**Interviewer:**

Okay. So yeah, if you want to say something about the company first, you are more than welcome. I bumped into the retro-like I said and I think that services you are providing are quite interesting, but of course, I have some checkboxes to fill in first, like is the startups, for instance, in serious A financing, which means like, are you at least getting funded or you are self-sustainable or what is the status of the startup, the company right now?

**Interviewee:**

The state is that we are looking for our first fund from our VC, our angel. We are previously been funded by the Research Council of Norway and Innovation Norway with $1,080,000 from those so now we need proper funding. The status is like I said, that we are, we are growing users but we don't have funding to work on this full-time. We are now five people who are working on it. And so we need proper funding if we need to work full-time. So right now we are part-time aside of our jobs.

**Interviewer:**

Hmm. Now it makes sense. So you are founded in what year?

**Interviewee:**

About 2017.

**Interviewer:**

Yeah. So it's less than five years. Do you think you have entered some sort of growth phase during this period in the last few years, sort of like having more clients and more customers

**Interviewee:**

The last year when we have been part part-time on it we have had more growth than we had all the previous years. And the reason we have more growth now is because of our jobs, our normal jobs, are get getting us access to networking because starting a startup, you don't have any networks and the other parties that are app and are starting to get good.

**Interviewer:**

I'll ask you a bit more about the product. So you think you are growing so more or less, how many customers do you see in the past few years you have acquired like five to 10 or, 10 to 20 or whatever, just the range.

**Interviewee:**

It is hard to say hard who are customers for us. We have now 560 users.

**Interviewer:**

Just a rough number. It's not very important. So are you self-owned? Like, do you have some headquarter, or is it like, just you bootstrap? As you say, we have jobs, but we do this online, this job for the startup.

**Interviewee:**

Yeah. It's from our home office and online. So we are fully distributed earlier we have had it. We have had an office earlier where it was co-located. We were full-time to the end of 2020, I think. Or to the end of 2019. So in 2020, we, all of us got full-time jobs and went to continue doing this part-time

**Interviewer:**

Okay. Yeah. But that's smart. Do you have any positive income so far?

**Interviewee:**

Not positive. Not registered this as positive, what should I say? I think it's registered in the official economy as zero income. We do have resources in what is called a company bank that is positive. But as a startup, they place your costs in a smart way. So we don't wanna tax our small income. Right? We need to use that income, spend that on marketing hiring people that can do stuff for us. All sort of related stuff. So I don't know what to say there.

**Interviewer:**

The startup at the core, do you have a development team?

**Interviewee:**

Yeah. We are four developers and one guy who is not a developer.

**Interviewer:**

This is just the check because I have to sort of, as you said to me in the email if I'm interested, I'm actually interested because you have almost four years and it seems your product is getting better and better and you're getting more customers. So that's a positive sign. If you ask me in the grow, it doesn't have to be translated into income, but it has to be translated into at least a growth perspective. So the core product and service, do you want to talk about that a bit, even though it's on the website, something special that I should know of? Do you want to share the screen maybe?

**Interviewee:**

I can do that. It depends on how much information you want.

**Interviewer:**

Just the basics. Wow. What I can't see from the website

**Interviewee:**

I can share my screen. Can you see it now? Our basic idea was started with retrospectives and I don't know if you are familiar with retrospectives, but retrospective is a part of the agile process. So for example in the scrum. If you look at the statistics we see now that there are a lot of people that are now using retrospectives and that has been steadily growing for the last 10 years. So if you look at the market, it looks like the market for a retrospective is growing it is also growing towards different industries rather than just IT. Our main target is the agile coach or the scrum master before we started trying to get sales B2B but sales in B2B are so slow and we don't get any feedback. If you don't get any feedback, you can't even improve our product in the right direction. So targeting the agile coach or the scrum masters or project leaders has so far been shown as a better path for us. So then we are sort of B2C towards that group, but in reality, this is a product used by businesses. But of course, agile courses, scrum masters is number one developers, designers, people that can use feedback in some sort of ways and use retrospective kind of process is our main target. Our app is just focused mostly on that. Just easy ways to gather feedback and run retrospectives and just share links with people where they can join the retrospective and do team health checks. The main, basic ways of doing that are asking what works well, what doesn't work well or ideas like activities like this, or a timeline activity where we kind of map how happy or dissatisfied you are with the last working period. So that is the base.

**Interviewer:**

Yeah. Is this too integrated with some of the tools like JIRA or maybe Trello or Asana?

