**Interviewer:**

When it comes to the company to the startup what is, can you tell me, what is the core product and service?

**Interviewee:**

We have three. One of them is not software it's basically consulting in security services. Then we have a security awareness portal for giving security awareness training to employees that is role-based. Okay. And we have a privacy management software called privacy box for managing access requests under the GDPR and the CCPA legislations. So those are the three services we have.

**Interviewer:**

Good. When were you established as a startup?

**Interviewee:**

2017 is the formal establishment of the company. The first prototyping project started in 2016.

**Interviewer:**

You are located in Ranheim or in other places as well?

**Interviewee:**

Well? Yes. Our company address is in Ranheim. Our founders are [inaudible] and [inaudible]

**Interviewer:**

And your role is in this case? What do you do?

**Interviewee:**

My role is to be the CEO really, but what I do is I guess we do a bit of everything. It's been a side project for all of us, so we have other jobs. There are four people owning the company. We run this on the side. So as we do a bit of everything, all of us, I guess.

**Interviewer:**

Do you have such an office or is it like you do work remotely?

**Interviewee:**

Currently, we only work remotely for the first six months we had an office, but then we decided we don't really need it. Okay. So yeah, it's only remote.

**Interviewer:**

Yeah, but you are not in a co-working space or something like that.

**Interviewee:**

No, nothing like that.

**Interviewer:**

How many employees, you said something about that?

**Interviewee:**

I guess the only person who is hired by the company is me. So it's in a 50% position. So this is half an employee, but we are four people. Four of founders.

**Interviewer:**

That has hands-on experience, I guess, on the product that you are developing.

**Interviewee:**

Yes.

**Interviewer:**

Including yourself. I mean, it's yeah. Well, what I wanted to know is that you have some background in software engineering or in computer science.

**Interviewee:**

So two of the founders, we have a tech background, we have done quite a bit of the software engineering ourselves, but we have also used freelancers for that. It's not only things we have developed ourselves.

**Interviewer:**

How many freelancing did you use?

**Interviewee:**

For software development, I guess it's only one person. But we have had several on design and on the video editing and things like this

**Interviewer:**

In terms of customer base, do you feel the company is growing? Do you have more customers?

**Interviewee:**

It is slowly growing, but to be fair, we have found that it is easier to generate growth within consulting than it is within our software services. Although what we actually want is the other way around. So we get more requests for sort of security consulting than we do for our software offerings. They also have different groups of people we have been targeted with so for our sort of awareness training, typically the target groups are smaller companies whereas the privacy box offering is more of an enterprise offering. So that is more by accident and planning to be honest, and, and it has cost a bit of challenge when it comes to, to marketing and so on. But we are slowly growing.

**Interviewer:**

All the services they generate revenue, right?

**Interviewee:**

They do generate revenue. Yes,

**Interviewer:**

That's good. If I ask a little bit more questions regarding software engineering practices and tools that you are using. You said something, but maybe a bit more in detail, what kind of software engineering practices do you feel you are using? What kind of tools? Programming languages?

**Interviewee:**

Yeah. If you talk about the organizing things we are [inaudible] based. If you can take the product that one product is easier to focus on and can look at the privacy box thing, it's quite feature-driven. So it's a lot of the sort of user stories and so on, they actually come from customers. So we have a baseline product that's quite useful to them, but sometimes they want like, can you fix this? Can you do this? And this is one of the things we have tried to be different than some of our competitors is in actually being receptive to that and, and implementing things that customers ask for. That has been quite successful and may get very good feedback of being receptive to that. But we do have this [inaudible] based approach, some of the requirements come from our own sort of, okay, we'd like to take the product in this direction. And some of it comes from customers asking, could we get this feature? Then we put that into a backlog basically, and we don't actually do continuous development all the time. We do some bursts when we have like, now we have enough work to sort of progress it a little bit. And then sometimes we've done that internally. And sometimes we have particularly one external freelancer that we have worked with that we have been very happy with. There was a Norwegian consultant, it's a senior person who has 15 years experience in software architecture and on, so it's been very easy to work with. But unfortunately, that person now has a full-time job. So we have to find someone new but yeah, so that's the last time we've done it. First, we do development, we write unit tests for new features. Then we take it into customer testing. Especially if it's a, customer's ask for a future, that customer is basically the tester. They get to try it out before it goes into production. See, is this how you want it? How can you do this bit of change? And then we push it to production after we do integration testing. So yeah, it's nothing out of the ordinary, I guess.

