

Interview transcription - 2021-04-14 - 18:00 over Zoom (remote)

Speaker 0 = interviewer

Speaker 1 = Interviewee

- Speaker 0 00:00:00 Uh, all right. So thank you for participating in spending your time. Um, so basically we will record this meeting and we will also transcribe it and of course remove any sensitive information such as names and the company names, et cetera. So feel free to speak freely without having to think too much about, uh, we, uh, kind of data. Um, so basically what we will discuss is the concept of, uh, user stories and user story splitting. And, uh, basically in terms of user stories, this is a concept or a term which might be, uh, heavily related to a specific syntax, such as a user or a different role. I would like to do something to get some benefits. in terms of this discussion and our thesis. Uh, we are not as strict in terms of that syntax. So basically when we mentioned user stories and what we would like to have you think, when you think about it is any kind of item. So either it could be an Epic, it could be a, of course, a feature or a task, or of course, a user story itself. And then the concept which we are trying to, uh, look into is what influences the splitting of such an item. So how come you do it this way, or how come you don't do it in a different way or where your influences are, et cetera. So in terms of that practice of dividing bigger things into smaller things.
- Speaker 0 00:01:54 So in order to get some demographics about you, uh, would you mind telling us a little bit about your experience within the field and also what you do now in your company or as your role?
- Speaker 1 00:02:08 Yeah, for sure. Um, so I studied, um, computer science as a bachelor and, um, interaction design as a master. Um, so UX design kind of, um, I did the bachelor in Holland where I'm from, and I've worked there for a year, uh, as a freelancer, uh, in various roles, both like, uh, by myself, just individual projects for clients. Um, but also in teams of two or three. And I got to Sweden, which was four ish years ago, I think, uh, did the masters program and have worked since then, uh, as a UX designer, as a developer, uh, focusing on front and focusing on web, but even doing some backend stuff and some iOS stuff. Um, and then recently since about a year, maybe I started taking on some project management like roles. Um, it's not a formal title that I carry, but more of a position within certain projects.
- Speaker 1 00:03:13 Um, I work as a consultant, so I I'm always involved with typically two, sometimes three or four in various projects, I'm, uh, mostly a developer and various projects. I'm a UX designer on. And then in one of the projects that I've been working with a long time now, I don't know, maybe like 30% capacity or full-time, um, for a year or year and a half, I've been doing kind of project management slash scrum masters slash proxy product owner. So I'm not the product owner myself are our clients. The client that we work for is that, um, but I work a lot with translating their desires and requirements and wishes to, well, both the technical implications, um, design implication. So I involve the right people at the right time and try to capture the right amount of information, um, and disseminate that that's, that's kind of what I do.
- Speaker 0 00:04:16 Okay. So, uh, in terms of your, maybe your present projects,

because that, that's my easiest to relate to when, uh, when you think about how the items or, uh, user stories or features, et cetera, are structured within your projects. Is there a hierarchy of certain, uh, tasks or these items in terms of what's the biggest ones and then how is the hierarchy constructed or is there a hierarchy?

- Speaker 1 00:04:48 Okay, so I'm not sure if it's necessarily a hierarchy, but it's one of the categorization methods that we have is the classic, uh, pointing or estimation. Um, according to story points, complexity, uh, one, two, three, five, eight or 13 points. We typically cap it at a max of 13 points. If a, if a story is bigger than 13 complexity points, we try to break it down further. Um, because we find that one person or one and a half person on one story can kind of max do a 13 pointer in one sprint. And if we have a bigger than that, then there's no chance of finishing it within the sprints. So in that case, we would try to break it down. Um, so that's, that's some sort of hierarchy, I guess, but it's not really applied to the whole backlog, it's filled with unspecified or an estimated, uh, points.
- Speaker 0 00:05:47 Okay. So in a bigger picture, you have X amount of unspecified items which then breaks down into, or are ones you reference in terms of a user story points size. Are those, are they always derived from something bigger?
- Speaker 1 00:06:10 Um, think thinking if I'm parsing your question correctly? I think so. Yes, because we do often have, uh, epics which described, but a very high level, a direction or a deliverable or a new big collection of features, um, which sometimes is described initially ahead of time, which we then break down into individual stories. Um, but that's, that's not always the way sometimes also we just start with a feature and then build, start with another feature on the note after a while the pattern of, Hey, these things belong together and belong to the same, um, group of related items, I guess you could call it an Epic. Uh, so we, we use it both in a top-down and bottom-up way. Okay.
- Speaker 0 00:07:01 Um, so when I, when I mentioned in the introduction about you user story splitting, is that something that you can describe, how it's, uh, from your experience, how that's treated or how it's done?
- Speaker 1 00:07:16 Uh, so our, our heuristic, there is, uh, 13 points max. And if we, during a backlog grooming or backlog refinement session, discover that a story turns out to be more than 13 points in complexity, then we have another go at it. Then we see where we can either simplify and remove a criteria on that ticket to make it simpler or through split it into. Um, so doing one part first, then another part second, um, or we do, um, so, so two to two things that we could do there. Sometimes we throw in a spike ticket, which is like, uh, uh, research ticket ahead of time. So in one sprint we might say, okay, uh, we'll take half a day and we'll research the various technical solutions to this, because right now we can't reasonably estimate this tickets or it's a really big ticket. Uh, so then we'll throw in, uh, uh, research ticket one sprint ahead and, uh, and look at that to see if we can break it down in some reasonable way. Or, and we sometimes do that as well. We throw in more of a task. Um, so our user stories, we always try, we don't always formulate it as a, uh, as a user I want to so that I..
- Speaker 1 00:08:37 Um, but we, we try to have the heuristic of every user story or every feature should have a value for the customer. Uh, so we can't really consider

