Interview transcription - 2021-04-14- 14:00 over Zoom (remote)
Speaker 0 = interviewer
Speaker 1 = Interviewee

- Speaker 0 00:00:01 Yep. All right. Great. So first of all, did you say thank you for participating in our study? So as I mentioned, this conversation will be recorded and audio will later be transcribed. Uh, we will remove, and the company names or colleague names or project names for that matter that are mentioned due to the confidentiality aspect. And after the transcription has been done, which will be done by use of a software tool called rev AI, uh, we will forward the transcription to you and you will have a possibility to check it out if you would like something to be changed or excluded. Um, and of course that's, uh, totally up to you, whether or not you want to reply or not. Great. So on, um, our research is to see how a user story splitting is conducted within industry and the reasons, reasons for why it is conducted and perhaps what the impact is of that, um, that's practice.
- Speaker 0 00:01:23 So the goal of this interview is to get insight into how you and, uh, your, your current team or previous team, uh, uh, worked with user stories predicting. So the interview will take a maximum up to an hour. We usually usually takes 40 minutes or so. Um, and, uh, we would first asked you some general questions about you and your work-life experience. And then we will introduce the topic a little bit more, and then we will move on to the actual questions regarding the research. So, um, if you wouldn't mind, uh, you could tell me a little bit about yourself and your experience within your team, as well as your current role
- Speaker 1 00:02:18 Experience. What do you mean by what I'm doing? Right?
- Speaker 0 00:02:22 Yeah,
- Speaker 1 00:02:24 My previous, well, I started the tech career in, uh, in a company called [COMPANY NAME], basically where I was a .net developer, uh, junior one, right after the high, uh, high school, no, but, uh, university. And, um, just did C# coding for windows forms. Um, later on became a team lead and helped as well to, to move the project along. Afterwards, I, I jumped around from one work to one work to find a better place, uh, until I landed in [COMPANYNAME] I mean, at [COMPANY NAME] have been working for now almost four years, started as a .net developer as well in a team called claim intake. Um, basically we are responsible for private claims intake. If you have a broke, if you have damage at your car, you come to our webpages, fill it out.
- Speaker 1 00:03:21 And then, uh, you, afterwards has registered the claim with [COMPANY NAME] , and that was our domain. Uh, later on, I became a tech lead that is more of, uh, making sure the team's technical depth and technical knowledge been increased and, uh, and, uh, represent the team as, as from a technical view. From there on, uh, I become a team lead and, and as well as scrum master. Uh, what does it mean? Basically making sure that that team, uh, operates smoothly and has its own needs, uh, filled out, that, uh, means that basically, uh, the backlog needs to be in order. The processes needs to be in order. It was my job to make sure, sure that they are there, they are executed and as well, of course, uh, backlog, as you have asked me to have this interview. Any specific question?
- Speaker 0 00:04:19 No, I think that was a great then we have in general idea. Sure. You're working with, so, uh, regarding the topic, um, what we mean when we are asking about use the stories in this context, uh, is not specifically, uh, this user

story formats that you familiar with as a user, I would like to, for some reason, um, but any kind of eight and, uh, which could be an Epic feature task and so forth. So user story splitting in this context is also related to any kind of granularity.