**Interviewer:**

No, but it shouldn't be why, why should it be out the ordinary.

**Interviewee:**

So if you look at tools. This service, this is really so to think about tools, the most important testing aspect here for us is the unique tests we write ourselves. We use OVA's dependency checker okay. To keep track of dependencies. And for this particular product that talked about is an old JS application. So we use the NBM audit as well. We find that they are pretty much in agreement nowadays. I guess that's it. Do you want to know a technology stack and so on as well? Or is this enough?

**Interviewer:**

Yeah. We can discuss that.

**Interviewer:**

In terms of the technology stack, what kind of technologies are you using to develop the software?

**Interviewee:**

In terms of the technology stack, we can start at the end of that pipeline, we host our services on Google cloud and because we don't have a lot of sort of infrastructure resources or anything like that, we use the platform as a service basically. So it's a lot of Google app engines. It works very well. Its scales very nicely require very little management. So going to the actual stack that we are controlling. So for this one product we talked about, it's a no JS express JS backend with a view from that. We found that that works quite well, that are developed separately, so we can add different people, almost different tech stacks. For databases, we use Mongo DB in the cloud. It's hosted on the Mongo app class also in Google cloud to save on services, different clusters for dev and production obviously. Cold quality.

**Interviewer:**

The next question is what are important features for quality like UX performance security, your usability?

**Interviewee:**

Yeah. So security is obviously important for both of our products. So we do a lot of test things in that respect, reliability is important to us. So latency is not so much of an issue because this isn't software where you have an end-user doing a lot of things. It’s, especially not a privacy box solution, that solution is, exposing a web form basically on the webpage of the customer where, where their end users can send in their privacy requests. And it's a web application for the privacy team to collaborate on responding to it and doing, audit logs and all of that stuff. So performance is, and UX is important to some degree, but by far the most important thing here is reliability.

**Interviewer:**

And you said some things about testing.

**Interviewee:**

About testing. We try to write negative unit tests for everything that's security-critical. Being a small team, unfortunately, we can't necessarily, we can't always separate between who is writing the actual software and who is writing the test. That is an issue. In addition to these tests that we write, we do run security scans of software using a [inaudible] that's not built in the pipeline. It's more of a manual thing that we do whenever we have changed something major. We use the dependency bot of GitHub. These are privately hosted repositories in GitHub. Then we have a set of dependency bots to generate pool requests, when there are new basically found that things are updated. We found that that works very well. On some occasions, it will break things we don't automatically deploy or anything. We run testing, but that's not automated. We manually test the poll request, does everything work and then we merge it to production if everything is okay, if not, we have to figure out what's the issue. On the backend, we have never seen any problems with it. On the front end unfortunately we have seen some problems with it. I wouldn't do the same technology stack on this front end again if I were going to rebuild it, because we end up with quite a large dependency tree, which of course causes some compatibility issues. So that's it on that side. And also in the cloud environment, we have set up a Google cloud security scanning service for it. So they have a built-in security scan that does typically over top 10 things. I am not sure how good it is because it's never found anything, but I have found things myself that I would've expected it to find. So we do have it, but I don't trust data.

**Interviewer:**

I guess you're a bit skeptical. I Understand. But, how about the documentation of your coding or practices?

**Interviewee:**

Yeah, documentation of the code is mostly integrated into the code itself with comments. Okay. We have end-user documentation. Doesn't focus on the code, of course, integration documentation for people wanting to integrate the software into their webpage, and so on. But that's very lean. When we talk about the privacy box product, it's mostly an eye frame.

**Interviewer:**

I understand. But still when you develop things, so when you test things you document, perhaps I know.

**Interviewee:**

Yeah. And we have documentation of the features that we run. We were using Jira for a while, but we found it was too heavy. So now we mostly use KANBAN boards in Trello. When we run scans and so on security scans, we will attach that travel issues and we have traceability of it. It's not very formal.

