

So you have some hint of DevOps.

**Interviewee:**

Well, I actually put that responsibility on all the guys that develop. Because if they need to feel the pain for, when she hits the fence in production, they will also have to chip in and, and actually fix the problem. Yeah. So, I don't have a specific person to do that. We've actually set it up in such a way that, when you, when you merge the code into the master branch, our solution will automatically pull out the new branch. Okay. So instantly, when you merge into the master, it's available in production.

**Interviewer:**

Yeah. That's pretty good. Also giving the responsibility to programmers is also good, so they don't get away. Yeah.

**Interviewee:**

That's the only way a programmer learns how to be thorough when he develops because if he develops sloppy code, he will have to work during the night to fix the problem.

**Interviewer:**

We're gonna come to that topic. What are the most important quality attributes for you, like performance in your product like performance, security, reusability, or something else you might think of?

**Interviewee:**

I think that quality attributes for our side are actually to provide a measurable value for our customers.

**Interviewer:**

I understand that's a bit broad, but when it comes to software, does the software need to be fast? Does it have to perform well? Does it have to be secure for your customers as well? Like this data have some privacy issues?

**Interviewee:**

There are privacy issues. So we all, of course, have to adhere to security, not necessarily in what you would think of as security on code level or bank-level security. Still, our customers don't want to share their data with other customers. And since they coexist in one platform, at least separation of the customer data is an important issue. Since a lot of our users are operating at sea, it has to have some kind of performance because the footprint can't be too heavy. After all, very often, it's uh some kind of mobile network or bad coverage when they use it. Technically for me, it's important that we always have a keen eye for everything that we develop should be a product. So it shouldn't be specifically made for one customer. So we have to have reusability on everything that we make. Yeah.

**Interviewer:**

To be a bit generic.

**Interviewee:**

Because I don't wanna have to write a new version of the same product to get a new customer. I want them to use the same system basically. When it comes to UX and the interface. So we are operating in an industry where digitalization hasn't really been that big yet. So we find that it's very often better for us to have a very simple UX not too complex, not too fancy because many of our customers are actually a guy in the age of 58, which doesn't really appreciate all these blings and blanks in the product.

**Interviewer:**

I see, understand. It has to fit the customer's needs. And when it comes to testing, is it like you do manual testing, automated testing?

**Interviewee:**

No, we do all our testing manually. Okay. I've been developing software for more than 25 years, and I've been through all these test regimes and automatic code tests and stuff like that. If you write a function in any language where you take two numbers as an argument, and you add them together and then make a test to actually verify that they can add two numbers correctly together, that's not really a good test because it only proves that the code and the plus in any language actually works. Yeah. So for me, it's functional testing, and it has to be done manually because you have to understand data and the user and the context you have to use it in.

**Interviewer:**

Okay. And how much time does it take to do all this manual testing for the development team?

**Interviewee:**

I would estimate that for any functionality that we develop we would, if you include the testing in that development phase until rollout in production, we would probably spend somewhere between 7 to 15% of the time testing. Okay.

**Interviewer:**

That's interesting. And do you document the code? I guess, what is the approach?

**Interviewee:**

Most of the code is self-documented, or if we need specific documentation, we usually put it straight in the code.

**Interviewer:**

When you say self-documented, does it mean like it's readable code?

**Interviewee:**

Yea. We use markdown files on GitHub to sort of provide the documentation that you need as a framework to understand what this code does in generic terms.

**Interviewer:**

Yeah. Yeah. But I understand. If I write the code and if it goes to the next guy, he can still understand and update that code. Yeah. So it doesn't have to be dependent on the programmer or something like that.

**Interviewee:**

That. That's right. Yeah. Yeah.

**Interviewer:**

But the most important part of this interview actually it's about the technical debt and pivoting. You are a real engineer, so I don't want to explain to you either of the terms, but if you have any doubts.

**Interviewee:**

I understand.

**Interviewer:**

I believe so. First of all, is how much aware are you about the technical debt within your company.