**Interviewee:**

We do have an API. So if you're a programmer right now, you can code that integration, if you want. We don't have this seamless integration yet. Where if you are a non-developer, you can just click a button, and then you are set up right now, we have to use Sapier or being a developer to do that integration. So that is on our horizon, but we are noticed that most of the actions you have at the end of a retrospective, don't fit in a regular Kanban board like JIRA. For example, most people do not want those types of actions in their JIRA board because they're so different from normal development tests. But again, some people want that. So of course, we gotta set up to do that in the future, but right now we are mostly focused on the user experience in the parts that we already have and asking people, what is our competition doing? Are you using them instead of us? Why are you using them instead of us? And based on that, we know where we have to prioritize.

**Interviewer:**

It's just an idea of, because some of this ceremony, this is a ceremony retrospective anyways, that you have to follow in agile. So I think it's very bright that you have done this, but it needs integration with other major players that provide agile perspective, but this is just the feedback from my side, so you take it as it is, I guess.

**Interviewee:**

Feedback noted.

**Interviewer:**

Yeah. I think the product is very interesting. I haven't seen it before, so I think it is quite an interesting product. It might have a good potential, you know, better in the market. So that's pretty cool. What is exactly your own role? I didn't quite catch it. Are you a Developer?

**Interviewee:**

I'm what you call a potato. I do whatever is needed to make things go forward. I'm not really, my education is more towards business development rather than, rather than IT and being a programmer. We do have programming courses. The reason I think I'm the CEO right now is that I have the clearest vision about, what the user needs. And I can share those use with my team and I can understand my team and I understand what they need. So my most important role is to understand the market, communicate that to the team, and understand what my team needs. So my official role is the CEO, and I think that is an important part of it, but maybe 60, 70% of my time is development. And that is really difficult. And it shouldn't be that way. But right now we don't have any choice because just gotta do whatever it takes.

**Interviewer:**

Yeah. But that's fine. I think in the start anyways, it's like that. And you play also a little bit like a scrum master, perhaps for the team to develop stuff. If you are doing it at trial, I guess

**Interviewee:**

In the beginning, I was, I was a scrum master/ agile coach, but we have done this so many times. So now we have a rotating coach role. So everyone is rotating. Yeah. We are rotating the role of the coach. And the role of the coach is to run the process, to run the stand-up retrospectives and all the normal scrum events. But in my normal job, I am an agile coach. I do courses for new companies where I use my tool and demonstrate my tool. And I'm also a part developer there as well.

**Interviewer:**

Yeah, that's good. That's excellent, actually. What is your team composition in terms of, are you men or there are some women, or is it like a mixed team?

**Interviewee:**

Oh yeah. We're just men. I can't find any lady programmers or female programmers or female people that we need right now, importantly.

**Interviewer:**

I understand. The gender balance is not so high.

**Interviewee:**

Yeah.

**Interviewer:**

Do you participate practically in the product life cycle such as product development testing release? Yeah. And you said it also to what extent? I guess, so I don't need to ask more about that, but when it comes to software development practices and tools that you are using, can you tell me a bit more about that just briefly? How do you develop the product?

**Interviewee:**

How do we develop it? The beginning was on the timeline. You see the timeline right here. Ceremonies like retrospective and use of timeline and the possibility to digitalize these kinds of methods. We looked at the market. We saw that the tools for retrospectives had a great potential to be a lot better than they are today. And I still believe that we are just at the beginning of that phase. We create a prototype, which we demonstrated to a company that was using retrospectives at the time. Our product was then at the very early stage with only this method you see right here, it continued to develop these other methods. That was more of the categories and the notes, where you can add notes in different categories as a way to get feedback and the whole retrospective, typical process. We gained our first paying customers quite early. They were using our biggest competitor and we manage to get in quite early after that, it was kind of stale for a while. We continued to develop and just focused on whatever feedback we had and the users we had at the time and was very focused on just prioritizing, always prioritizing whatever we felt like the users needed the most. And we are still just always focused on prioritization. And the prioritization is based on how much feedback we get on things. So for example, you now say that you believe that integration with Trello for a year, for example, would be good. The more people that say that the more towards that direction we develop.

**Interviewer:**

I'm also part of this customer-driven course, at [inaudible]. I'm actually supervising some teams their agile teams and they work with customers such as Syntac for whatever you name it from the industry and I think it'll be interesting if I bring up your product during the course because anywhere they do retrospectives, they do ceremonies. So it might be useful for you afterward to say this is another tool in the market that might be interesting for students to check out. I know it's not directly connected with the business model, but it's still relevant, I guess, for you to get feedback from the students that use it in a real setting, real-life setting anyway because they're developing projects. So I think it should be interesting, but we'll discuss this, I think at the end again and afterward, I can bring it up with the course leader as well about this tool. She will be very interested.