- Speaker 1 00:05:02 Yep, that's how I treat, treat everything as well.
- Speaker 0 00:05:05 Yeah, exactly. Um, so, um, looking at that, uh, a, uh, say start point, how would you say that your team are currently working with a user story hierarchy, like Epic feature tasks or so,
- Speaker 1 00:05:28 So the epics come mostly from the business representatives of the business side. Uh, you get the big thing that we need to do. Uh, it can be whatever it's, it's a new claim intake. It's, it's some new system we need to build or integration, uh, that is the epic. Uh, second thing that afterwards we do, then our analysis of what it is, uh, and we figure out either split them in into features. And if there isn't any feature needed, then we can directly link, uh, create a user story. So when we talk about features, features in our team is more meant like a combination with, with, uh, somehow the, that one big Epic, uh, can be thought. So they're separated in some logical steps and bigger chunks. Um, and those are not the end product for developer team to do. They are more like combining them.
- Speaker 1 00:06:32 So after we have figure out what are our features, we then, uh, create the skeleton most likely for, for, uh, for user stories that will be the end and task for the user or it's, uh, it can be, uh, for, for user experience or tester or developer, it doesn't matter. But basically we create the skeleton and, uh, um, not always, we fill it in fully. We give the developers a challenge to figure out all of the stuff that is needed there as well to do them by themselves. So I'm not, does this answer your question?
- Speaker 0 00:07:10 Yes. Yes, it does. Um, but, um, so in terms of, uh, the user stories yep. Uh, do you ever break down or split those as well?
- Speaker 1 00:07:26 Uh, can you expand on this question please?
- Speaker 0 00:07:29 So from epics to features and then features can contain user stories, but how do you split?
- Speaker 1 00:07:37 No, we don't. We are working in Kanban, uh, manner, and there is no need for that. Uh, though we have a possibility to add, uh, so-called tasks. Uh, but those are more for, uh, the, the developer or the person who are UX person who will do that for themselves, so they can, uh, mark down what they need to do. But, but, but the unit of user story is one single, uh, thing that you can produce. That's a one whole thing. It cannot be how to say, uh, it's one thing then you can deploy or one thing then you will do, and that's the result end result of something.
- Speaker 0 00:08:21 All right. Great. Um, so in, in general, could you, could you describe how, for example, you work with splitting an Epic or a feature?
- Speaker 1 00:08:35 Yeah, but, um, for, for myself, uh, I'm most likely me and, uh, someone from business person or my product owner, uh, starts it off doing the baseline, afterwards when the baseline is done, we go with them to go through them together and see okay, does this makes logical sense, uh, is everything included in those features? So that's the first thing, figure out if the features contain everything and they are logical, um, from there on, um, it's good to start to, we, we, we try to figure out if all of the, all of the requirements are somewhere located or are we already to start work if we are, then we together or separately create a user story. It depends who at that point have the best knowledge of creating those user stories or features somehow like, something like that.

- Speaker 0 00:09:36 Yeah. Um, when would you say that you, uh, when did you do this user stories splitting, at what point in your workflow process?
- Speaker 1 00:09:53 Uh, at what point, um, it's in the research, I would say it's quite early in the first stages before, uh, I have, don't have an specific definition for that, but I would say when, when the research starts, or even before the research, if we have the user story, we have known the, what we need to do. The second part is basically split the features or split into the features. And then, uh, from those features, try to figure out what is missing and start to research and do UX work and architect work from there on after collecting requirements, but before doing the full, uh, full design of a product or Epic or end result.
- Speaker 0 00:10:49 Okay. Um, considering that you mentioned that you worked Kanban style, um, w w would that mean that for example, the epics, uh, uh, you, you start working with an Epic whenever it comes to you, um, are you working at all with some kind of ceremony, like a sprint planning?
- Speaker 1 00:11:15 No, we don't have any spring planning. That's, that's, that's the thing when we are done with the thing we are done, uh, you give the user story to developer, developer, do it for launch it to production, and then it's done. And when we, and we just, uh, add more and more, uh, uh, work items into the column. So everyone has, and when you are done with the first big thing, you just take the next epic, prioritize that epic and start to work with that one. Uh, of course, prior to that you have to put plan in yourself that, Hey, developers are currently doing Epic one. They are halfway through, they will soon need to start working, uh, additionally, and they will soon will need additional work. So you have to start work on Epic two already, the UX, the architectural, the business decisions, and make sure that the other teams are involved as well.
- Speaker 0 00:12:06 But, um, uh, so in terms of this, um, for example, features being connected to an Epic, um, and it sounds like an Epic sort of have some defined, uh, business value or that it comes from the business side. Um, other tasks, for example, I don't know, it could be, uh, bugs or re-factoring work or some kind of more technical performance issue, or those linked together as epics as well, or?
- Speaker 1 00:12:43 Yes. Uh, for us, uh, I've created this separate Epic. Um, we have done it more in different ways currently. Uh, previously we had one Epic each year for, uh, business, um, technical depth that everything that is recorded comes from a business side that isn't a part of Epic, or we find out that we need to change something. We just store it in that Epic. Then there is one for technical debt. Um, basically that's an Epic that stores all the code related stuff. Uh, and we have tests test, uh, technical debt. That means that there are some unit tests, uh, automation tests that we are lacking or, or, and all that information is stored underneath those epics. Uh, currently we changed a little bit up. We split, I had a quite big team, uh, so we split them and split them up in two separate ones. And now we have an Epic for each separate team and in our, Oh, I probably should mention that, but you already know we have those super sprints, uh, that are company based. And then those, the idea of those epics that will be specific for each team that are technical, whatever that will, will be stored for, for a super sprint one super sprint iteration.
- Speaker 0 00:14:10 Right. Right. So you mentioned a little bit about you being involved, for example, when an Epic comes in, uh, as well as the product owner, I believe
- Speaker 1 00:14:23 A little bit here and there,

