

Nicole Kelley: [00:00:00] Welcome or welcome back to user centered by design podcast, where we talk about all things related to user centered design. In this podcast, we will focus on topics in the medical device development realm, but the principles we discussed can absolutely be applied to other industries as well. I'm Nicole Kelly UX UI designer at sunrise labs.

Kelly Catale: [00:00:29] I'm Kelly Vitale, principal, human factors, engineer.

Alex Therrien: [00:00:32] I'm Alex Therrien the director of user centered design at sunrise labs. And today we have a new friend.

Dave Hibbard: [00:00:38] Hi, I'm Dave Hibbard, uh, director of programs at sunrise labs.

Alex Therrien: [00:00:42] And today's episode. We're going to talk about where we have crossover between agile development processes. And a user centered design. We found that there's quite a bit of crossover that that works well. And we want to share our experience and some things that may help to, uh, find success within your own team. So let's, let's get started off. All right.

Kelly Catale: [00:01:06] Well, we've already introduced Dave. Director of programs here at sunrise, and he brings a wealth of experience to our team and our conversation today to talk a bit about agile and development processes in general.

So we can set the stage for the rest of the conversation. So. Dave, can you explain to us the two most common methods in medical device development?

Dave Hibbard: [00:01:29] Yeah. So for a long time medical device development was done doing a waterfall approach, and this was the approach that was originally recommended by the FDA.

And really, almost, even more than recommended, it was written into a lot of the procedures that they, they published, uh, in. Somewhat reason we'll say the last 20 years or so Agile's become, uh, much more readily acceptable and certainly the commercial development space, but, you know, even into the medical development space.

So in a waterfall process, all the design is done upfront. So the product is conceptualized designed, then implemented and then tested in an agile approach. It's a much more iterative process. So the idea is to be able to get feedback from users or, um, Testing to iterate, to iterate on the product and influence the way the product is developed, designed, and the way that the user interacts with it.

Nicole Kelley: [00:02:19] Awesome.

Alex Therrien: [00:02:20] What are some of the advantages from a why you would do one versus the other, you know, what, what did waterfall get people when we were using that a lot more frequently in medical devices?

Dave Hibbard: [00:02:31] I would say a couple of things. I mean, one, I don't think people really had the concept of agile, you know, before. Maybe 25 years or so ago. And so waterfall was, was really the most common process out there. A lot of the program management techniques come out of construction industries and things like that, where you're really, you're doing a plan. You're ordering materials, you were constructing sort of a road and maybe that requires, you know, bypassing an on ramp or off ramp, and then building the final product that you want.

Um, and medical device development. Um, the, the concept of using agile really has. No blossomed over the last 25 years. And I think the main reason for that is that, you know, many people have

realized that doing an iterative approach allows a much better control over what the final product is and a much better, uh, target as to what the final product should be.

Uh, in terms of being able to get feedback from users, develop a minimum viable product, instead of something that was originally visited and also allowing you to. Change with the market over the course of the development. So there are a lot of medical device products that are developed over a long period of time.

So you have problems to deal with like obsolescence, uh, changing markets, competitor, introductions of products, and the agile process allows you to really look at your product and make sure that you're not developing something that is at that point, months or years out of date.

Kelly Catale: [00:03:58] It's funny. I just had a thought about agile in the construction industry.

It definitely wouldn't work the same way as you were describing it. I'm thinking you're right. That that's how things started back in the day program management. Really probably a lot of its foundation is from well foundation, no pun intended from construction. And agile definitely wouldn't work in that, that instance, but it does, it does have a lot of value in the medical device realm.

Alex Therrien: [00:04:26] There might be some unforeseen agility and in that industry with, Hey, I made your kitchen. I don't like it. You got to be doing it's the most painful form of agile. So we talk about the crossover and in what agile process. You know, the film, the philosophical approach that we brought up earlier, that iterative approach is really where user centered design.

