**Interviewer:**

Yeah. Okay. So I would like to know first about, a little bit more about the company. I can figure out some things from the website, but maybe you can tell me a little bit more like right now are you having some financing? Is it self-standing company? Are you receiving VCs for instance and so on or, yeah, you can just mention a little bit about that.

**Interviewee:**

Yeah. So we are currently, we have launched a couple of products and we are currently earning money on the product, but we are still also funded. So we are like a mix there. So we are still running on investor funding and we plan to do that for I don't know yet, but at least a year or a bit more

**Interviewer:**

Just quick checks, but do you have like five years in the market or a bit more?

**Interviewee:**

The company is five years old, but the product was launched maybe three years ago.

**Interviewer:**

That's good. And is it growing since the last two years?

**Interviewee:**

Yeah, we are growing quite rapidly. We have had like a, a halt during Corona, but we were growing rapidly before Corona and we started to grow again now. So I would definitely say the last four years we've grown from four employees to 50 employees. Okay. And this year we will hire I think 10 to 15.

**Interviewer:**

Yeah. That's quite good. And you said it's ownership, it's like [inaudible] somehow, but with some mixture there and you have independent headquarters, I guess, where you work or is it like a distributed home office or something like that?

**Interviewee:**

We had a headquarter where everyone is working, except because of Corona everyone is having home office, but yeah, all R and D and the management is happening in Oslo. The only thing that is happening in other offices is, is sales.

**Interviewer:**

Yeah. I see. And you have had, I guess positive return incomes during the past two years.

**Interviewee:**

I'm not a financial, I have no idea what that means.

**Interviewer:**

Yeah. It means that you are not going bankrupt with your initiative. Yeah. It doesn't matter. It's just a question.

**Interviewee:**

We have a healthy economy, as long as you include the funding from investors.

**Interviewer:**

You have this core development team. How many people are actually developing software? I know you are the lead software engineer.

**Interviewee:**

Yeah. So we are, I would say 12 to 13 people developing software.

**Interviewer:**

Okay. And these are all women or all men?

**Interviewee:**

We have a part to time position for a girl and the rest are man. Really difficult to find qualified applications.

**Interviewer:**

You said we have a product like this already in the market. Can you tell me a little bit more about the core product and service that you are offering? You said there are two products if I heard it

**Interviewee:**

At core, we are a hardware company. Yeah. So we make a camera. The camera is out in [inaudible] one and two. We released the first version two years ago and we released a created version this year. So that's the main product and to accompany the hardware, we have a software development kit that includes libraries and user interface and command line tools and documentation and so on. It's bundled with the camera. So if you buy the camera, you get all the software.

**Interviewer:**

But you still have to develop the software.

**Interviewee:**

Yeah. Because this camera is going to be integrated into a big evolution where you usually pair it with a robot or a conveyor belt and, and things like that. Maybe other cameras and other sensors. So there's a integration phase where our customers need a program against our camera. It's not a product you can use for anything useful standalone.

**Interviewer:**

Yeah. And just to clarify a bit better, you are located in Oslo, you've said, and also in Germany?

**Interviewee:**

No longer in [inaudible]. We had one guy in [inaudible ] one of the founders, but he quit two years ago. So now we are only located in Oslo plus sales office abroad.

**Interviewer:**

I see. And your role, just to clarify, like, is it like CTO? Lead software Engineer?

**Interviewee:**

My title is head of software engineering. So I report to the vice President of engineering.

**Interviewer:**

But you lead this team of 12, 13 developers if I understand it.

**Interviewee:**

That's correct. We have a small part of software development that's happening in another team, which is like embedded a code for a camera. It's hard. That is happening in the hardware team, but like 90% of software development is led by me.

**Interviewer:**

Yeah. Yeah. That's good. You, so at the ecosystem and employees wise, I understand. So so I guess since you are leading the software development, I guess you are also participating into like production of the actual final version of the software testing, the release and so on. So are you doing all these activities?

**Interviewee:**

Yes, that's correct. So I've been here since day one. So basically in the beginning I did everything and now I'm outsourcing more and more, but I'm still in charge of everything.

**Interviewer:**

Yeah. But still just to make it like a measure there, would you say, like you are about 50% development testing or how much do you get really involved in terms of software component development?

**Interviewee:**

Yeah, I'm not sure, can you just tell me what is the other part? So I just know what this other part.