- Speaker 0   00:14:26   Um, at these specific levels, um, who is usually involved in user story splitting role-based
- Speaker 1   00:14:37   Well, it depends from the user story to user story and the competence of the people. I would say it doesn't depend on, on one particular role more than who can do it better. Uh, either if it's more of a business related question, then I go to my PO who knows better. If it's a technical question, then, then it's maybe me or architect, if it's a UX based, uh, thing, then the user experience a representative or person can do it way better than, than us. So it lies in the competencies rather than the roles.
- Speaker 0   00:15:14   And would you say that you are responsible for identifying what roles needs to be involved
- Speaker 1   00:15:23   To an extent yes. Uh, uh, well, uh, my team is built in such a way, or at least I tried that, uh, we work as a one separate as one unit. So if someone doesn't know what each and every member can kind of act by themselves, if they feel so, but yes, uh, I am one of, one of the main persons that should, can help to figure out which person should be involved.
- Speaker 0   00:15:49   Okay. But a for example, a UX person, or able to address that this feature or user story might need UX work as well. Yes.
- Speaker 1   00:16:03   Uh, to extent again, uh, there comes an epics which allows, uh,UX to be involved quite, quite early, and then there is epics that are prepared for and get, get the UX involved quite in time. So it depends, for example, if we are currently building a new system, a called [PROJECT NAME] or a building, and there, the UX is one of the main drivers that is doing those, those, um, and user splitting and figuring out what is needed next as well with architect. But when, when it comes to more of a claim, uh, intake related epics, then there is more of a PO involvement rather than UX or someone else.
- Speaker 0   00:16:59   All right. You mentioned that, for example, um, you might, uh, you might involve, uh, different roles depending on what areas and a user story, for example involves. And you also mentioned that it sounded like you did on user story level, it is split for example, vertically. So you are able to deliver a value per each user story independent of each other.
- Speaker 1   00:17:20   Yes. Yes.
- Speaker 0   00:17:25   So, is it anything else that you take account when you split the user story apart from expertise and perhaps this vertical lining?
- Speaker 1   00:17:58   None that it cannot, it's hard to say, no, I can't figure out anything currently on the spot. So I would say no, at least.
- Speaker 0   00:18:15   It's a very general question so that, um, but okay. I, are you, do you know if you're using any specific guidelines or techniques for a user story splitting?
- Speaker 1   00:18:30   No, we are not. We, I know there is a name for, for using, uh, uh, splitting up epics in a more sensible chunks that are represented by, by some feature or level it's called somehow. But I don't recall the name for that specific methodology.
- Speaker 0   00:18:51   So that's an outside methodology to do think about it. It's not their company specific
- Speaker 1   00:18:58   To an extent. Yes. It's more of a, how it is, this methodology makes sense. And so you, um, kind of when you work with, with the, with the, with it just make sense and you don't actually need to think about it. It's basically telling you

to split everything in sensible chunks that make sense or are representative of some value for the customer or for the solution. But I don't remember the right name, but no, we don't have any company internal guidelines, how to do it,