**Interviewer:**

No. It's more flexible now. I would like to shift to another part of this discussion, which is about the technical dept and pivoting. I mean, you may already know both terms, but it doesn't hurt if I, gave a short briefing. Like technical dept, it's mostly about, when you have developed and when you are developing features, sometimes you take shortcuts because you want to, you know, fulfill the deadline or something like that. Yeah. But you don't adopt the best practices, software practices, and this leads to a sort of debt that you might have to repay later by refactoring your code or improving. Pivoting, it's more like changing direction with your company, your startup. So, before I jump into pivoting and what kind of research classifications we make for pivoting, I would like to ask a few questions about technical debt. So, first of all, how much aware are you about technical debt within, your startup?

**Interviewee:**

Yeah, we do have a significant amount of it, as I said, it is like a lot of startups. We have a lot of sort of prototyping that turned into production software that tends to generate technical debt and it has cost us to spend some efforts on refactoring. In fact, that was the first use of an external consultant was to hire somebody to clean up code and refactor things that we knew were problematic. So we have not quantified it. We don't have a dashboard for technical debt, estimates how many hours that we take to fix, or anything like that. What I think is the most obvious issue that we have a technical debt is called duplication and collecting things into good reusable components, which sort of reimplementing the same sort of features across various places, which means that maintainability is hurt by it. It's not a huge problem as such, but when we have to fix it, it means that we will have to refactor things and it takes more time than it would have taken if we have better code quality.

**Interviewer:**

What's your perception of technical debt in general?

**Interviewee:**

It has a very high-interest rate, is my perception of it. This is very close to that, I work on application security in a more enterprise setting in my day job. So I would really like to avoid it, but at the same when you're starting out and you don't really know what direction you are going, you try a lot of different things. And then often the implementation comes before design and that generates technical debt. And I think that's quite common in startups. I've talked to a lot of other people who have started small companies and they sort of see the same problems with code. The issue afterward is it's often hard to find time to do the refactoring you would like to do. I'd say technology is the issue and when it comes to this idea of a lean startup,I think this is partly driven by a lack of funds in the startup when it's really a bootstrap startup, you really want to push things to market to get cash on the time you spend on software development. So the long-term return on that investment in software development is diminished by this, but even if you know it, it's hard to avoid it in the start. So I think it would be very good if you could have a more structured software engineering approach where you have better code quality, lower complexity, and so from the beginning. But, think we have the same issues as a lot of other people. That's a lot of like trial and error at the beginning and that sort of leaves on in there because you never find time to refactor those things later.

**Interviewer:**

Yeah. Interestingly enough I've been running a lot of interviews about this technical debt in US, here in Norway. And I sort of found out four dimensions, like some people at an early stage, some startups actually were trying to ignore or even accept technical debt. Ignore, that was more because they knew what it was more or less, but they said, okay, I don't care about it. When they sort of say, oh, we have to implement, as you said, but we have to get the job done, but we accept that there is technical debt and this also happened at a later stage also, but when things grow and become more mature then what happens is that startups want to avoid, you mentioned, I think to avoid at some level technical debt if it's possible and even also manage with some good practices, how to manage technical debt. Where do you see yourself now with technical debt, with respect to before? Like, do you want to avoid it, manage it, accept it.

**Interviewee:**

I just manage it, is the correct place to put it. We would like to avoid new technical debt touching things that in particular, if you talk about this type of code duplication, what you typically end up doing, if you see you need that sort of feature again, then we start refactoring putting it into its own component, its own class, something like that. And then we can reference it and so on. As I said, it's like, yeah, it's a very lean startup. We don't have full-time software developers or anything. So we are not doing this sort of continuous focus on code improvements. But we would've liked to if you had the funds to do it.

**Interviewer:**

Compared to the early, very early times when you started the company in 2016 and now you are in 2021. How did you cope then and how do you want to cope now?

**Interviewee:**

Well in the beginning you sort of ignored it because we were throwing things together to get like demos, the minimum viable products to show the people, and so on. And then sort of ignored it at that stage. We knew that we were not following good practice, but we were really focused on getting something that we could show the world. And so that has changed into, okay, now obviously that's not the focus because we do have products. So now it's more okay. When we do improvements of those products, we don't want to make more problems for ourselves in the future. So we try to follow better practices.

**Interviewer:**

When it comes to, if we shift this a little bit with trying to connect technical debt with pivoting, so first of all, did you pivot like the startup focus, transitioning primarily to the growth phase? Has any of the situations that I've listed a little bit here, like 10 situations, but we can either take them one by one. First of all, you have to tell me if you have pivoted a little bit within the startup.

