

## Interview transcription - 2021-04-19- 17:00 over Zoom (remote)

Speaker 0 = Interviewee

Speaker 1 = interviewer

- Speaker 1 00:00:04 Yes. Okay, great. Uh, thank you, Eric, for participating in our study. As I said before, this conversation will be recorded and audio will later be transcribed. We will remove any names, uh, companies or partners, or where you're currently working, uh, due to confidentiality. So you don't need to hinder yourself in terms of what you're speaking about. And of course you will be, uh, getting the transcription afterwards. So you're able to revise anything later on as well. Um, and the interview will take approximately 45 minutes. So, um, our research is to see how, uh, teams and team members are using user story splitting and how it is conducted in their teams. And, uh, the goal of the interview is to get insight into how you and your team work with user stories, splitting and why you do it as you do it. And, uh, yeah, we will begin with some general questions about you and what you're doing right now. And then we will introduce the topic a little bit more on some definitions, and then we will move on to the questions regarding how you and your team would work with user stories. So, um, a little bit of an introduction of what are you, uh, what are your current, uh, role and what are you working with at the moment?
- Speaker 0 00:01:47 Uh, I'm a backend and dev ops engineer. Uh, I'm currently working on a system for, uh, tracking education for, uh, surgery, surgeries, uh, uh, I'm uh, yeah, I'm the only one working on the backend. And, uh, what'd you call it operations part, like, uh, deploying things to the cloud and stuff. Um, yeah. Uh, I, I'm not sure how much detail you.
- Speaker 1 00:02:26 Yeah. A little bit about, I didn't ask you this, so I need to correct myself if you would tell me a little bit about your experience since graduating, perhaps.
- Speaker 0 00:02:39 Oh yeah. Uh, I graduated, uh, three years ago. Uh, and I've been doing this kind of work since then, uh, at two different companies. Uh, the, the processes have been, uh, quite different, I think, uh, before it was very chaotic. Uh, my first job now it's more structured and say more, more thought through, uh, not just, uh, extinguishing fires because everything is falling apart.
- Speaker 1 00:03:22 Yeah. All right. Great. So yes, introduction to the topic. Um, so just for, for your understanding through the rest of the interview, when, uh, I will talk about user stories. Uh, I, we will specifically mean we, we won't specifically mean user story as this format that you might know more of as a user, I would like to, for some benefit, but rather any kind of work item, uh, that could be a Epic or a feature or a task, for example, or a requirement that you need, uh, that you often break down into smaller tasks or smaller features, for example. So that's also what we mean with use the story, splitting the practice of breaking down any kind of job or job item ticket or work at. Alright. Yes. So, and, um, if we usually within a team or, or a project or a company, uh, it's uh, often you some kind of hierarchy for, uh, structuring the user stories or work items, for example, often connected to some kind of work board. And that could be epics are broken down into features and broken down into user stories, for example. So in

terms of, uh, your team and what the, perhaps the project that you're currently working on, how would you describe that hero key at the moment?

- Speaker 0 00:05:15 Uh, we mostly use two tiers in the hierarchy, uh, one for epics and one for, uh, it's either tasks or features that, uh, we've tried a bit with like tasks, but we generally just go with features, uh, because we did it, it was just too much work, uh, with, uh, having a third level of like splitting it up even more. Yeah.
- Speaker 1 00:05:52 Okay. So usually epics then features, but sometimes tasks on the board. Yeah. Okay. Okay. And if you, if you would describe how you do use the story splitting in your team, how would you describe that process?
- Speaker 0 00:06:23 Uh, mostly it's just a meeting. Uh, it's like, uh, when we feel like this is too much, uh, to be completed, then say in a week, we try to like remove criteria or like move them out of scope, uh, and create a new ticket for that instead. Uh, then we also split when we actually work, we split it in a different way because some, one person is doing the front end, like a React app, and I'm doing the node JS servers, but that's not really, that's a very implicit split because it's not really, that's the part where we used to create tickets for, or like tasks on every feature, uh, where some tasks where like this needs to happen on the backend and this things needs to happen on the front end, but we've pretty much stopped that. So now you can have a ticket, that's a, it's a full feature, but somehow implicitly we're dividing the tasks into this is what the, this is the responsibility of the backend, and this is the responsibility of the front end, but it's not, it's not documented in any way. Uh,
- Speaker 1 00:08:01 So would that, which you are describing there, would that be on feature level or Epic level?
- Speaker 0 00:08:09 I would say that we split features into tasks. Uh, but that's, yeah. Uh, it's completely, I would say it's not tracked in the process that we do it. Uh, so it's just that the, yeah. Uh, the developers are specialized specialized in either part of the system and, uh, yeah, we just talk it through and I'll, I'll create an API like this, or I need an IPA like this and we'll yeah, just do it on the fly. It's actually, it is some sort of user story splitting. It's just that don't really, uh, have it it's, it's not in documentation, it's just either through, in our meetings or some chat.
- Speaker 1 00:09:07 Okay. So for example, a feature, let's say a feature at the moment, uh, you will discuss this at this meeting that you speak of, and then the front end person knows about this ticket, what they should do. And you as a backend developer knows what you should do.
- Speaker 0 00:09:29 Yeah. Uh, pretty much we have, we have like regular meetings every week, uh, where yeah. Everything is discussed. Uh, but there are also, yeah. Uh, why have you worked? There are lots of questions coming up and how you need to redefine stuff, things like that. Uh, so it's, yeah, it's, it's happening all the time, sort of.
- Speaker 1 00:10:03 Okay. So that sort of leads me into the next question. Uh, which is when do you usually do user story splitting? So it sounds like it is both in this meeting, perhaps, and then at, when it's necessary, perhaps.
- Speaker 0 00:10:22 Yes. Uh, I would say so because we have a, uh, I can't remember what we call them meetings, but we have like one sprint planning, uh, and