- Speaker 0 00:19:33 Have you yourself developed any guidelines that are usable?
- Speaker 1 00:19:43 No. Basically it's on, on your common sense because each of us, each of every Epic can be a little bit different and you need to figure out what the, value you are bringing, with the Epic, then, then split it up in sensible chunks. What, what are the features? How can you separate them? Maybe, um, one method is to do the user journey, uh, split the features and user journey, uh, kind of manner. What will the user do, or if it's a technical thing, split them up and, and technical parts or, or, or chunks of code or what you can deliver separately. So it, it depends from the one the Epic actually is.
- Speaker 0 00:20:32 All right. Do, do you have any, um, do you have any definition of when you see a story splitting or a splitting of any kind of level is done correctly?
- Speaker 1 00:20:48 No. Um, we don't even tend to make such a, uh, presumptions, uh, when we start basically on, when the user, when we provide the user stories to developers, we go through them and the developers can tell us, yes, there's everything. I think there is everything or no, there isn't, and then comes in the agile, uh, the agile operating model tells us that we should be able to do to have, uh, a fast response and, uh, able to change our minds in the middle of our delivery process. So at one point we show everything that the team or to the, that we are starting, this project, is this okay? They say, yes, we move forward. If something comes up, then, then, then we split again and again, and again, as much as needed until we deliver the value that is needed for the customer.
- Speaker 0 00:21:45 Great. And, uh, on that subject, uh, maybe not a definition, but do you remember a time when a split has not been done correctly
- Speaker 1 00:22:02 Multiple times, uh, or, or for example, recently we had an, uh, a new area coming in that I saw, and then we split it up, uh, uh, technical ways that these all questions should go under one component. And when we showed it to the team, the team state, they stated, Hey, this is not okay. And we have to go through another review. So, so it was given to you X to go through those questions and figure out either those requirements can be changed up and split in somehow differently. And so it, it went through another review quite recently.
- Speaker 0 00:22:44 And would you say that's, uh, that's sort of necessary step or is it something that you could have done to mitigate that extra step?
- Speaker 1 00:22:56 Well, it depends from the people, uh, yes, if you have, uh, if you have a super competent people who know the areas and all those, then you can solve it, but always something will slip up and you have to try to mitigate it as much as you can, but you have to expect still that those things will happen. Well. Uh, if you, any specific, uh, things to do, yeah. Do more, involve all people in much earlier phases, and that can solve a lot of problems in the future when you show them the user stories.
- Speaker 0 00:23:39 And in terms of this, do you ever evaluate the process of your story splitting?
- Speaker 1 00:23:48 Um, not quite. We have all our all working processes. Their stories are just the two, how we work. And then when we evaluate or figure out how to change our day-to-day work, it's all as well somehow affects our user story splitting

and so on. Uh, for example, um, again, coming back to the UX, uh, part we currently in architectural part, we are currently understand that, Hey, sometimes we tend to forget to involve them. So we, we try to create or split, uh, one part that is just for research that we didn't do before, just so we remember and see that, Oh, okay. They're architects have checked this, the user UX have checked this and we can move forward with it. So we don't end up in the situation where we're, we show it to the team and team is not happy with how it's split to them, what is written in their user stories.

- Speaker 0 00:24:48 Okay. And do, do, do, do you keep that in your hand or do you have some kind of checklist not to,
- Speaker 1 00:24:58 So it's a good question. Uh, I think there is no checklist. We have at least, uh, [PO NAME] only, or also the PO has created, uh, a process of, or workflow, how it should go, like when it comes, who, who, in which stages, which people should be involved. So it's not the checklist. It's more of an involvement in product development stages.
- Speaker 0 00:25:25 Okay, great. And, um, yeah. Um, what do you think the main benefits of split thing, user stories or work items are main benefits?
- Speaker 1 00:25:44 Main benefits? It's a again. How do you use the tool? It's just a tool. It doesn't give you any volume. If you can't find one, for example, for me, uh, how we tend to use it, this is to make a clearer vision and easier understanding for, for the people who is working with the thing we are building. If the epic is super huge, and we will give it, give it as such to the, to the people who need to do execute it. We will end up in a big mess, cause most likely half of it will be missed. Half of it will be done incorrectly. If we split them into smaller chunks, they become more understandable or we can overview them better and we can analyze them better. So we can do our job better as well. But then again, uh, but it's just the tool. If you do the splitting wrong, or you can end up in the other direction, you, Hey, we did it wrong. And we ended up in the other end of the spectrum where, where the, the work became harder and then the development become much messier and we delivered something bad that we shouldn't do.
- Speaker 0 00:27:03 Great. Um, is there any, is there any drawbacks of this process that you can think of?
- Speaker 1 00:27:15 Hmm, drawbacks, Hmm.
- Speaker 0 00:27:18 In relation to perhaps working with larger, uh, work items?
- Speaker 1 00:27:27 Well, I currently can't think of, if I put thing together, for example, I'm putting this methodology with waterfall style style of working when you define everything and then give it to the, to the next team to do development. I see only benefits. Uh, so I, I can't imagine any better way to present the job currently. So maybe there are drawbacks that I don't know any better solution that could solve some issues that I don't don't think at the current moment. Yeah. I hope you understand what I'm trying to mumble.
- Speaker 0 00:28:10 Yeah. Um, so, um, if you, lastly, if you think about, uh, what your colleagues would say to say that you express or the product owner or a developer, what, what do you think they would say about how your process of splitting user stories work?
- Speaker 1 00:28:37 Mm, well, it depends from the people to people. Uh, UX would say, Hey, we need it. Doesn't work. We need the need to be involved faster. Um, business representatives would say most likely that they cannot do it fully