something complete if it's just the backend implementation or if it's just the front end implementation that's that doesn't do anything. Um, but sometimes it's not really practical to split it into a smaller user story. In which case we might split it into a backend task on the front end task and start with one of them, um, mainly to reduce the, the necessary communication between different parties. Um, if we see that the backend has lots of space in the sprints, but the front end is really packed with other stuff and we might split it into a backend on the front end ticket so that we can progress on one, um, yeah.

Aspect.

- Speaker 0 00:09:32 Okay. So you mentioned that, uh, certain, uh, user stories have different complexity. Is there a way to describe what drives, uh, putting 13 points in comparison to eight, for example? So what decides the level of complexity?
- Speaker 1 00:09:47 Hmm. Um, ideally we want to compare that to other stories that we've done and think of other 13 pointers or eighth pointers. Um, so I think that's our, our goal to be able to compare stuff to previous work that we've done. Um, but I don't think that's always necessarily how it happens. Sometimes it just how a 13 pointer is a pretty overwhelming ticket where we don't see how the solution is to be done yet. Uh, an eight pointer is well-defined, but just a lot of work on a five or a three pointer or, or smaller things that we can finish in a couple of days. Um, so the differentiation between eight and 13, I think is mostly gut feeling. Do I do it, do we see how we can actually implement this or do we need to figure it out still, uh, races with higher tickets? So 13 points tickets are a lot more uncertain than, than eight points or five points or, or three points tickets. Okay.
- Speaker 0 00:10:59 And are those, uh, noun, just the winging the questions, but are those points, do you ever take into consideration who, uh, does that task, does that guide the complexity of, or the points given?
- Speaker 1 00:11:17 Uh, again, ideally no, of course not. in practice. Yes, it does. Um, we, we typically do a, uh, planning poker, uh, per story. Um, now in these times that's done by holding up your hands and, and, and indicating since we weren't all actually around the same table, um, for front-end tickets, uh, given that I know more about front end than back end and two in our team of four, our front enders. So we ended up with three front enders who can estimate together. And in that case, it's very easy to, uh, try not to think of whoever is going to be implemented, but to take, you know, the average or, or discuss, or, uh, but when it comes to backend tickets where, uh, the front ender is in our team on me and myself are not that good at estimating that's, uh, we let ourselves guide a little bit more by [NAME OF COLLEAGUE], for example, who was, who was more on the backend, um, on, and if he indicates that something is big, we just tend to believe that because he probably knows best. Um, so there's no point for me to, to, to sit there and argue. Um, and so in that case, it is a little bit more driven by the one person who will influence that.
- Speaker 0 00:12:36 So we're going back a little bit to, to user story splitting or the practice or the process, uh, who is generally involved when you do these kinds of discussions or these, uh, this process ?
- Speaker 1 00:12:51 That's, uh, the, our team of developers. Um, so three developers

and me, I know as a half developer or project monitor, um, then there's PO at the client. Who's always on call. Um, and sometimes the, uh, head of software development at our client. So PO boss, I believe it is. I'm not quite sure how the relationship with the client works, but there is another stakeholder from the client and are typically around six people. Okay.