then, uh, where we like decide what to do during a sprint. Uh, and then we have, uh, that's every two weeks. And then, uh, at the same time, the weeks we don't have sprint planning, we have some sort of, pre-planning where we have a more technical discussion about each of the tickets and what they actually entail. Uh, and I'd say the sprint planning is where everything is like decided. Uh, but then, uh, everyone, uh, has a quite good understanding of the entire feature or what the feature entails. Uh, most of the time it's quite clear, uh, who needs to do what, but, uh, sometimes it's like, uh, this edge case needs to be fixed. And then, yeah, it's just on demand when someone realized this, uh, the current course isn't really working, so we need to fix it, do something else.

- Speaker 1 00:11:44 Okay. Okay. So, uh, just for my understanding, um, would that mean if someone from the outside were looking at your tickets, uh, without any specific technical knowledge or insight, they could not see, for example, a task, which says, um, make the API and backend make API and font and make this button or something?
- Speaker 0 00:12:16 No, uh, uh, that's not how the tickets is specified. Usually it's more like, uh, more, uh, higher, you would need a higher level of understanding of the project to understand most tickets because, uh, it's lots of very product specific things like, uh, and the language that is used, uh, but really it should be right down acceptance criteria, but like, yeah, this information needs to be shown this needs to happen when you click this button or, uh, stuff like that. But it, uh, tickets rarely describe, uh, where in the system, things like this, if I should I say, uh, uh, most tickets involved back in work, uh, but none of the tickets actually says, uh, mentions anything about the or backend services or stuff like that. Um, so the tickets are described like what the user would want to see or how the system should behave, uh, on there. Um, so yeah. Uh, I'm not sure if I answered your question.
- Speaker 1 00:13:54 Uh, I think it's especially the last part to explain that a lot of perhaps how it's defined on how it's structured. So if you consider yourself, for example, with one of these tickets and, um, uh, implicit backend part that you mentioned, um, if there are like several things that you need to do in terms of the backend work for that ticket, do you, um, do you, do, do you structure that in some way, for example, with the check list or something?
- Speaker 0 00:14:37 Nope. It's just in my head
- Speaker 1 00:14:47 And, um, from, I don't want it to sound too much of a leading question, but if, if you in the team, if you were to backend engineers, you think it would be different in some way?
- Speaker 0 00:15:15 Nope. Uh, uh, there's not for my part because I think we would divide the features so that we each get their own feature, uh, at least because there are two front end developers and they usually solve it that way. So they just take a whole feature and, uh, work on them in parallel.
- Speaker 1 00:15:42 All right. All right. Um, so you, you mentioned a little bit about you as a backend engineer and then the two, uh, front-end persons involved. Is there, is there any anyone else or any role else involved in when you do use a story splitting, for example, at the spring planning?