without other people, they need help from different persons. Um, just to make sure that the user story are actually, uh, done correctly because they can, uh, for example, from business side, split them up on deal, a point where there comes in a technical stuff you need to do, then, then it needs other involvement. What else could they say? Um, the developers, uh, not sure maybe they are, they can tell that we are not providing them enough information from time to time. Um, but that's how we are, um, making sure doing our work, we allow them to figure out things as well, but it's, and, uh, from product owner perspective, uh, what could be, what could they say? Yeah. They don't know the full process maybe, and it more needs to be more, but the overall, basically everyone would say, we need a better overview of how it, how things are doing done. That, that the main thing that they, the people could say better overview of how things are done. Long story.

- Speaker 0 00:30:22 Um, so is there, is there anything related to this topic that you have thought about that we maybe had not discussed?
- Speaker 1 00:30:35 I, no. No. Um, yeah. When you talk about user story splitting as such, uh, for me personally, it depends from what framework you use and how you work with it. Uh, uh, in, in, in, in, uh, theory, the scrum should have it more granular than, than Kanban and such. And then for example, you, how you create the user story, how you create a task for users to read how you work with them in the sprint. But that's that I think that, that, I think maybe it's not important currently, but yeah, that's, that's one thing to consider when you're thinking about thinking about this topic,
- Speaker 0 00:31:20 Have you ever, have you observed this difference in your previous work life experience?
- Speaker 1 00:31:26 Um, if you open up, for example, a Microsoft dev ops board that helps you all to organize it. Yes. It depends. Um, for example, how, what, what is the structure for, for splitting them up? If, if the Kanban, uh, can be epics user stories or epics users to reach features, or you can put whatever in between them then, then feed. And then in scrum, it's more of Epic features, user stories, and then there's tasks that you do. And then how, and that, that comes really closely on deployments as well. Uh, if, if, uh, for example, in Kanban, the user stories split the data in a, in a chunk that can be easily, for example, deployable, it can be, it has a blind line life of itself. Then, uh, scrum, you have a sprint and you tend to deploy everything at the end of the, uh, the end of your cycle. So you can split the user stories up in different manner. So that the two user stories needs to be deployed together and they have the one load and they combined into one chunk of one deployable thing. Yes. I hope you understood what I'm trying to say.
- Speaker 0 00:32:56 Yes. I was also thinking about, uh, the aspect that you mentioned in your term that developers might, um, sort of make a list of tasks of their own, just to know what, what they are doing for this use story is there and correct me if I'm wrong, but I'm assuming that this might not be on the board or
- Speaker 1 00:33:22 No, it is, you can add, we have enabled that feature, but we are not actively using it. And then some have some, some users at some doesn't it's for their own. And for example, one, one, one task could be, don't forget to add the configuration. Don't forget to write to this person on, don't forget, then those are kind of reminders. The things that you can do. In theory, if you would create from those user stories, uh, then the, those each separate user stories could not be deployed separately. And maybe that would be a good way to do in scrum, but in Kanban,

when we, when our team, when we try to deploy as fast as possible, we have to make, make sure that they user stories deployable and it comes as one. Okay.

- Speaker 0   00:34:20   And from your point of view, if you were to look at the user story that developers were working with, and let's say that developer is not using any of these tasks, that person doesn't work like that. And do you feel that you might be missing out on some overview of what the current status is of that?
- Speaker 1   00:34:46   No. That because we have boards as well, and, and I, and that comes into the different topic as well. How you work with the team, how do you trust your team? Do you think they are competent enough to have those decisions? You'll have given them the job they have asked the information you have provided as much as you can you have. I have given them the context, if you don't know where to ask for more information and they can do the job, I don't need to be involved in there anymore. I just need to know either it's in testing or it's done and that I can easily see from, from the boards columns.
- Speaker 0   00:35:29   I think that's, uh, what's a good wrap up. Great. Uh, down. I just want to say again, thank you very much for the interview. Welcome. I hope it will help you. Yeah, for sure. It was very beneficial.
-