**Interviewee:**

We sort of had, but I guess it's more. What we did was we brought on a new founder after a couple of years, he was very privacy-focused. He's a privacy lawyer the ID here was to help manage GDPR issues. So we sort of built a separate product for that. So that was sort of a halfway pivot and I'm not sure I would have done the same decision over again because we saw afterward it was hard to. That ended up in different customer segments and that was not the intention of it. So you could stretch the resources without this being very planned. So it was a pivot, but not very successful. I think it has sort of harmed our ability to focus on one uniform platform. And another thing adding to that is that we have a different tech stack for this product. This means that the same typically hiring freelancers would typically mean that the same person isn't necessarily good at working for both of them. So the one I talked about is no JS JavaScript. The other one is Python-based, so yeah. It's quite different. We have pivoted, but I wouldn't actually say it was a well-planned one. It's not been a disaster, but it's also not been easy. So it's closer to your sum out, but those are not really well-integrated products. It's more like two different products.

**Interviewer:**

Yeah. And the customer segment, you said something about this, I guess.

**Interviewee:**

This is kind of interesting because we have been a bit all over the place with this because the company started out though. We were three people working together in a large consulting company. We did the information security threat assessment basically for large companies, large multinational companies. And of course, buying this from a big consulting firm is very expensive. And what we saw as a situation is much different today, actually five years later, it's a bit sad. But what we saw is that a lot of the practices that came out of doing the threat assessments for that were quite formal and for the same all the time. And we wanted to make some of that process and then offer it to smaller companies that can't afford to pay 2 million to get this done. So we started on that journey and then we tried to sort of getting into the sort of SMB segment companies with less than 250 employees and so on and in 2017 nobody was interested in that. I didn't get a single yes, this is something we want is like what are you talking about? So then we went to an insurance company, talked to them and they were struggling to sell in the same market. They wanted to bring cybersecurity insurance through the same companies that we were trying to sell this threat assessment software. And they were struggling with a lot of the same things and we tried to, okay, maybe if we partner up here and sort of package it up, so if they do this it's structured threat assessments, they will get cheaper insurance. And we had a lot of good interactions with the Insurance company on nobody was interested even if you packaged it up nicely. So that didn't work very well. We need some cash flow. So we still wanted to hit these smaller companies. And then we decided like, if we need some cash flow, maybe we should start selling some consulting services because we knew how to do that. And then, okay. Yeah, that works. That's not so hard. Okay. We get some cash flow we can sell like you go out and do threat assessments and we can go out and talk about DevOps and, and yeah. Like the things that, that people want to some help with. And then they found out that some of these bigger customers that typically buy these consulting services were also interested in software. But not as users of the software, but they wanted to use it, so we went into working with that and then we ended up working a lot with its companies and so on, but still like the software bed is 80, 90% of it. So it's definitely pivot, but it's more driven because we need money to come into the company. The other pivot was that privacy product coming out of GDPR people saying, okay, we have, have to do this GDPR compliance thing. Can you help us do it? And then we built something to do that, which more like market-driven, much easier of course because then you have a customer coming and saying, we have this problem. Can you help us solve it? And then we, okay. We build this software. Okay. Yeah. Thank you. This helped us solve that problem. So that's a much easier product to sell, but now we're back on the security side trying to reach those SMB companies. And it's part of that webinar on Friday is part of that thing, because we're now targetting like smaller software houses that don't have this in-house competence in security engineering.

**Interviewer:**

Sounds interesting. But before I make this question on connecting the technical debt, how do you want to approach it? Because first I have some more pivoting scenarios, but I would like you to make an effort maybe at the end, this is not what I did, but we can try to make an effort at the end and see if the pivoting was somehow connected to technical debt.

**Interviewee:**

I'd say it was not, it was entirely market-driven.