- Speaker 0 00:13:23 Do you have any ideas about why that many roles or why those roles or persons specifically?
- Speaker 1 00:13:34 So the PO I think because it's a very valuable to have a short feedback loop there, um, we, as developers can identify away. Okay. So we can scale down on the complexity of this ticket by removing this, um, criteria, but we're not the ones who get to make that decision ultimately of is this an, a reasonable way of reducing complexity? Um, and sometimes it's okay. And sometimes it's, uh, it's really critical that we capture everything at once and then it becomes a bigger ticket and it's really useful to have the PO on call so that you can ask him right then and there, uh, reduces the amount of writing back and forth and, and meetings. And so there's a lot of people there, not because everybody is critical at every moment, but because it's just useful to, to get a lot of stuff done if everybody's there.
- Speaker 0 00:14:30 So if you were to put yourself in the shoes of a colleague, what do you think, or how do you think that they would describe the outcome of, um, or the results of user stories being split, so to speak. So do you think that they find it in a good way on why or do you think problematic and why? Of course.
- Speaker 1 00:14:55 I, I think my colleagues, um, appreciate things being split into smaller tasks, um, because it reduces the uncertainty for everybody. It makes it easier, um, for my colleagues to estimate or to, to sit down and, um, have an overview of what needs to be done, but it also, um, is a way of expectation management for, well, for me, I guess, um, for our, uh, product owner at the client on another cycle or not the clients so that they can see, okay, if we split it into smaller things, then it's more likely that we'll, we'll get to all of the small things in this order, rather than if we have one big amorphous blob of, uh, of user story. Um, yeah, it's, it's way harder to manage the expectations. And I think that's a positive thing for people in my team as well. Um, because it reduces the stress, but like, it's easier to, to kind of commit to it tickets in that way. Hm.
- Speaker 0 00:16:00 Um, so do you know if there are any, or have you experienced any specific, um, um, any specific aspects that are always taken into account when you perform a split of a user story, or do you have an idea of what is actually, you know, what is taken into account?
- Speaker 1 00:16:25 Um, I mean, I guess I can mostly speak from myself there when I tried to give an guy one of the main things that I try to keep in a, in a con when, when deciding to split the story on, on along what lines they split it is to, um, reduce the amount of people that need to be involved in the work. Um, I would prefer, uh, stories to be just within the theme of developers. So it might, the story might involve back end and front end work and that's fine. Um, but as best if the story doesn't involve also design and a decision from the B out and a input from another technical person, at the client, because the more other people need to be involved, the less, um, predictable the tickets,

it becomes again, because you don't really know when you'll get an answer from them or when they have time to sound board, uh, well, when they will come back or when they will make a decision. Um, so that's, that's kind of one heuristic. I tried to optimize for one split splitting to reduce the amounts of stakeholders per ticket. So I started to speak, um, along with the other things that I mentioned, that we, we do try to keep the tickets as a deliver value, so merge this tickets and end-user can next week start using it. Um, so that it doesn't become just back end or just front end or, uh, preparation for something in the future. That's another thing. Um, yeah, I, I think that's mostly it.

- Speaker 0 00:18:11 Yeah. So in terms of, uh, you talked a little bit about, uh, techniques maybe, and also guidelines, but are there any specified guidelines or techniques either within your company or just within the team that are always, uh, followed when you do a split, when you split the user story or perform user story splitting apart from the ones that you've mentioned,
- Speaker 1 00:18:36 We don't have anything written down formally, uh, in, in, in that case now we do have, I mean, what is a guideline here or a guideline is that we have, uh, a sprint demo every two weeks on a planning session afterwards. And then on every other week we have on the same day, uh, date and time we have our backlog grooming sessions so that those are followed quite rigorously. Um, no, I don't think there's any hard and fast rules. There's another colleague of mine in this project. Um, who's a agile coach, so he comes and helps out. Um, sometimes he listens into our meetings. He guides us a little bit. Um, he might be following some, some strict rules, but I don't think so either most of it is, is his feeling and, um, addressing thing based on intuition, based on what other people say based on feedback on, based on that retros, of course, we have a retro, maybe every third sprint, roughly every three sprints.
- Speaker 1 00:19:46 We do a retro for an hour, um, with the development team and we and that stuff we, we tried to implement going forward. Uh, oftentimes it's, it's helps sometimes it's really hard to make changes, um, in between the demo and the sprint planning session. We have a really quick retro around with our clients, um, which I think is mostly about, uh, us communicating to the client, how we are, uh, dealing with the project. If it's, if it's stressful, if it's too much, if we're, if things are unclear, it's a drop and an opportunity for the clients to indicate whether or not they are satisfied with the work that we're doing. Um, so even though we call it the retro, that moment is less of a agile retrospective, maybe, uh, those we do more every once, every two weeks or three sprints.
- Speaker 0 00:20:42 And so, uh, looking at, uh, if you imagine a, the perfect split of a user story, uh, would you be able to describe or define what goes into that or what defines the perfect match maybe?
- Speaker 1 00:21:11 My internet connection seems to have a dropped.
- Speaker 0 00:21:14 I thought you were back now. Yep. So, uh, I don't know how much you heard, maybe nothing,
- Speaker 1. Nothing, nothing at all
- Speaker 0 Okay. So if you wouldn't imagine the perfect split of a user story, are there any definition what, uh, how that would be or what, uh, how that would look?