**Interviewee:**

I'm very aware of technical debts, basically. I'm we operate with at least two typical debts. It's not just a technical one, but it's, it's technical with two sides because there's a technical debt when it comes to, when you fastly develop code, sometimes you develop and you make a decision at some point in time that I will not add necessary security or add necessary performance at the moment because it's more important for me to actually have the functionality and you have a debt that, you know, that you will have to fix sometime in the future. When it comes to technical debt given like if we were using a framework or something that sort of went out of stock we don't really have a lot of that. Of course, old Java code cannot run forever. So if you want to catch up with the Java stack, you basically have to do some migration work all the time. But it's fairly important for us to try to keep up with the versioning and sort of update our code with the new frameworks and the new versions of whichever code we use.

**Interviewer:**

That's right. Sometimes I like the idea, you just said now that, sometimes developers want to get the thing done, but then they don't think about, software engineering practice. The best practice is how to get it done right. Yeah. But, then you take some shortcuts that might have some, issues in the future.

**Interviewee:**

I'm pretty happy to do those shortcuts as long as we are aware of them. That's why I also try to keep a close eye on everything that we do so that we do know about all these things that we'll have to fix sometime in the future.

**Interviewer:**

But when it comes to, I mean, the follow-up question is what is your perception about technical debt? And here I have four keywords. Like, do you want to ignore it? Do you want to accept it? You said something right now. Yeah. Do you want to manage the technical debt or do you want to totally avoid it and say, okay, we don't want technical debt at all?

**Interviewee:**

No, I want to manage it because I do want to have some technical debt because it's a business decision where timing often is more important than perfection.

**Interviewer:**

Yeah. And also because the company now has been around for nine years. So how was it at the beginning? Do you remember a little bit and compared to now?

**Interviewee:**

Yes. In the beginning, we had multiple developers which were developing more on their own. That basically in the end gave us a technical debt that was almost impossible to solve especially when it comes to a couple of those mobile apps that we have because mobile apps are, is a nightmare when it comes to, well, it's gonna work on both iOS and Android and suddenly this or that framework wasn't supported anymore. Now it's a lot closer team that does the development we are in daily contact with each other and we sort of have a very open discussion on how to solve and how to do it and all the developers are very aware of how we handle or manage the technical debt that we actually bring into the code ourselves.

**Interviewer:**

Yeah. That's quite interesting. I asked this question now this is more about pivoting and technical debt. So this is like pivoting has a bit of concept here, like, did you change the direction of the company of the product, et cetera, but when it comes to pivoting we have identified from a research perspective. When I say we, researchers before me 10 types of pivoting, I would like to see if they can connect with technical debt that you might have faced, like in the beginning or now. And how did this affect the actual pivoting of the company product? So the first one is about zooming in like a single feature of the product that you have becomes the whole product itself. Let's say you have a broad product with many features, but then you realize a lot of those features are not even needed for the customer, so you just focus on one of them. And if this happened, then explain what could have been the role of technical debt and the same applies to the other ones that we can go one by one. I think it's easier to zoom out has the whole product becomes a single feature. As I said, mainly because the original point is sufficient to others by the customer needs. And then again, explain the technical debt. Did you change the customer segment, perhaps? You said something, we had some mobile applications that weren't supported at the beginning, et cetera. All the platforms changed. So it was very difficult to keep up with those implementations. Also because the customers didn't want the web up, but didn't want the mobile app, but they wanted the web up. Customer needs, et cetera. But let's go one by one. I think it's easier. So about zooming in, what do you think?

**Interviewee:**

I think that the product that we started out with was a very broad product. So definitely zooming in is something that we've done. I'm not sure that technical debt really has a role in that.

**Interviewer:**

It doesn't have to. How about zooming out? It is a bit like the opposite, as the whole product.

**Interviewee:**

No, I don't think that we've zoomed out very much.

**Interviewer:**

How about the customer segment? What do you think?