**Interviewer:**

The first part. How about customer need, you said something about pivoting for customer need perhaps

**Interviewee:**

I think the situation isn't really a need, but at the time it wasn't really recognized that smaller companies needed to do any sort of structured work in cyber security that has changed a little bit. I think the need was latently there back then as well. But then in 2016, the thinking in smaller companies around cyber security was if I buy a firewall and office 365, I don't need to do anything else. Yeah. That has changed a little bit still selling this to smaller companies is difficult because until it is needed, it's seen knows like something that's just going to cost. So that's part of that problem. But the customer knew that that was very clear was when we had people coming to us and saying, you know, about the security part, cause you're not so worried about it, but we are very worried about GDPR Compliance's definitely one of those situations. What they worried about was primarily spam, to be honest, because they have this email address, they put on their webpages like privacy@whatevercompany.com and they get loads of spam to that mailbox. Yeah. They try to use a spam filter and so on, but, but it's difficult. And then they have a bunch of lawyers using this. They're not technical people. They have to reply to this and they have to track that they're doing it and so on. So we built a simple portal for them where basically people fill out a web form, they have to like to click a confirmation link that they're emailed to check that the email address they're supplying is valid and so on. And that reduces the amount of noise and process for these lawyers quite a lot. So that was sort of the need to serve everything behind there is relatively simple. And in sort of tracking things, creating logs that people have sent messages and so on. But the primary need that came up was okay, GDPR is coming. It creates a lot of these reports for our lawyers where they used to before GDPR, they get like two to five emails per day in the mailbox like this. And suddenly they have 500 emails per day. So then obviously it didn't work sorting through all that manual. So yeah a customer need is, is definitely one thing for that product pivot.

**Interviewer:**

Yeah. We'll just go through the potential pivots and then ask the question about technical debt in the end, but the platform pivot, like your application, has turned into a platform perhaps, or vice versa. These are not listed for me, this is research-wise, this is what research has identified as pivots in startups. So it is interesting to see through those lenses, actually. This architecture, like your startup, has switched the business architecture from low volume and high margin, for instance,

**Interviewee:**

This is true because in the beginning, we want to reach a mass market with low customer interaction. And so on, that's the stuff we saw didn't work. So, we changed to like a lower volume, higher margin type business.

**Interviewer:**

And then value capture, like you have changed the way to capture the value and monetize. I think you mentioned something about this somehow.

**Interviewee:**

The primary cash flow has been from consulting services, but that's basically been the situation most of the time. So we haven't really changed much in that, but we want to change it cause you want to stay away from that model and get more into recurring revenue from software. We have some of that, but the majority of our revenue is from consultancy.

**Interviewer:**

Yeah. Out of curiosity, what does this consultancy consist of?

**Interviewee:**

It's primarily application security-related.

**Interviewer:**

So you have to develop an application for somebody else?

**Interviewee:**

Oh no. We go in and help their teams do threat modeling. Come up with mitigation plans also on organizing security planning, penetration tests configuring their cloud environments, stuff like this. We don't configure it for them. We ask 'em questions. Have you thought about this? And then, oh, we didn't think about that. What could happen then? So a sort of risk assessment type services.

**Interviewer:**

Just an idea, my impression is that this can also lead to training services

**Interviewee:**

We have a bit of that too. So we've, for example, we've done insider threat training for managers, for large government organizations, we do things like this. We have software supporting these services, but that software is only used by us. For example, we have an automated threat intelligence collection chain that is cloud-based, but we don't really offer it to anyone else. But we could, but we haven't like tried to build marketing around it or anything since we used this for like building threat landscape descriptions, customers, and things like this. Yeah. It's part of consulting.

**Interviewer:**

That's good. Engine growth, like you, has made significant changes in its linear strategy, how to seek rapid and more profitable growth.

**Interviewee:**

We haven't really changed so much there. Part of the capacity, I guess, as well as I said, we are doing this sort of as a side business to other jobs, we work in very different companies for our day jobs, one of our founders is a psychologist being at the department at the hospital.

**Interviewer:**

Channel pivoting like your startup identified a more effective way to reach the customers.

**Interviewee:**

Yeah. I guess they've sort of touched upon this. It's relationship-based sales and it's more enterprisey I guess, so that hasn't changed much. That's what we have seen work. We have tried a lot of different marketing strategies and so on. Nothing has really stuck. There's one thing that we really wanted to make work and that was this collaboration with that insurance company. We really tried and they have quite a large customer base, but unfortunately, it didn't really work out and it didn't work out for them either because their product didn't sell very well. So there are several reasons for this. This is a very traditional company and you're trying to be like an elite startup. They have very long sales cycles and that collaboration was difficult for all of the reasons that collaboration between small and big companies tends to be difficult. But also I think we were not approaching the market in the right way there. But a channel pivot is something that we would like to see because we would like to reach this SMB market still. That's sort of the vision of the company, that's what you want to make happen. But we haven't succeeded with that yet. So let's see if that happens in the future.