**Interviewer:**

You said I do more outsourcing of the tasks, but that means that you are maybe doing more project management rather than actual development.

**Interviewee:**

Yeah. Currently I would say I do 70% management and 30% development.

**Interviewer:**

When it comes to the software engineering practices, this is a bit more focused, what would you say about them? What software development practices tools are you using and can you describe them?

**Interviewee:**

Practices, we are using an agile methodology, but we are not following any of the established practices because we are too small to take full advantages of it. So we started out like with just a few people and then as we grow, we take in more and more from basically scrum I would say. I have certification in scrum, but we are not trying to follow it. So we are picking what we need and as we grow bit more and more. So currently we have a backlog grooming in JIRA. We have daily meetings. We have a sprint planning meeting where the sprints are three weeks long. We have a sprint retrospective. That's basically it.

**Interviewer:**

But it seems you are doing most of the scrum framework practices and and ceremonies. And how about tools? Do you have some specific tools you use to develop the technology?

**Interviewee:**

Yeah. So for management or managing the development we are using JIRA and GIT Hub. Is that the kind of tools you're asking for?

**Interviewer:**

Yeah. Those are mainly for versioning of the software keeping source code and track, and also project management. But how specific tools like in eDay environments that you use.

**Interviewee:**

Yeah. So our product is cross platform, C ++ application. I like to give developers a bit of freedom. So we do not impose any ideal environment or operating system. Basically we use [inaudible that's basically it, and every developer can choose to use whatever Linux distribution they want to do or windows. It doesn't matter to me. They also can use whatever ID they want to use. So I know people are using visual studio code, visual studio, sea lion EMAX and whim. People are pretty much using what they want to do.

**Interviewer:**

And this is embedded system. So you are using C ++ I guess, as a programming

**Interviewee:**

So the product is mostly not embedded. We have some software running on the camera, which is embedded software, which is written in C ++ but most of the software is running on the PC as a server or desktop software. And, but still it's developed in C ++ and in GPU programming.

**Interviewer:**

Okay. Yeah. That's very good. How about quality attributes? Which ones do you think are most important? Like performance, security, usability?

**Interviewee:**

I would say that to the priority list that I usually tell people to go for is maintainability. First it's correctness then it's maintainability and then it's performance. Maintainability also. Readability is also probably a sub- section of maintainability.

**Interviewer:**

Yeah, exactly. So when you say readability, you mean readable code

**Interviewee:**

Yeah, that's correct. Okay. So it is very important to me that the code is easy for the next one that comes after you to work with and performance wise, we do not even try to squeeze out the last 10 to 20 percent. It's not worth effort. I see much better to keep the code base small and manageable. And I think that in the long run, you get better performance anyway.

**Interviewer:**

Yeah, I see. That's good idea. And how about testing? Do you have some testing practices in place, like validating verifying the software?

**Interviewee:**

We do a lot of testing. I can say it starts with a static analyzing of code. There is a lot of different tools we use firstly, we always use the compilers with a lot of warnings where we treat old warnings as errors. Then we use [inaudible] which is static analyzers for C ++ code. For PIP code, we use, Pilot and Flake eight. In addition to we also have automatic formatting of all source code. So we don't have to think about that to minimize diffs and just don't have discussions about formatting. So that's the static part. And then we have dynamic analysis of code where we use, sanitizers for C ++, the various sanitizers that comes with the compiler. After that, we have unit testing where we do, we do some unit testing. It's not a thing we focus the most on, but preferably you should add some unit tests when you make reusable components, but we do not track the coverage of our unit tests. We want to do that, but it hasn't been prioritized. So we have unit tests and you are encouraged to add unit tests, but it's not the biggest focus. And then we have system tests, where we have more focus, since we are not a pure software company, just unit tests. We could still have a lot of bugs because of hardware and timing issues and race conditions and instability and hardware and so on. So we have a lot of system tests where we run the CI cluster locally because we need to add all the hardware we produce into the CI. And every commit is executed, on each of our products in the CI. And also we in the CI also since we do a lot of heavy computation, which is done on the GPU and the GPU environment is not as mature, and standardized as CPU. If you run something on the CPU and it works, you can pretty certain it works on the next CPU as well. Even if it's from another vendor, that's not true for GP code. So in the CI we have, GPS from Intel and video and AMD. So when we test the code in the CI, we have a matrix over all the GP vendors. So that's also executed on every commit. So that's all the automatic testing we have. And in addition to that, before each release, we have like a test suite where we have maybe five, six different pieces where we deploy the software and we connect, our hardware and we run it for one to two weeks with stress testing before a release. And the last defense is that we have a break get on before it release where most people with software competence in the company sit down for one to two hours, really just try to manually test and break the new features.