**Interviewee:**

Yes, we sort of shifted in the beginning, the customer that we focused on was the actual fish farmers in the market, but in the zoom in, we saw that the product that we had that was pretty broad actually was initially adopted by the guys operating the boats. So we shifted our focus away from the actual fish farmers and towards the ship owners and the guys operating the boats. And in some sense, also the guys producing fish feed, which then again, they had boats or they charted boats to freight the fish feed. But I'm not sure if we said that we would, we haven't shifted the customer segment because we also have the fish farmers as customers. So we sort of, we maybe broaden it, but it's still within the same segment.

**Interviewer:**

Yeah. You just expanded a bit from the previous original one. What do you have to say technical debt?

**Interviewee:**

No, I don't think technical debt had any, it's more business-wise.

**Interviewer:**

Yeah. Customer need again you realize the problem you're trying to solve et cetera, not very important for the customer.

**Interviewee:**

Yeah. So that I think is also fundamental basis of the same Sumin that we sort of had a broad product and, what we thought was the problem was not really the problem that customers wanted to pay for. So we had to refocus and sharpen our focus to actually find the problem they were willing to pay for. Yeah. But again, not the technical debt-related issue.

**Interviewer:**

How about the platform? You said you have software as a service today. Was it like that at the beginning?

**Interviewee:**

Yeah. I've been doing this, software as a service in many companies and, and built several startups doing this, and that's what I do, but when it comes to platform pivots, I would say we've at least had one, but it's more maybe a technology pivot, when it comes to well, at least this mobile app, the whole framework there has shifted or changed.

**Interviewer:**

Yeah. But since it is technological a little bit do you connect to technical debt somehow? Like since you shifted from one

**Interviewee:**

Yes, because when it comes to the mobile app we used to develop that in what's it called it a phone gap. It's based on Apache Cordova, but we used that to avoid having to build specifically an iOS and an Android app. Yep. And the Apache Cordova platform hasn't really gone anywhere the last few years. So we had problems supporting the new versions of Android. So we basically changed, and now we use two different frameworks for mobile apps. So that was basically a technical debt that we sort of,

**Interviewer:**

You use a different framework for developing the app or cross-compiling it? So now when you use a new platform for cost compilation, I guess, or is it from like ionic, whatever you mentioned?

**Interviewee:**

Well, the thing is, a lot of the base code is still the same because it's mainly written in, HTML CSS, and the OS script. Yeah. But when you change the cross-compiling you basically have to rewrite the whole orchestra of how to build the app and how it's gonna behave.

**Interviewer:**

Because in the Cordova, you made sure that the menus and stuff worked in Android this way. So you always have some specific files for each platform you wanna cross-compile to manage some of the differences on the platforms. Yeah. The whole orchestration of the compiling had to be redone.

So it's more cross-compilation plus some developments. Business, business architecture, like your startup has switched a little bit from low volume to high margin clients.

**Interviewee:**

So in a startup, you will always have the need to plan the growth. So at the beginning with few customers, we didn't really have a big stack of tech to support the solution, but we've always been relying on big data when it comes to vessel movements and geographical data. So that's been massive from the beginning, but we've, we've migrated from smaller self-control servers and into more reliable cloud services through Google cloud at the moment. Yeah.

**Interviewer:**

That's pretty good. And this means that you have a bit more, more clients now because you are growing and they don't have to pay a lot. Maybe they have fees peruse.

**Interviewee:**

Some of the solutions that we make are actually quite expensive.

**Interviewer:**

Maybe it peruses. It's not like they have to, or per subscription?

**Interviewee:**

It's a subscription model

**Interviewer:**

Value capture, like and again if I repeat it here, but I don't think it's connected, like technical debt. Did it play any role in this?

**Interviewee:**

No architecture.

**Interviewer:**

So value capture, your startup has changed the way method you capture value monetize?

**Interviewee:**

No. I don't think so.

**Interviewer:**

But that's okay. As long as it works and you monetize well, you don't have to change

**Interviewee:**

We'll see. You never know.

**Interviewer:**

You never know. Engine growth has made significant changes in its growth strategy to keep rapid and more profitable growth. You mentioned growth a little bit.