**Interviewer:**

Finally, a technology pivot like your startup has delivered the same solution by using different techniques. You've mentioned something about this, I guess.

**Interviewee:**

We have different technology platforms. We've not delivered the same type of service using a completely different technology that hasn't happened, but we have migrated some of the sorts of more traditional backend things developed in Python to using serverless and it's not very market-driven. It was because we wanted things to scale better and we wanted to save some money so that's more the reason for it. But that has worked very well, to be honest, we thought it would be more work-intensive to do that, but that was actually not the case. So it's using cloud functions with Google which is the same as AWS lamb, the functions a lot, we using a bit of fire-based, so services, which also has worked out quite nicely. We have had some issues with our cloud platforms like bugs and things like this, but mostly it's been very reliable and very well documented. So we're happy with that, but, but it's not a big pivot or anything. It's more like, okay, we've taken some of the things that background jobs and things like that and taking it out of a Python environment, putting it into a more cloud-native environment, just because it's easier to manage that way.

**Interviewer:**

But just to sort of try to make an understanding here with a technical dept I understand that most of the pivots maybe were not related to coding issues or to development issues with the code.

**Interviewee:**

I guess none of them, to be honest if we didn't change anything because of problems related to our code.

**Interviewer:**

Yeah. Because in some cases it happens that companies developed very poor code due to technical debt and they decide, oh, we have to pivot now redo choosing different platform or choose different technologies. And then maybe things are better and we can avoid technical debt this way or something like that. But in the future let's position ourselves in the future. What is your opinion here? You have been spending around four years on this startup, I guess. Next five, when you see all this list of potential pivoting and technical debt awareness, I saw you had a very strong background there. So how would you see it connected to technical debt? Because maybe it might end up in that scenario.

**Interviewee:**

Yes. In terms of technical debt, I guess a technology pivot to make it easier to track good freelancers with the way we have chosen to build our software could actually be a relevant point and to help build smaller because it is sort of a monolith application, especially on this security awareness side. I think moving towards a more micro-service-based architecture. Basically not because it's needed for the customers or anything like that, but because it's easier for us than to outsource parts of it and at the same time handle encapsulation of code issues within that single service. So I think that is something we have discussed. We haven't really done anything about it, but it's probably going to happen. And that's also part of what I talked about. Right. We've taken some of the services out that we need scheduled background jobs to create server-less architectures for doing it. It's easier gets like, and then we can make that work and we can remove it from the old cold base after that. I guess we will continue moving in that direction. It's not primarily related to technical debt, but it makes it easier to avoid it or to manage it.

Interviewer:

Yeah. That's a good answer. The final question is more like how would you explain the role of technical debt in your startup pivoting scenarios in one or two sentences? But I don't think this is the case. I think this is what you said to me.

**Interviewee:**

It's not really related. It's not what's happened in the past.

**Interviewer:**

That's okay. Not everybody has to be involved in technical debt and pivoting together. Otherwise, my research will be very ambiguous.

**Interviewee:**

I think one of the issues you've probably seen a lot of companies is the lack of visibility of technical dept, including ours. We don't have metrics for it. And we haven't focused on it, but as I work with this in a more enterprisey world from day today. If I didn't have metrics measuring technical debt, it would be very hard to measure it. But to be honest, I've never thought about that in the startup context. Yeah. So I don't know if you do you know if you have relatively small startups that do actively measure technical debt?

**Interviewer:**

No. I think most startups, say, oh, we are aware of it. We do also startups that have grown actually pretty well. They say, oh, we accept it sometimes because we understand we have to get the job done. But then afterward we also want to track, these are the best scenarios. Like you said a little bit, we want to check and fix it later. It's not like, how do we forget about technical debt? Unfortunately, the tools are very limited to do the tracking. This is also a research-wise context there are few frameworks or tools that you can use.