**Interviewee:**

Yeah. Two things I wanna add to that is that there are always students at NTU and IO that are using this product and all students use it for free whenever they want. So just log in and create a user. It is always free for students. And the other thing is that in our process of making this tool we actually made a big mistake at the beginning which I can talk a little bit about. And that was what we started to develop before we had a finished design of the tool. I learned now that having a design before you start to develop can be a real-time saver. So right now that we are developing our new design for the next version of a tool of course designing a tool first and getting feedback on that for users have been a big part. And that is maybe I think one of our biggest mistakes.

**Interviewer:**

Yeah. But it's fine. You learn by it. So it's a learning curve I think it's no big deal. Anyways. You have to update if you design today in a couple of years, it will be old anyways, or in one year it'll be old, so you'll have to upgrade. So that's totally fine. But what kinda tools have you used for development? Was this developed on the cloud system or was it like in Python, in C sharp, or what is the tool about?

**Interviewee:**

React in the front end and it's node JS in the background. The reason for that is so that you have JavaScript all the way in one language, easier to learn. We have [inaudible] as querying language because of its ability to be easy for front-end developers and tackle high loads. MongoDB as a database. We have [inaudible] as bundlers. We usually show pseudo-code for development. We have this live share thing. We use a lot of peer programs and we share our screens. So two people can code on the same thing at once. We usually use that at the end of whatever we code and we use the digital ocean as our server host. So it is in the cloud.

**Interviewer:**

Yeah. That's very good. And in terms of quality attributes of your software system what have, you thought about like UX performance security or usability, for instance, what kind of features attributes you for seeing the product? Like, are you emphasizing performance for instance, or security of the product usability like UX, reusability, whatever

**Interviewee:**

Usability is priority number one. We are also highly focused on security. We do think a lot about your security in terms of the quality of the code itself. When we started to do this, we hadn't any experience with React how we should structure or code the best possible way. So of course that could create technical debts over time that we need to brew over time. So I think, I think we have, we have been much better at security aspects and usability than actual structuring our code to be as simple as possible in the long run because we had to learn React, and we didn't know how to structure a code as best possible as we do now. And right now we are also starting to learn to automate our tests because our philosophy is to have a sort of a minimal toolset as possible. Right now our manual tests are so long that we now need to add automatic tests. That wasn't something that we began with because we wanted to push something to the users as fast as possible. And we haven't had any prior knowledge of how to best use automatic tests with these kinds of toolsets. So we add those sorts of quality checks as we develop and grow further. And that of course creates some technical debt for us. But we usually combine the rewriting of code or adding tests whenever we change some design features with the app. So when you change the design, we also have to rewrite our code. And that is a good time to just improve the code and add tests as well in one single sprint.

**Interviewer:**

I understand. And somehow you got into my next question about testing practices and you say we do manual testing, but not really regression testing or something like automated testing for our product. So you are moving towards that. Amazing.

**Interviewee:**

Yeah. We have unit tests in our backend right now. We don't have API tests yet and we don't have front-end tests yet. So our manual test plus backend unit test is where we are now.

**Interviewer:**

Yeah. But you don't have to reinvent the wheel anyway. There are a lot of tools like Test Team or Swagger, open the eye, whatever, and they test the things for you I guess.

**Interviewee:**

To some extent. Yeah.

**Interviewer:**

How much time do you see you invest in testing activities? Is it like 50%, 40% of the time, or less?

**Interviewee:**

That is a difficult question because in a team I've seen that we have so different approaches to it. As a more frontend-focused guy, I test continuously as I develop because I want feedback as fast as possible. And when you're working on the backend the guy who is more focused on the backend than the other people, he's not so focused on manual testing, whatever he does. He has this process of creating some kind of new thing, me and him together, or him and the other guy that is, some other developer test things with him. So we always do pair programming when we test things, and change things together based on whatever, however, the test goes. And then we test manually often, the guy who is not developer, he's kind of, he has this dedicated role to being a tester because it is very useful to us, for us to have one guy who isn't a developer that is also a tester, because we, as a developer, we use the system in a different way than the guy who is not a developer.

**Interviewer:**

Yeah, definitely. You develop with a certain mindset, but then you have to be in a different mindset. From what I understand you are trying to be an agile team. Correct me if I'm wrong on that because you mentioned quite a lot of peer programming, which is coming from extreme programming, but any other agile practices you might think of that you're using? Is it common that you do friends or whatever?