- Speaker 1 00:21:39 Um, I mean, I'm making this stuff up on the spot, of course, but if I were to give a different definition, is that I split results in more tickets with a lower total complexity value. Sorry, I just got a call. Um, so if, say we have a tick worth 13 complexity points, um, I would imagine the perfect split to result in two tickets worth five points each. Um, and then two times five is 10. And obviously, so the reducing the complexity and uncertainty to such an extent that the individual tickets by themselves are less complex than the big ticket.
- Speaker 0 00:22:24 Is that something that you try and actively work towards, like all the time? Or is it.
- Speaker 1 00:22:32 No. Like I mentioned, uh, that the special was made up on the spot slide. If I'm pressed to give a definition of that, that's what I would say, but that's not really something that I'm aiming for in my day to day. Maybe something I should though.
- Speaker 0: Sounds good.
- Speaker 1 : Sounds good. In theory.
- Speaker 0 00:22:49 Yeah. Uh, so, uh, you mentioned a little bit about retrospectives. Uh, is there any, do you ever address, uh, what you could do to improve the process of then going from 13 points to five points, for example, or is it more of a retrospective, uh, like in total?
- Speaker 1 00:23:13 But typically they are in total, but we have discussed, uh, at certain retrospectives our way of working around estimation and pointing. Um, particularly after a couple of rough sprints where we, where we misestimated a little bit and we try to uncover some of the reasons that's, um, that's, uh, estimation is hard, uh, or why we're misestimating or how we can improve that, um, result of that retro was one, um, earlier involvement from
- Speaker 1 00:23:46 The client and the PO on tickets. So not just in time scheduling, but a couple of sprints further down the road, which is something we've implemented on, which seems to be going all right. Uh, and other parts of the outcome was also accepting the fact that it's uncertainty, um, and, um, looking at each individual tickets, uh, taken might be pointed as eights, but in reality has been 13 and are their tickets might've been pointed as I put in reality, I've been five. Um, and I think part of it is also that's that we, um, how do you say, uh, expressed, uh, explicitly that it's, that it's also an estimation and not a contract, so those are the things that we've discussed and then kind of put into practice. I think we've also discussed if we should just go away from pointing and sprint, uh, altogether and just start doing combined. Um, so dragging things in and working on them as long as they need to, and then moving them to done and then driving something new in. Um, but we haven't, we haven't implemented that..
- Speaker 0 00:24:58 Hmm. Okay So, uh, would you be able to, um, uh, tell us some of the, in your perspective, some of the main benefits of, uh, dividing bigger things into smaller things apart from them, uh, having a lower complexity?
- Speaker 1 00:25:15 Hmm, Hmm. I think it's, I think it's easier for everybody involved to get it right. Um, so we have some designers on staff that sometimes going on and work for specific tickets or for specific features. Um, we have long-term visions that are set out by the, uh, product owners, but clients. And I think that the value of breaking it

down into smaller steps is, uh, is mostly that the chance of getting it right is much higher. Um, and the risk is much lower of if you do get it wrong, well, then we do this three pointer or this five pointer again, but we don't need to do a really big story again. Uh, and so in terms of project management, I, I think unlike deliveries, I think that's the, the big upside, splitting it into smaller things, make them more likely to succeed. And in terms of day to day work, I think it's probably more fun to work on smaller, more well-defined tickets, smaller tickets. It is the easier it is to define it. Um, the better defined it is. The, the more fun it is to work on it, because it's, it's easier to get into it. It's easier to see if it's done or not. Um, smaller changes are more easy to keep elegant. So keeping your, your additions to the code base, uh, clean of is easier in smaller tickets. So in that regard albums, that is just more fun to work with small tickets.