- Speaker 0 00:16:06 Uh, Oh yeah. Our project leader is his, uh, yeah, [PROJECT LEADER NAME], he knows about most things might not have the like exact technical details, but he, of the, like, he grasps most of the moreover on a slightly higher level with email, he reads most commits and stuff like that. Uh, so he's part of it. Uh, yeah. And then we have a designer. Uh she's but I wouldn't say she's part of the user story splitting because she mostly she's preparing stuff that will be developed like a week ahead or something like that. Uh, I'm not sure how she's not like, um, let's see. Yeah, she's working one week ahead of us or, or one sprint ahead of us because she's like doing the designs for the things we're doing the following sprint. Uh, I guess sometimes we, she does too much design or sometimes she does too little design for like what we decide when we split up the story, but I'm not sure if doesn't really interfere with our way of working. Um, when we, yeah. When we implement it, we, we can, we can like request to science or ask her about things that are not, or like if she missed a part, like, yeah, we need a save button here. Uh, we just ask her, uh, about how, how it should be implemented, like fix it on demand or whatever.
- Speaker 2 00:18:19 Okay. Okay.
- Speaker 1 00:18:22 So, um, apart from, um, Um, let you, if we're you imagine an Epic, for example, that will be broken into features, um, what, what would you say that you take into account when you split this Epic? For example,
- Speaker 0 00:19:00 We try to make each like split, uh, contain something of value to the end user. Uh, so that yeah. So it's not like we yeah. Create the API end-point, uh, for a feature, but doesn't include the like front end, that standpoint, or the other way around to almost all features are like a, you call it slice through all components of the system. Um, I think that's the like main part and then try to make it so small that we can, it should never take more than a sprint, uh, for us to complete. Um, I think, yeah, I'd say that those are the main things.
- Speaker 1 00:20:08 Okay. So epics does not necessarily have to be implemented or delivered within the sprint, is that correct?
- Speaker 0 00:20:18 That is correct. Yes.
- Speaker 1 00:20:22 Um, okay. So you mentioned, You mentioned that you, you slice or split, uh, through the layers of the systems to perhaps keep the user value for each individual feature. Um, is that, is there any other reason than that, that you split in this way? For instance,
- Speaker 0 00:21:03 Oh, well, yeah, I think it's, uh, it's easier for us when we build, uh, the things, because if, uh, I'm building an API point at the same time as, uh, someone building like, uh, folks that say I build, uh, end points for a form input form, uh, if someone is building the input form at the same time, it's much easier to discuss like, uh, all the edge cases when, uh, the person is like, have the same, uh, things in their head, uh, at the same time, instead of like, I create API end points and then a month later someone is going to implement them and I have completely forget them, everything about those end points, uh, uh, and, uh, yeah, I'm busy with something else then it's much easier to like, contain everything in the same sprint.
- Speaker 1 00:22:24 Is there any specific, like guideline or technique for this, uh, this

way of user story splitting that you have derived this technique from?

- Speaker 0 00:22:37 Um, um, I'm not aware of anything. I'm not sure if, Uh, someone else has like read up about this, but, uh, I'm just like, yeah. Inventing things as we go.
- Speaker 1 00:23:04 Okay. So it's This, uh, not from a specific guideline or technique that you, at least that you know of, but is it,
- Speaker 0 00:23:19 Well, I guess we're, we usually do retrospectives, uh, like every month or so. Uh, and we have like, so most of our processes like derived from our retrospectives and like, yeah, this seems to be working. This seems not to be working. Uh, so it's like, yeah, it's a result of like the, if it's like, uh, eight months of work on doing retrospectives. Uh,
- Speaker 1 00:24:08 Okay. Is that eight months of doing retrospectives that have taken you to this point?
- Speaker 0 00:24:14 Yeah. Uh, or at least I started in this project eight months ago.
- Speaker 1 00:24:23 Um, is there, uh, during those eight months, have you, do you remember any point in time when you specifically change something about how you split?
- Speaker 0 00:24:49 Uh, no. I can't really remember any exact details.
- Speaker 1 00:24:57 Um, and, uh, re evaluation process. You've mentioned. Um, do you have any kind of a definition in place for when your story splitting is done correctly?
- Speaker 0 00:25:24 No. I think that's a lot of like, just feeling that all the developers involved are like, yeah, this seems like a reasonable amount of work. Um, we usually do some sort of like planning poker, uh, or everyone has to like, yeah, do one, two, three, and show a number of fingers, uh, to determine the size of each tickets. Um, so that, that's how we try to reach consensus about whether something is too big or too small, or, yeah.
- Speaker 1 00:26:13 So, uh, which is sort of my next question is when, perhaps it's not known out of a definition when it's done correctly, but at least when you come to the conclusion that something is too big, then it sounds like you decide that you need to split it.
- Speaker 0 00:26:35 Yeah. It's usually when, yeah. Uh, we're doing the like, uh, Fibonacci numbers yeah. For the poker. So if it reaches 13 or, uh, sometimes 13 and every time it's 21, uh, it will be split up. Uh, so that's usually, that's like, uh, yeah, it's not an exact science, but, uh, those sort of like the general guidelines. Uh,
- Speaker 1 00:27:15 Yes, of course. Um, so, um, if you were to find yourself in your colleagues, uh, shoes for a minute, perhaps the, the people at the front end site, uh, how, how would you, how would you think that they would consider how you split user stories, how it is done at the moment?
- Speaker 0 00:28:05 Uh, I have no idea how to answer that.
- Speaker 1 00:28:10 Uh, it was a very long sentence. So, um, Yeah, let's leave that at that. Um, in terms of more generally, um, you can consider this current project or perhaps your previous experience as well. Um, what, what do you think are the main benefits of splitting user stories or work items?
- Speaker 0 00:29:03 Looking from the customer perspective. I think it's good because

we continually deliver things. Uh, and it's like, we can show that we're continually making progress and not like, uh, yeah. And we can continually also verify that we're doing the right things, but just nice.