**Interviewee:**

We have different processes from when we are co-located to when we are distributed. When we are co-located, we do signups every single day we do retrospectives every Friday, every, so every single week. So we have a high focus on new ideas and experimentation. We do sprint planning every other week when we are distributed right now, we do standup once a week. We have retrospectives once a month and we do sprint planning whenever we need to. And we do demos for each other whenever we need to. So we don't follow the usual scrum sprints, typical sprints. We do sprints instead of we have retrospectives in regular intervals, but sprint planning or demos, we do that whenever something is finished and the reason I don't have hard sprints is that it creates stress on the team that is very difficult to deal with when you are part-time and have a full-time job as well. And for us, it is more important to prioritize always the important things first, rather than creating, like a time limit on different things we develop. So it is based on scrum and uses the Kanban board in Trello a lot. We have moved a little bit away from the normal scrum.

**Interviewer:**

That's also fine. As long as you have smaller scopes, perhaps, I guess you don't have to do sprints with a lot of person-hours. So maybe it's fine to do shorter sprints with fewer tasks that's up to you. But how about documentation? Do you document anything at different phases of development and testing?

**Interviewee:**

Yeah, we have been forth and back when it comes to documentation in the beginning, we started documenting manually by commenting in the code. And we started looking at JS docs for commenting, but in recent years, the new way of doing documentation is just writing tests because tests are documentation in itself and more and more people are moving away from the JS docs and we try to make the code as readable as possible in itself, rather than doing documentation. We have started using a design system. So we are moving components up to this design system where each component should be described, but that is in the beginning phase. We are now starting to move to a design system and test in the front end, which we haven't had earlier. So we are progressing towards that. But right now, in terms of documentation, we have this confluence where you have a short Wiki about some practices other than that it is some tests, unit tests in the back end. And the code should be self explainable as possible. If it's not self-explainable, we add some comments. Yeah,

**Interviewer:**

I'll move on to the next phase of the interview, but as I said before, this customer-driven course might be interesting for you also if you want to develop more features, maybe this helps, but let's say the students anyways develop different software or different companies. So it might be interesting if you want to join as a customer for them and try to develop a certain feature of your software, existing software, but this is up to you, as long as you think about it, it's just resources that you get, you may ask for. They do quite a good job. Anyways, you have been an engineer before, so maybe you didn't do this course customer-driven, but I think it's also interesting for this particular tool to have some because we are getting anyways, the same customer every year, I want to change things. I want to introduce new companies.

**Interviewee:**

There was someone that suggested to us earlier that [inaudible] has this customer-driven project in some kind of subject they have so we applied for it as a potential customer and we made a task for the students.

**Interviewer:**

Yeah, but now I'm an inside person. So I can say, the company is doing a good job because I've seen the product. I don't think we have written a couple of papers about the process. So I have written actually about how the companies are selected and we are a bit disappointed, some of the results, so yeah, maybe something needs to change.

**Interviewee:**

To change. Yeah. Maybe. The last task we wrote was this, the news sky on the team created this recently. So this is just a simple profile page with some details and some statistics about our retrospectives. So this was a student task that we applied with. So this is a separate part. So as students, we then will then get access to our API and we'll develop and react and create something like this. So that was our task.

**Interviewer:**

We applied again this year and you hear some good feedback about that.

**Interviewee:**

Send me information about it.

**Interviewer:**

So the last part, it's a bit more focused on technical debt and pivoting. So I know these are two terms, but technical debt I assume you were aware of you understood the concept, when it comes to pivoting, it's sort of changing direction within your company for different reasons than we'll discuss. Maybe it'll be easier at the moment that I shared this screen with you and, have some specific 10 questions about pivoting, like changing direction for different reasons, like zooming in zooming out, or, whatever you change the customers, customer segment, or customer needs, et cetera. So, first I want to ask this question and then I will share the screen at the moment. The first question is, how much aware are you about the technical debt within your company or your startup, the awareness part

**Interviewee:**

I don't know what the maximum level of awareness someone can have about technical debt, I think I have a good idea because I know the code base myself. I know all of it's pretty good, so I know which part should be rewritten. I have a very, very good insight into what's technical debt that we have, but I don't know what kind of practices other people have if they have some tools that can help them do that. I think for me, technical debt is subjective because we ourselves see that we want the code to be in a different state than it is now. And therefore we have technical debt.

**Interviewer:**

Yeah, no, what I meant was more sometimes it's also about the fact that some developers take shortcuts because it's easier to do it in a certain way, and maybe it's not the best way, but it's still a good solution for the moment.