**Interviewee:**

For Startup 7 it's always been hard because it's we are in an industry that is slow adopters of technology. So a typical sales cycle in our business could go from between two to three years. Of course, it would have been nice to have capitalized on the company better so that we had more money to spend on more fast and more rapid development. Yeah. But again, we've sort of ended up since the cycles of sales take so long time, we've sort of built a bit more by brick kind of company. So we haven't really made any significant changes in the growth strategy.

**Interviewer:**

And the channel pivoting do you think, like your startup has identified a more effective way to reach the customers than its previous ones? Like, do you have better?

**Interviewee:**

No, I wish there was possible to have a more effective way to reach our customers. But maybe let's say that during the Corona I think that this industry even has provided an option for us to meet the customers face to face easier than before. Yeah. Because it's a business that is very facing to face. It's very personal. Usually, you have to go to Aquanor and all these big network gatherings to actually have a sit-down and talk with our customers. But now it's very easy when it comes to teams, zoom, Google meets. So maybe that has been a more efficient way to reach our customers, but I wouldn't say that it's a technical debt thing. It's more like a Corona thing.

**Interviewer:**

Yeah. Does it have to be no, the final one is the technology pivot and I think you have had some technology pivoting so far.

**Interviewee:**

Yes. We've done a couple of technology pivots, so I think that we could say that we are on the third iteration of different technology at the moment.

**Interviewer:**

Is it connected to technical debt a bit?

**Interviewee:**

Yes. You could say it is because the technology that was chosen in the beginning, wasn't suitable for building the software that we wanted to build. So that basically means that during the past six years, what we started out with the first version of an actual working software we had was back in 2014. But that platform couldn't really do what we do today. So we basically had to redo the whole platform.

**Interviewer:**

Yeah. And how about the coding part? You have changed some programmers, I imagine.

**Interviewee:**

Yeah. So we've gotten new programmers since then. And that was one of the reasons why we basically got new programmers as well.

**Interviewer:**

Yeah, I understand. But then you had to throw away a lot of code that you had already developed.

**Interviewee:**

Yes.

**Interviewer:**

And a lot of effort from before. So I understand. So do you think it would've changed something if you restarted today and do the things that you have done since 2012? What would you have done better? This is my question to avoid a lot of technology switching.

**Interviewee:**

So I came on board in late 2014 and of course, if I had been a part of the company from the beginning I would've done exactly what I do now, but it wasn't me calling the shots the first couple of years. So then they made some bad decisions, but that was based on their lack of insight into the technology. So they listened to external consultants choosing a bad platform for scaling.

**Interviewer:**

Totally understand. Yeah. By the way, do you use Jungle perhaps with Python?

**Interviewee:**

We don't use jungle.

**Interviewer:**

Why not?

**Interviewee:**

We have relatively small backends written in Python. All these Python backends are basically small Docker containers that we all constraint in, in Kubernetes, and they do very specific things.

**Interviewer:**

It's mainly scripted then.

**Interviewee:**

Well, some of them are programs, but the whole platform is basically JavaScript-operated.

**Interviewer:**

The reason I ask is that I've worked on these technologies before. When you talk about this Google cloud, mobile apps, hybrid applications, I fully understand what you're trying to say, but I was just curious because I've been using jungle a lot and it's solved a lot of issues for me, like viewing, et cetera. So that's why it was just a curiosity, but the final, final question, how would you explain the role of technical debt in your startup pivoting scenarios in one or two sentences? Like I understood that you had technical debt and pivoting connected when it comes to technology. Like platform pivoting and technology pivot. So how do you state in a couple of sentences, what is the role of technical debt in startup pivoting?

**Interviewee:**

I would say that Startup 7, we are aware of technical debt and we manage it. We don't avoid it. We actually actively choose to use it sometimes because the means of delivery on time is maybe more important than avoiding technical debt.