**Interviewee:**

What I have seen has given the most practical benefit is to measure cyclical, complexity, co-duplication, and static analysis. It's very rudimentary, but it's more helpful for seeing if you're driving things in the right direction, at least.

**Interviewer:**

Yeah. But a good idea might be that many of the companies whenever they make the software engineers, because this is not just a problem of sort of, to say of the people that lead the company, it's also of the problem of the developers because you hire out freelancers, they have their own competence. The world is not ideal. And when they developed a software solution in a certain way, it's also because they knew how to do it in that way. It's not just because they were trying to do the best solution possible. So there are some limitations here also from a human side.

**Interviewee:**

Definitely.

**Interviewer:**

And I've created many projects before with these Google cloud platforms, mobile, et cetera. Whenever I worked with developers I understood the fact that there are limitations with people. So sometimes you have to tell the freelancer or the guy that's working for you. Oh. But you find the best solution you can. So then it happens that you get the technical dept in an unfortunate way. So you have to deal with it.

**Interviewee:**

Yeah. I had a project a couple of years ago we were assisting another startup actually. And they had made a mobile application by outsourcing their application, which has to do with these things, like a list of features, the company in Ukraine. And they had some collaboration problems and we were sort of hired to mediate and make people agree with each other again and so on. And then we did a cold review together, which was sort of the source blood discussion. And I definitely had that because that company had basically built their own thing without any sort of good architecting principles or anything, and amassing a lot of issues including security issues. So I think when managing technical debt is also part of managing your outsourced partners and following up things in a more hands-on way during development

**Interviewer:**

[Inaudible] to go at the end of the project not at the very end, but at the end of the development phase, before you start testing and talk to the developers and tell them, okay, can we do a cold full code review, see how things have been developed, what you have thought about the architecture and so on.

**Interviewee:**

Yeah. Architecting principles at the outset are, is a very good idea. We don't really have that at Startup 8 but I think maybe we should have it

**Interviewer:**

Sitting down and talking to people. I think it's very useful in the end. The development phase before you start testing and whenever you see issues not so good practices yeah, maybe bad practices. Then what you can do is start thinking of refactoring things or improving the actual architecture. Transforming if you reuse Python, transforming functions into classes, et cetera. So make it more object-oriented and stuff like that. But yeah. Do you have something else to add because I'm mainly finished here with my part? Did you find it a surprising interview? Did you expect something? I always ask this question

**Interviewee:**

I didn't really think about it, but I thought like, yeah, I was happy to contribute to research. I wasn't really surprised, but I dunno what I expected to be honest, but it was okay. Was a nice chat. I think it's it sounds like interesting research. So

**Interviewer:**

Yeah, it's actually the first time we do try to map technical dept that with people starting startups, not so many, actually no papers about this before. Yeah.

**Interviewee:**

Yeah. I had a very interesting discussion about your topic with [inaudible] at secure practice, like a year or two ago. Probably a good person to talk to. I know. He has thought about these issues before.

**Interviewer:**

Do you want to recommend him so I can give him some email or something?

**Interviewee:**

You can write him an email. I'm sure you can find his email address his company is called secure practice

**Interviewer:**

Practice in Norway

**Interviewee:**

Secure practice. It's in Norway. It's based in [inaudible]. He used to work for Syntaf before as a researcher. You can tell him that I said you should probably talk to him that's okay. Yeah.

**Interviewer:**

Okay. Most probably accept.

**Interviewee:**

He's nice a guy. I'm sure he is happy to talk to you about this. What

**Interviewer:**

What's his name again?

**Interviewee:**

We probably have some chat thing in this. I can write it, just get the volumes some power first.

**Interviewer:**

You got my chat.

**Interviewee:**

Yep. Just a second. That's the guy. I misspelled his name.

**Interviewer:**

It's a bit strange because I know people from Ireland, but not in Ireland. Yeah. But thank you a lot. We actually had a very good paper that got the best paper award in late November last year about this technical dept and different types of perceptions. So it's quite an interesting paper we wrote and found out all this ignoring, avoiding, accepting stuff that was not mentioned before in research.

**Interviewee:**

Is it open access or could you email it to me or something? Yeah,

**Interviewer:**

I'll send you the paper.

**Interviewee:**

Yeah, I'd be happy to read that.

**Interviewer:**

Interesting. Thanks. And thank you so much for your time. And I hope it has been useful also.