**Interviewee:**

We have taken a lot of shortcuts earlier and that is intentional because there is this concept where, what if we have this tool we can install with MPM, for example, that do a lot of the job for us, for example, this can be a time picker, a calendar, for example, where if you use, and we download that. We use that. We apply that in our app and we know that in the long run, we have to rewrite that because it doesn't cover all of our needs, but we do that regularly just to push something to our users and to get feedback on it. So then we create value in the meantime, and then we can create, but in the long run, we are using more time, but we create value in the meantime. And that is, that is something that is really difficult. I think. So we do take short sometimes and we just push things out to just see if it works well. The problem is how good are we at rewriting or optimizing or improving that code at later stages? And in my experience in our team, it takes quite a time before I look at that code again. So the rewriting of the code is quite delayed. I think so. Yeah. I'm pretty aware of it, we've tried to optimize it. I don't know what ideal way to practice that. I think that we are okay, when it comes to technical debt, you do some shortcuts and we are aware of it, and we are aware of what part of the code needs to be improved.

**Interviewer:**

Yeah. And what is your perception? Because my next question is, what is your perception about technical debt and how do you cope with it? Like, do you ignore? You said something now, like, do you ignore it? Do you accept the technical debt? Maybe you said, accept, do you manage it or do you try to avoid it? So first, what is your perception? Is it something like good or bad?

**Interviewee:**

Technical debt will always happen. I think. I think it depends on your perspective, I think because sometimes your perspective is in the long run and then you are what is called less affected by technical debt. But sometimes you just have gotta do something in the short run. Maybe we have a hotfix to gotta move out, for example, and that takes to something. Then we create technical debt for ourselves, and we just need to do that.

**Interviewer:**

Maybe this falls into the category of accepting technical debt?

**Interviewee:**

I think we accept it when it's needed.

**Interviewer:**

Do you try to avoid it at a later stage, I guess after some years doing the product four years?

**Interviewee:**

I think right now we are much better at avoiding technical debt than when we were at the beginning because we can now code much closer to whatever the code should. Whatever state the code should be then we did before. And that is because we have grown as developers over the years. We are still young. We have now three to four years of actual coding experience after the university, you know? So still learning

**Interviewer:**

That's still good. You don't have to have 20 years of experience.d. So you said at the early phases, I think you answered my other question, like at the early phases, you are sort of accepting it, but now it's more like to avoid technical debt and manage. Manage the situations. This is what I perceived the, oh, from your answers. I want to share the screen for a moment because the next questions are more can I stop your sharing, I guess?

**Interviewee:**

Okay.

**Interviewer:**

The next question is more like technical debt and pivoting, like changing directions, because technical debt affected some situations with your company and your software solution. So the first one is zooming in and what it means is like, this is a pivoting situation. You have a single feature of the product become the whole product itself. And if, yes, you can say, even no, if you ask then explain how the technical debt played a role in this situation. And then of course I have some others like zooming out and we can take them one by one, I guess. So what do you feel about zooming in?

**Interviewee:**

I've heard a lot of companies that have done this and we have not done this yet. In terms of pivoting one thing that might happen is that I see now that our app is very useful to just get feedback on whatever product. For example, we create all sorts of different products and I use our product to get feedback on our prototypes and our finished product because it's so much faster to send a link to a lot of users. And I have them just create some feedback notes in our app. And then we do this video session where they explain to me whatever those notes were about. And they can show me directly in-app, for example. So I use this as some sort of design thinking process type of thing. And I think it might be so we, we haven't pivoted at all yet. So the answer there is no but, I think it might happen in the future that some sort of touch of the system is more used than the other. And we just focus more on that.

**Interviewer:**

Yeah. But then the other one is like, has the whole product become a single feature for instance? Like you have, when I hear at your solution and you are talking about retrospectives and it's a tool about retrospectives, and I'm not sure when you started this project, was it just about retrospectives or was it like the retrospective was a single feature of a broader product? That was dealing with agile practices, like let's say retrospectives stand up meeting, and then you had also some other events or ceremonies, and then you said, okay, we just take retrospectives and that would be it, or vice versa.

**Interviewee:**

It tells just always been retrospectives. It, for us, has almost been the opposite way because now we are looking broader. We have tested our product in different ways. I mean, now, although looking at other, other types of customers or usages for our system, that previously was just retrospectives, but now it is also feedback in the design process. One company also uses this for its action plan. So the leadership is using this as an action plan, follow-up tool.