**Interviewer:**

Yeah. That's a very bright idea. And how about this testing activities? Do they take a long time compared to development? Like, which has the priority timing wise?

**Interviewee:**

So it's a bit of a struggle really, because to get this quite complex CI system with lots of hardware, moving parts stable. It actually takes a lot of effort. It's not really seen by all parts of the organization. So there's a quite high pressure for making features. So I think we spend probably 70% on features and implementing tests and probably 30% of our time go to just making the CI stable and robust and maintain it and extend it.

**Interviewer:**

That's interesting. And how about documentation wise? You said something about readable code?

**Interviewee:**

Documentation is really discouraged. Basically we have public, we have public documentation of course, for our interfaces and that documentation needs to be really good for our customers. Internally, the code should be structured and self explanatory. We have some very high level design documents, maybe few lines here and there but we try to discourage it. Also its going out of date. So usually a place to put your documentation. If you want to explain what you have done, we'll do it in the commit message because then it's, it's kind of understood that this is outdated information immediately. If you put it into a you expect it to be up to date. But generally, the code should be understandable and stand on its own feet. So no inline comments and no documentation for functions. The only exception is if you write any code that is surprising, like if you read the code and you will be surprised, then you need to add a comment about why this code is surprising. Yeah. Like if it's just out for no reason put a sleep in your code, then you need to document why do we need to sleep here? Because usually that's terrible.

**Interviewer:**

To move bit forward about this technical debt and pivoting themes, I guess you are well acquainted with technical debt. I don't have to explain it in any sort of way, but it's still debt related to how well you, actually perform the software practices, the good software practices and the pivoting side is how much you have changed directions in the company. We change direction for different reasons, but let's start a little bit with the technical debt. How much aware are you, as an experienced, software developer about technical debt within the company?

**Interviewee:**

I would say I'm aware of everything.

**Interviewer:**

Yeah. Including technical debt?

**Interviewee:**

Yeah. Especially technical debt. I really care about that.

**Interviewer:**

What is your perception about that?

**Interviewee:**

What in my company?

**Interviewer:**

What is the perception and how much technical debt you have? Do you accept it? Do you ignore it? Do you avoid it? What do you do?

**Interviewee:**

So first of all, we track it. You cannot commit any technical debt to repository without adding a comment in the code that this is technical debt and track it in a [inaudible] So that's for one. And the other one is that we have a high focus on avoiding getting technical debt. Yeah. So it's expected that people do the task terribly and take the time to not leave technical debt. So people are encouraged to do that. We also have a big focus on moving forward when it comes to tool chains and so on. So whenever there's a new version of a tool chain, we jump on it immediately so we can get small increments instead of trying to jump four/five versions ahead every fifth year.

**Interviewer:**

And was it like this also before, like when the company started?

**Interviewee:**

So I've been here since the company started basically. So this is a spinoff from [inaudible]. So there was a code base, but to be honest, the code base, when I started, was it was like a research code base, not remotely ready for production software. So, I would say the first year was spent on cleaning it up to get it into a proper state. And then I would say after that, we didn't have a tech debt. So we started with the blank sheet. We have of course accumulate some tech debt, but we also take the time to fix it eventually. So for example, if you have a deadline and you need to take a shortcut to meet the deadline, like fixing up the tech debt after the deadline is prioritized as a high priority task.

**Interviewer:**

I just made this question. How about pivoting? I would like to, discuss this in sort of detail. Pivoting it might mean several things. It's not just changing direction for one simple reason. It might be several reasons. And research wise, we have a list, when I say we researchers have a list of why companies do actually pivot and it doesn't have to, apply to every point in the list, but some of them, or one of them at least, and I would like to see, and if it's possible to discuss the connection to the technical debt of the pivoting situations. So the first one is about zooming in like, you have a feature, single feature of your product that eventually becomes the whole product because you see the other features are not really very, very important, but you think that this feature is more important and we are gonna shift the product towards this feature. Has this happened so far?