**Interviewer:**

Imagine that whenever you try to manage, manage, manage at some point, maybe you end up with a pivoting situation anyways, in the future because of the technology and the technical debt you accumulate. Because even though if you manage you accumulate, so yeah. Yeah. So do you think it influences the pivoting at the end of the day?

**Interviewee:**

I think that it will eventually. So the thing about technology is that it, I mean, we could all have been still written stuff in Cobalt and been quite happy about it, but that's the problem with technology. So technology is almost like a fashion sometimes because some guy comes along and he has a bright idea to build a much better programing language and blah, blah, blah. And it's the same still story all over again. And in the end, it's all about choosing tools and a platform that has enough support in the community that so many people are using because then there will be libraries available, then there will be an option to use it for a long time. So what's it called? If you're familiar with Jquery, it's typically, it has been jQuery and J query mobile super popular framework for a long, long, long time. Still works, but it's sort of the end of life. Because now you have to move on, it reacts or its view or something else. I think that this will be the future as well. I think that in the future, there will be more cool stuff and there will be a new version of whatever. And in the end, you will have to pivot into some new technology. I also like to try new stuff. That's why we have so many different technology stacks that we work on because I find that if you want to handle big databases with fast lookups currently the most performing system we have is written in pure Java. And it uses a very good database at the bottom. But maybe in the future, there will be some other technology stack that we can utilize that even performs better. Yeah.

**Interviewer:**

Oh, I see. Yeah. But I think this has been informative for me at least and I hope it has been also for you good conversation. Yeah. Do you have anything to add because we are at the end of the time?

**Interviewee:**

I don't think so. No. I think the startups, in general, I'm a CEO of Startup 7, but me and two colleagues, I basically work on a daily basis I think we have 14 or 15 startups because we invest in startups. And then in some startups like Startup 7, we invested, but we discovered pretty soon that the guys, the founders of Startup 7, didn't have technology in their background and they didn't have management in their background. So at some point in time, when we had invested enough money, we sort of came, moved in, and took over both the technology and the management staff. So today I would call us co-founders of Startup 7, but most of the startups that we have, all struggle with choosing the technology platform. Very often it's because a startup doesn't really take it seriously enough the CTO role. They need someone who has been out swimming there before and knows how to choose something that they can live with for a while and to manage that technical debt and the reason involved.

**Interviewer:**

I agree. And sometimes it's very good when you have this CTO role matched with the CEO, at least at the very beginning, because it gives a better perspective. Different perspectives for the company to grow and to make the right decisions. Yeah. So they don't have to throw away code or change technology every few years because they find out things, not just change because of fashion, but also because of need.

**Interviewee:**

I see companies that sit here alongside us in digs. One of the companies had the first version was built by some young students who wanted to work while they were studying. They had to throw away everything after one year. And then they got an outsourcing team in, Ukraine, and they developed their next version, but they still hadn't any CTO or any technical, person in their company. So when they hired the initial, tech guy, he almost threw away everything that was built in Ukraine, and now they're on a third leg of the same product basically. I feel sorry for them. But just to finish off, I can say that, I very often meet developers who have, they suffer from a syndrome that is not so very nice and that's the not invented here syndrome if you heard of it?

**Interviewer:**

Not invented here. Okay.

**Interviewee:**

So when you hire a new tech guy, new developer, and he comes in, and he looks at your code. It's almost like a hundred percent match if there's a new developer coming in, he will, instead of trying to manage and further develop the things that have been built before, they are suddenly struck with, I'm not the inventor of this, so I have to rewrite it—the not invented here syndrome.

**Interviewer:**

I have to redo everything because this doesn't look nice. Yeah. Right. I understand. Yeah. That means that a lot of people are not building usable code. They don't understand the generic usability functions when they write code.

**Interviewee:**

I think that very often, I mean, I think that you really need good arguments if you have to redo everything because fixing things that don't work is usually less costly than remaking it.

**Interviewer:**

Thank you so much. As I said, this is very useful for me. And please send me back when you have time the paper to sign from NSD; I will. Yeah. And have a good Easter holiday and all the best and good luck with all the companies.