**Interviewer:**

Yeah. And then you said something before I think about this because you are not saying, oh, we haven't pivoted it at all, but you mentioned something about changing the customer targets.

**Interviewee:**

Yeah, yeah.

**Interviewer:**

So customer segment from how, while trying to solve the right problem, you start different segment of customers than the original one. So maybe you move from B2 B to B2 C maybe on B to whatever government that depends on you. Please connect it if you can, with the technical debt, if it is connectable

**Interviewer:**

First, tell me how you have pivoted with the customer and why?

**Interviewee:**

It is pivoted from B to B, to B, to C and the customers you are now focused on are agile scrum masters and maybe more and more designers, of tools they develop. The reason we moved from B2B is that B2B is so slow, so it is not technical debt in sense of programming stuff, but it is technical debt in sense of sales resources. B2B sales take so many resources from our team and B2B sales mean that I have to do sales and manage those relationships rather than programming and coding and getting it and creating better products. It was for us right now, too slow. So we pivoted towards B2C rather than B2B,

**Interviewer:**

But if I can make an example, because I know these concepts are not very easy to connect. And if I can make a small example let's say you introduce a lot of technical debt with your product, which means that the product becomes so hard to change. At some point you cannot even change it, let's say for the business side. So you have the B2B business model, and then you have the technical debt because programmers were taking shortcuts to make it work for the businesses, et cetera. But when you go to the businesses, they have a lot of different feedbacks. And when you try to adapt your code, then you realize, oh, but we have taken so many shortcuts. So now it's very easy to adapt to their needs, but I'm talking from a software perspective, not from a market perspective or whatever. So has that kind of situation occurred sometime in the process?

**Interviewee:**

I can say that one of our main goals in the beginning when we created this was to be when we started something that is retrospective the core of our business was agile. We also created the code to be as agile as possible. So we want to be able to change every single part of the code as easily as possible. That is a part of the reason why we use [inaudible] for example, and we don't have hard risk APIs, for example, because it's easier to change the same with the code. We have made the code I think a lot easier to change maybe than others have done. But for example, there are parts of the code that we have gotten feedback from the customers that are not good enough yet that we have already changed at least one time earlier. I can just share my screen again. I'm giving a practical example of something like that. So this is a retrospective. Here is our feedback in three different categories in retrospective. Normally notes across these different categories can be about the same subject. So when we are starting to create some kind of solution for problems or improvement ideas, we group them in the groups that we have on the left side. Previously we had this like the first prototype was in the very first prototype, was when we created this using this grouping section was a whole different beast. Then we moved to develop our actual app, to react, we had to of course rewrite that code, but it wasn't a lot of code, we have so much more code now, than we had with then, but this section right now, this grouping section, even though we improved it back, then we still need to improve this because it isn't easy enough for, for new users to understand this. And it's annoying that you can just move one of the nodes and it's sort of restricted in terms of how people want to move notes, group notes and be full of flexibility in terms of that. So now we have to move towards sort of this free grouping, where we can grab notes by dragging them on top of each other, and just moving groups around as, as you feel like it, and this is a big change for us. So this creates a technical depth for us because we know we have to change this and we have already changed this before, and it's still not good enough. This section right here, when we now have this free grouping, it started to look more like some of our competitors that are, it is this free system where we have sort of this free space where you can move all sort of objects around on a free space. And we now think that this being able to zoom out and grabbing some kind of note right there and just moving around and we see that now at that point is really good. And you should sort of um adapt that sort of way of grouping and being free, being free to be able to have this free space to the group and create sort of connections between notes. So that is sort of a pivot, but that is the song part of our product. I don't call this a whole product pivot, but is a pivot of a part of a code

**Interviewer:**

No, but that's fine. I think we can explore a little bit more to this because from a research perspective, there are 10 regions you can pivot. So I can share the screen again if you wish. I get your point when you say, okay, we haven't changed the whole product, but we are adopting it for different needs for different customers and also for being in a way better for the end-user. So this is what I understand now. But please let me share the screen again. So, the customer we do need, we understand that then platform, I think you for maybe a little bit on this platform Pivot your application turned into supportive type platform or vice versa, maybe because it's still supportive platform anyway, however you change it. From what I understand business architecture, your startup has switched to business architecture, like aiming for low volume with a high margin of interest. And instead of focusing on the mass market or vice versa

**Interviewee:**

Our pricing is going to change. The business architecture we have, I don't know all the things that go into business architecture, but previously, we didn't want to have this free version of a project because we have this business mentor that said to us that we shouldn't give anything out for free because there are so many businesses that struggle to gain money as they give, something out for free. But we now see that that was a mistake on our part, especially because we are not anymore B2B. We are more B2C. When you're doing B2C you should really have a free version of a product and rather incentivize people to go over their version. So we are going moreover to mass market and high volume rather than low volume high margin. It is not like a complete shift, but it is some shift in that direction.