**Interviewee:**

No it has not happened. We have discussed many times, especially in the beginning of whether we wanted to go broad or we wanted to go more narrow and be really, really good at what we do. And we have chosen to go narrow. So it means that we have not tried to make a lot of different things. We are a bit quite conscious about that we are like 10, 12 developers. It's not a lot. You cannot make five, six products. It's just not possible. So we have quite a big focus on just being the best we can on this camera and the SDK for the camera. And it's often tempting to make surrounding products. But we have not done that. So I would say this has not happened.

**Interviewer:**

And the other one is like, sort of the opposite, like zoom out, like having a lot of feature, but then you realize, okay, this amount of features can be like joined into one single product. Does it make more sense?

**Interviewee:**

Let's see. I'm just reading.

**Interviewer:**

Has the whole product become a single feature of a larger product?

**Interviewee:**

No, I would say that has also not happened. We are staying very true to the original plan. Yeah.

**Interviewer:**

How about customer segment? Because I think you have a market and you are selling products, so has this customer segment changed over time?

**Interviewee:**

Yeah, they probably have a bit more, so our customer base is very much driven by the capabilities of the hardware and not the, the software. So our software can be a differentiator when it comes to how good it is and how easy it is to use and so on. But if the hardware is not good enough, or if it's too expensive, customer will not buy it regardless of the software. So software plays like a second role here. It's not that important. When it comes to new segments, our goal here is really to try to support what the customers are using. Our software must be integrated into a bigger environment. And we cannot dictate the terms really. So whatever is common in the customer base try to support, and here we have some learnings that we have started out to support different programming languages. There's some standards we can implement and so on

**Interviewer:**

Because this is more like a customer segment direction because it comes a little bit later this changing the code base, perhaps. But the customer segment you haven't changed it so much until now. For instance, you are doing the camera development as hardware, plus some software, but you haven't shifted towards a different product or something like that?

**Interviewee:**

No. We're not.

**Interviewer:**

So you cannot, but then it is customer needs. Like your company has realized the problem you are trying to solve is not important and discovered that other problems are more important. Has that happened? My guess? Not. Like, you are solving the same problems since the start.

**Interviewee:**

We have a lot of technical issues that when we start to grow customers get more and more demanding, we get bigger customers that are more demanding than the first small customers. And then we discover that our product is not good enough in this particular field. And we need to improve it. I don't know if that is related to this question.

**Interviewer:**

But is it like connected also to technical debt somehow? Or why isn't the product not fulfilling all the needs? Why is it more demanding?

**Interviewee:**

Yeah. So it's partly related to software, but one thing we do is that we build the camera, it's a physical device. And what we try to do is to make a 3d image of a scene and each camera is unique. So to get like the accuracy that customer want to do every single camera is calibrated, customly calibrated. And this calibration, it's a mixture of software and hardware where you run software and you take a picture of a scene, you put a camera in a heating chamber and you take up and down temperature, you shake it and so on. And then you try to calibrate the camera and this method is it can never be perfect then we would be rich, but it can always be better. So I would say that the method we are currently using is not good enough. I don't know if we can consider it technical debt, but this is something we should have fixed probably before. And it's been delayed because it's very complicated.

**Interviewer:**

Yeah. And how about the platform? Application has turned into a sporting platform or vice versa or some sort of situations, because maybe not because it's very specific embedded systems that you are developing mainly and some stop applications.

**Interviewee:**

That has not happened. So I can say, I don't know which point it fits into, but I can say in the beginning when we were taking it from like a research project to a production software, the initial software written only for windows, but in the market, like the big customers are not using windows. So there we had to change out some technology in the software cause we are relying on windows specific features.

**Interviewer:**

Yeah. So yeah, I can jump that's no problem. So if you want to discuss a bit more technology pivot? So you think you have changed a bit, the technologies from the start and it was windows at the beginning now it's more several platforms?

**Interviewee:**

Yeah, that's correct. So in the beginning, just to get like the prototype up and running, when it was a research project, they kind of took some shortcuts and they happened to be windows people. So they developed a windows product.

**Interviewer:**

Maybe that's some technical debt, I guess.

**Interviewee:**

Yeah. That's fixed now, but we basically had to replace one tech stack, basically the entire GPU stack was windows only.

**Interviewer:**

This was because developer, like you said, knew only how to develop software in windows?

**Interviewee:**

I think the reason was not because they could not have done it portable. It was to get quicker. It was the easiest route for research project. You just want to like demonstrate your idea. You don't need to make a general solution. You want to make a paper and get the best performance at the quickest. And then it's easier to use to select the technology and also to select something that maybe is very, very good, except that you lock yourself in.