- Speaker 0 00:26:50 So I'm imagining just imagining in front of of in my head, uh, something looking similar to a Trello board, if you know what that means. So one ticket would then be one card, for example, would it then be structured as a, with a check checklist of things that, so when do you know that you're actually done with everything? Has someone defined a checklist for you, or is that something you do yourself?
- Speaker 1 00:27:17 What we do there is we define the acceptance criteria. So we have a bullet list inside the description of the tickets, um, describing it's w we need to be able to do this. We need to be able to do that. We need to be able to do this. Um, we described some edge cases that we can think of ahead of time. Um, I don't know if we're implementing a search like what's, can we search on which characters are allowed or that those type of edge cases we try to during the grooming sessions, think a little bit of ahead of time. Um, now, um, uh, just as a kind of a reminder or a support, um, we also have an explicit out of scope section in each ticket. Um, meaning these things do not need to be covered by this ticket. Of course they're allowed to be covered, but it's, it's explicitly not a requirement.
- Speaker 1 00:28:12 Um, because sometimes it's hard to kind of delineate where the feature stops, um, on, by putting stuff into, out of scope. It makes it easier to, you know, to, to do demark, the, the entries of the tickets. Um, we don't do so much with the checklist. We've, we've tried that a little bit, um, in which order things should be implemented or, or who like maybe the backend needs to deliver an interface first before the front end can do something with it. Um, but we, we don't typically do that for every story to mostly acceptance criteria and out of scope. Okay.
- Speaker 0 00:28:50 And the, the out of scope section comes from, uh, you as well doing the grooming session?
- Speaker 1 00:28:57 Yeah. Okay. So we have, uh, I have a weekly contact moment with, uh, the PO at our clients during which we set up, uh, a rough first draft, um, and that's pre grooming section, what we call it, which was just me on the designer on the, on our product owner. We discussed on, we, we set up a little sketch of, okay, this could be probably be out of scope. This is probably acceptance by the area. Um, but it's at a very high level, very vague. Um, then we both individually work on it's a little bit. So I think a little bit more about it. I capture some extra details and that tickets our PO does the same. Um, sometimes we have a little discussion in the common section of the, each

tickets, and then we have our grooming session together with, uh, with all of the developers.

- Speaker 1 00:29:49 And that's where we kind of finalize the acceptance criteria and out of scope. So we created the little sketch initially, and then together kind of agree on, on this and on the, um, and the condition for the, what, when our acceptance criteria and, uh, out of scope is done, those are done. If we can estimate it, it's kind of, that's how we try to, so we discussed the ticket. Then we write down stuff and acceptance criteria or out of scope until we feel comfortable estimating. And if we don't feel comfortable estimating it, then we need more criteria or split it up into different tickets.
- Speaker 0 00:30:32 Uh, so, uh, to give some, uh, the devil his due, what are some drawbacks of, uh, the user story splitting in your, from your experience? Or are there any?
- Speaker 1 00:30:44 Yeah, yeah, definitely. Um, it's a lot of overhead. Um, you, you get all the benefits of stuff we've talked about now, fewer risks, more fun work. Um, but it's also a lot of work, um, in this project I'm 25, 30% full-time involved on it's, it's practically the only thing that I do, I look over our backlog. I look over at sprints, I talk to people, I ensure that things are in the right size class, and complexity level, and that I try to specify them with acceptance criteria, so good or as good as possible so that it can be estimated and, and split if necessary, um, which is, uh, I think it works fine and it reduces risk, but it's, it's a lot of time that goes into that. Um, if we weren't to do any of that only just randomly talk them to randomly implemented, then we would have 30% time leftover to, to spend on extra development. Uh, so that's the downside, I guess.
- Speaker 0 00:31:58 So I'm out of questions. Uh, thank you for a lot of valuable information. Uh, do you think that we have, or I have missed asking you something that you might find beneficial for us to know?
- Speaker 1 00:32:17 Um, nothing I can think of off the top of my head. I don't think, um, yeah, a bit of context, maybe that's, this is how it works in this project. That's, um, I'm involved in that has been running like this for awhile. Um, but that, that process of course differs a lot to, for each project that I'm involved in. This one has, uh, because it has been running for, I dunno, like a year and a half or something, we've had the time to adjust our process, to tweak, um, to have all these retros, to, to find our way of working within the team based on everybody's preferences. And it's also a big enough project that we can dedicate, like twenty-five percent of the time to, uh, to, to, to, to maintaining this, uh, this backlog on user stories in this way, other projects that are smaller that's, you're either working by yourself or, uh, that are under more time pressure or that have, uh, that aren't sold in as like agile projects would rather a fixed and dates under certain set of features that you need to deliver for each project. The process is a little bit different. Um, I, what I described here was very much from the perspective of that one project. Um, not necessarily everything that we do at our company. Yeah.
- Speaker 1 00:33:51 Thank you. That's a good place to stop, I guess. Thank you.