**Interviewer:**

And I guess you said this first you had J query and then you changed for the front end?

**Interviewee:**

To React to react.

**Interviewer:**

And then changing so much the technology, because I know both technologies it's a way of transitioning, but it's also a way of throwing away a lot of code. So that might be very hard to transform Ja query to react without having to throw away all the code. So yeah, all the technical debt you have accumulated in Jake's query, then you realize, okay, because also a technological choice, it's a technical debt at the end of the day. So yeah, it makes you throw away and I'm asking more questions rather than an answer. What I'm trying to ask is that you did change this because you felt it was better for the customers to have it in React at the end of the day this kind of B2C model that you transformed, you felt, okay, maybe this customer working, were feeling better to have a product, which is based on React because it provides more features like you say, moving around?

**Interviewee:**

The pathway to React was that most of the more easily changeable part of the code was thrown away. And we started from scratch. This timeline method this section you see right here this was a part that was J for a long time. This was just J code that we implemented inside our React app and it worked just fine that way for quite a long time. But we have changed that right now to be pure react code. So we do it gradually, but, just a small part that we didn't change right away because it works.

**Interviewer:**

Maybe better.

**Interviewee:**

But of course in the long run, what your question was, is React better for customers than [inaudible]. it's much easier to scale a project with react than [inaudible]. So we thought that when we have a prototype and we thought that this could be a business, we chose whatever framework we thought was best in the long run, but doing [inaudible] was great to create some kind of prototype, because that was the language that I knew at that time. So I used what I knew. But I do think in terms of scalability React was the right choice?

**Interviewer:**

No, I understand. And value capture, it's more like your startup has changed the way method to capture value monetize. Have you had this kind of situation like it is somehow, yes, because you changed from B2B, B2C? Yeah. But then again how do you feel again, if we call this transition shifting from one technology to the other and accumulating this initial technical debt because you have changed technology anyway, do you feel that has an impact on how you monetize the product today? It's hard to connect them but.

**Interviewee:**

Yeah. We have done it with our value capture process, we'd be the same if we didn't move to React for example, or have it changed some, some big thing in the app in order to create better value for us/ I think actually that the reason we have changed these things is more rigged towards our developer experience rather than the value capture because the developer experience is better in a large scale, but in the long run, again, you can push with our technology stack right now, we can push new features a lot faster. It is much easier to change things around so we can push value towards the user faster right now than we would have been able to do before. I think you can structure your code just fine as you do your React code. But I think it is much easier to start messing around with some spaghetti madness rather than React. Yeah. Yeah, I agree.

**Interviewer:**

But the bottom line to me is there are three more of them, but when I look at the last one technology peanut, I think that's the one that applies the most like your startup has delivered the same solution by using completely different technology. And I think we can reason on this a bit more like JQuery is a good technology at first, but then you realized a lot of features cannot be implemented easily scaled and implemented in JQuery. So this is falling out to me, but you have to say your own opinion to me falls out toward the technical debt issue because you are making things because you know them but maybe that's not the right software engineering solution at the end of the day that you should have chosen from a technology perspective. So we try to connect the dots. It's not easy. I know it's at least there are 10 points that help out. And when I look at your solution and your startup, I summarize in my mind that you have made the technological shift and somehow that's also connected to technical depth because anyways you couldn't implement anything, but you have to say on that, you have a thing on that.

**Interviewee:**

Yeah. I think we would have a lot more technical debt if we did not shift the technology towards React and Graph Q and node. It's so much easier to change things around now. It's such a more flexible stack. And another thing that was important for us in our technology shows that is web-based is that you know, I can use electron or similar types of apps that you can create native apps on desktop Mac via phone by using web technology. And that was of course important to us. We didn't want to have separate iOS people in terms of Android people when we create mobile apps and the same with the Linux-based system or windows based systems or Mac or whatever. Yeah.