- Speaker 0 00:29:47 I previously we didn't really split stories we used. Um, but then we used more of a combined style of working and stuff just took this time. They took, uh, there was not no real, like, we didn't really track progress in the, in the form of tickets. Um, but then there were, but th that sort of created the problem that we didn't really know where the things were like, correct or not. Sometimes it took too long for them to like go through the steps of being written down and implemented and tested and verified, uh, delivered. Uh, so that was kind of, so I think it's, uh, but yeah, I think it's the, for me, the most values we continually like gets feedback, uh, from the, from the customer who are using the system that the that's, we're all on the right track, building that building what they want.
- Speaker 1 00:31:15 Yes. Great. That turns some valuable benefits for sure. Um, and of course, in relation to that, is there, is there any drawbacks that you can think of, of this process, of your story splitting, for example, splitting a larger task into a smaller one and focusing on the smaller ones rather than implementing the larger ones?
- Speaker 0 00:31:47 Yeah. I can see that, uh, like some architect show perspective is like missing because you just like continually add small pieces of code, but you'd never really consider the system as a whole. Um, and like how everything is connected. Uh, uh, yeah, I think that, yeah, I can really, uh, I have experienced several times where like, yeah, uh, I can't really do this ticket right now because they need to fix these things because they're all like, yeah. Uh, but at the same time, it's like, yeah, I'm not sure how to explain this. Um,
- Speaker 0 00:32:50 Yeah. It's like, you learn a lot about what you're about to build when you build it. Uh, so I'm not sure that the architectural decisions would have been better if someone, if they were like taken beforehand, because a lot of I'll change during like three months, uh, new features are invented. Uh, you, yeah. Uh, I guess, uh, I guess you could fix it by like having more, uh, longer free planning sometimes, I guess, or something. I, I'm not really sure how to fix it, but I feel like the architectural part there is suffering because you don't, there's never really time to do it. I think that that could be the problem. It's like one of those maintenance things you need to do sometimes, uh, reorganize, uh that's when you, you're always focusing on the, on delivering value, um, those, uh, those tasks like tend to be forgotten because yeah. To, to the customer, there is no real value of like yeah. Refactoring code, uh, or there is no direct value because you wouldn't add a new feature, but it's a very abstract kind of value where the next features will be delivered faster, but it's hard, I guess it's harder to grasp. Um,
- Speaker 1 00:34:42 But of course refactoring could be valuable, uh, for you as a developer.
- Speaker 0 00:34:54 Yeah. Yeah. If, if the customer wants the project to gone refactoring, it's valuable. Yeah. It's, it's just harder to sell, uh, a ticket that's just refactoring, uh, then like we're going to add feature a, uh, so yeah, from my point of view, the solution is usually like, yeah, you don't mention it to the customer, you just do it. Uh,

and some tasks just take away longer because of it.

- Speaker 1 00:35:35 Yeah. That's, that's a really good example and very interesting as well. And that's sort of what I've been thinking about during this interview with you specifically as a backend engineer in terms of this, which is fairly popular with vertical splitting is, um, you might sort of have given a way this answer, but if you would not do vertical splitting, but through our, uh, horizontally or by layer, um, uh, do you think, what, what do you think would be the, uh, the main differences between vertical and horizontal?
- Speaker 0 00:36:41 Uh, I think, uh, horizontal, like it creates extra work because, uh, even if you write, like, if I'm writing billing an endpoint or writing it, simple tests for that, I will always make, I will like always miss an edge case. Uh, so I will always have to go back and like fix some little detail, like, yeah, this should be a maybe string instead of a string or some like little details like that. Uh, and I think we kind of, those kind of little edge cases are discovered, uh, by the front end engineers. Uh, and I think it saves time when like you discover them much earlier. Uh, yeah, I guess that's what I answered before pretty much. Uh, but yeah.
- Speaker 1 00:37:54 Yeah. Great. But, uh, yeah, especially that last part about the architectural, uh, perhaps missing things in the big picture, uh, might result in a problem. So, um, we are pretty much done from our part, if, uh, does my colleague have any other question now? We're good to go there. Um, uh, Eric, is there anything that you have thought about during this interview that you think that might be interesting for us in this topic?
- Speaker 0 00:38:41 Well, I can't think of anything. All right, great. Now we can pause the recording.