**Interviewer:**

I understand. And so let's say this pivoting happened because of technical debt.

**Interviewee:**

Yeah. I would say that.

**Interviewer:**

I don't have to go through the others, maybe value capture, have you changed the value, how you monetize business architecture, how you maybe change the strategy you have. Like instead of focusing on large market, you focus on a small market, but of course you charge a bit more.

**Interviewee:**

I can say that for value capture, we are currently giving away all the software if you buy the camera.

**Interviewer:**

Why is that? Why are you giving out away all the software while you charge for the camera?

**Interviewee:**

That's what I'm getting at. We are discussing how to do this better, but because we have a quite narrow focus, I would say considering now that the camera is useless without the software. The software, isn't more isn't than the minimal viable product. There's nothing there we can take away and still have a working product. We plan to add services that will add value on top of this, but we have not gotten there yet.

**Interviewer:**

Yeah. Maybe I'm not in place to say just out of this discussion, but perhaps it could be a good idea to integrate software with cloud solutions.

**Interviewee:**

Yeah. No, that's exactly what we're planning. To add recurring revenue and some subscription model and cloud services, but then we need to have some additional value.

**Interviewer:**

SIS Software as a service might be a good business model. That's out of the interview scope, but it just popped up in my mind. But how do you explain since we are at the end of this discussion, but how do you explain the role of technical debt in your company? Pivoting situations scenarios in one or two sentences. So I understood you had technology pivoting apart from the other nine. So how do you state a few sentences and say, okay, this is what I feel about it or what I think about it.

**Interviewee:**

Yeah. So I have a very conscious mind for technical debt. So usually we try to avoid it, but we see sometimes to get out product in a reasonable time. We need to take on some technical debt, but we have quite big respect for the cost of technical debt over the long run. So even though we have gotten the product out, we build and prioritize to fix up the technical debt because we don't want to bleed forever because of that deadline.

**Interviewer:**

Yeah. And do you think if you didn't do like this, do you feel like at some point in time it'll make your company pivot towards a different software solution

**Interviewee:**

No, that's not what I'm mostly afraid of. What I'm mostly afraid of is that if we have a company with, or code base with a lot of technical debt, it's not going to be a fun place to work, and we are going to lose the best people. Right now, I'm very happy with the employees. We have really, really good employees. I use this a lot in recruiting. I'm able to recruit really skilled people and skilled people that can go everywhere. They will not stay in a company and fight technical debt.

**Interviewer:**

Yeah. That's true. Very bright idea. Yeah. But yes, I think for me, this discussion has been very useful and I would like to thank you so much for it. And do you have something to add yourself or something that you had in mind? Was this interview what you expected or was it different from what you expected?

**Interviewee:**

To be honest, I didn't expect much but I thought that it could be a learning experience for me based on what you ask me since you ask a lot of startups questions and so on. I was just saying yes to this because I was curious what I could kind of learn from it, so I didn't expect anything and I didn't have anything I wanted to say. I just wanted to see what you care about and see if I can learn anything from it.

**Interviewer:**

Yeah. So what do you think you learned something or was it like normal interview?

**Interviewee:**

It was not a normal interview because I participated in things like this before. I didn't invest a lot of time in this, so I think I learned enough that it was worth my time. It was interesting to see what you care about.

**Interviewer:**

But yeah, thank you very much. We actually have some publications on this topics and some were very well received, like one technical debt paper got best paper evaluation last year in software business conference and we are planning some journal papers. I can share it with you. And it talks about like the things you say here just to make a comparison I've interviewed companies in US and pretty much when I see your state of the company, like 12 or a bit larger development team you say almost the same things as companies in United States about technical debt and how you perceive it, how you deal with it and software engineering wise I would say, yeah, it's almost the same practices and the same way of dealing with software. So I see some compatibility with the level of growth of the company and the actual software engineering practices, which is I would say, which is normal. So this is something that is interesting as well, so that kind of research go very well received. So I'll just share it with you if you want to read it, that's fine. If you don't have time, it's also fine.

**Interviewee:**

It's very interesting. Just share it with me. I would like to look at it.

**Interviewer:**

Thanks a lot for taking the time and talking to me today. Just keep in touch and I'll just have to close this because when I do end meeting, it stops also puts the recording on cloud. So thanks a lot and talk you soon.

**Interviewee:**

Okay. Thank you.