**Interviewer:**

Maybe a tip on this. I've been doing mobile app and cloud app development myself, and also tried a lot of consultancy before this with many projects. And I've been using Cordova a lot, [inaudible] and Cordova. Like you can transform your web app into a mobile application and Cordova. I can write it down in the chat. So Cordova and Phone gap. These two are quite easy. It'll help you a lot in transforming your web app into any other app, any other cross-platform conversion. So you can transform it to Android IOs, windows up, whatever, but it's not so much coding effort. It is done effortlessly. Yeah. Yeah. Almost effortless because you don't need teams for, IOs for Android, for Windows phones, et cetera. But you just develop a web application, you transform it into the hybrid version of the mobile app. So I've been into this doing this kind of thing for many years as in the Norwegian markets. And I understand that a lot of companies, don't want to do native. They want, sometimes they want to, especially an app, which is very web-based. So this is just a tip

**Interviewee:**

Thank you.

**Interviewer:**

The very last question is about how do you explain the role of technical debt in your startup pivoting scenarios in one or two sentences? So how do you feel like we made a discussion now almost one hour I hope you have understood something from this discussion, at least from the technical debt and pivoting perspective something maybe you haven't thought before, but how can you summarize in a couple of sentences?

**Interviewee:**

I think for our product changes will happen no matter what, and this technical debt will happen. So our code needs to be changeable as well. And that is our reasoning for pivoting towards a different technology than we had before.

**Interviewer:**

You are stating what I was expecting, so thanks. Any other questions you have for me because I'm very happy with the interview. I hope I've made good use of your time as well, and I thank you. But do you have any other questions or something you want to bring up

**Interviewee:**

It's just the focus of this interview is quite interesting because it's so different because it is focused. The pivot part is connected to the technical debt part. And for me, when I think about pivoting, it is a lot. I base it in terms of what the market says about the product. So if our market has given us feedback that a part of our apps should be just this and not all this other stuff, we, of course, will pivot that way, and that will create technical debt because we had a lot of things that we need to change. But when you asked the questions, I was kind of forced to think, from a program perspective, and a development perspective. So it was quite a different perspective from it.

**Interviewer:**

Yeah. I didn't force you, but because you said I was forced to think.

**Interviewee:**

It was more like I was moved in that direction by the questions.

**Interviewer:**

Yeah. Yeah. It sounds like I forced you to think about it. That's funny, but I'd like to thank you immensely for your time. I would like to say that it's a good idea to keep in touch because I work with this customer-driven course and of course, it's a very good course. I published some recent papers, but I want to bring in the customer-driven, new ideas, new startups, eventually, and companies that are doing new things, not just the old fashion companies, like Syntek ocean, or some other stuff. Yeah. I just have to think about it now, but all these companies have been coming to customer-driven for the past five, 10 years, and they ask the same, I wouldn't say stupid projects, but a little bit like niche projects that are maybe not so useful in modernizing the demand for the students.

**Interviewee:**

I also think you have the same age group as the students. So then you have a better communication process and you have fewer gaps in the communication. And I work closely with the course leader. So I'll talk to her and I'll explain about this interview, your product. And she's very enthusiastic about bringing startups as well this year. Also, something I want to add is the fact that I want to introduce things such as innovation concepts with camping do in the course so students can have innovative ideas, not just code for you, but also bring some innovative feedback to what you are trying to do, because it's not just a matter to get a small feature development feature done. It's also a matter of having done doing the right feature for the right customers. So this is my feeling, at least this is what companies should get on the course. And of course, this research goes back to the university context where I run this, as I said approach of teaching software engineering in interdisciplinary groups and groups that have some gender diversity a bit more than what you have right now, but it's important that they learn software engineering practices by doing and by creating startups. So it's quite nice that I hear you have done your own startup after you finished your own master thesis. And, you have gotten so far with what you are doing. And I think this, this project is quite interesting and it will have its own, path, in the market. So it'll find its own position in maybe a few years because it's relevant and it needs some integration from my perspective, but, of course, people have different perspectives. Maybe you got certified in, scrum master role or [inaudible]. So if you did that, then you understand the need, this is my opinion. So you really understand the need for what is going on. So anything you want to add, besides me talking too much

**Interviewer:**

And no, you're right. I have six certifications in a scrum right now and also just send me information about the student products. And we can just follow up on whatever

**Interviewee:**

I'll tell you when the application process starts, maybe pretty soon. I'm not sure I'll ask, and then I will make a part of that process. So apart from that, I would like really like to thank you. I think I took more time than you were expecting today for the interview. So I really like to thank you for all the answers and the very good communication process.

**Interviewer:**

Yeah. Thanks as well. It was interesting. Yeah.

**Interviewee:**

Please send the sign form back, whatever you feel comfortable about that, but not after two weeks, because anyways, I need to get it in my drive. So I have this done. Yeah. Thanks a lot. We'll talk soon, then. Just keep in touch.

**Interviewer:**

Bye.