It happens, you know, being able to figure out what isn't working, how to refine, how to retarget based on what we learn as we go. And I think that's going to be a theme that resonates throughout today is that. If we can anchor on, on some of these core philosophical approaches, that's where we have found, you know, what we like about it.

One more, a softball for you. Dave is, uh, you know, what has led sunrise to adopt agile approaches? Because you know, it, it's something that we try and do across the board here. So what, what, let us do that. And what's been your role in that process.

Dave Hibbard: [00:05:34] Yes. I just wanted to talk about a couple of things there.

So one is that in the medical device industry, it's hard to follow a pure agile technique. You know, the, the medical device industry is governed by the FDA. Still has certain criteria. It has to meet, um, such thing such as plans need to be developed before implementation can be done. You know, design needs to be created before.

You develop a product and acceptance criteria needs to be created before you test to verify a product. Um, so what we've done at sunrise is we've adapted the agile methodology into different phases in an overall waterfall approach. Um, so we still have a design phase and implementation date phase, and the test phase, but within each of those phases, what we've done is, is adapt agile to allow us to quickly iterate.

Across each phase, but most definitely inside the development phase. Um, so when I came to sunrise in. Well five years ago. Yes. 2016, uh, had been using Angela at another company where I was previous to this. And when I came into sunrise, it looked like it could be a good fit for the type of projects that we do.

Uh, we do a fair amount of software, but we also do system engineering, UCD. Chemical engineering, electrical engineering. And since we do our heavy software, it was a great place to start and introduce

the company to the agile methodologies. And I think it's really benefited our company in being able to want, as I said before, really concentrate on the minimal viable product, but also have much better prediction about when we're going to be done.

Is the project tracking to a schedule tracking to a budget. Um, you know, are we getting done the stories that we want to get done in a certain period of time? Um, are we getting good feedback from the users or the product owner? And I think sunrise has really found those techniques invaluable over the course of the last four or five years, and really has helped to develop and grow our development processes and our company as a whole, uh, to take on bigger and more complex projects.

Alex Therrien: [00:07:39] Yeah. I personally have seen it as a real benefit to the way that we run programs here. It makes the integration of the kind of techniques that we look to do in our, in a project work really well. Fair disclosure here, Dave is actually one of the, the best PM's that we've worked with collectively at integrating user centered design, into a medical device process.

And, you know, it's been a, a benefit to learn. Some of the rules of the road of agile from the programs that we've worked with you. And we're certainly not experts in the Jew in agile approaches, but we, we appreciate getting to work with folks like yourself who are.

Dave Hibbard: [00:08:19] So I think the, the greatest thing about the combination of both agile and the user centered design approach is that they both have this iterative process design implement, evaluate and repeat.

And you can just repeat and repeat and repeat until you end up with what you want. And that makes. You know, both of them worked together. Great. Right. So let's go through a sprint. Let's develop some features that can be, uh, evaluated. Let's evaluate it. Let's get feedback on it. And then let's iterate that design and try again until we get it right.

Alex Therrien: [00:08:48] Yeah. Couldn't have said it better. I, so when we, when we go into these, there's the planning upfront and there's the. The strategic focus that we can take to this. So as we're thinking about where we're going to learn and where we're going to iterate, one of the parts where I feel like UCD has been able to, to meet the team halfway is to come in with a focus on the strategy.

What do we know from a product owner that they want to have from a product experience and be able to prototype those things early and to, to really lay out a program of effort where. We can say, we know what the focus and the strategic priorities are in the product, either from a human factor standpoint or from a design standpoint and, and start getting into those as, as part of the plan early on with the sprint and then get those into evaluation.

So we have enough looks at the key areas of the product and, you know, Kelly with your area of focus around human factors. Do you want to talk a little bit about how that strategy can come together to benefit?

Kelly Catale: [00:09:59] I think when you've been talking about iterative approaches for both agile and then UCD process, the first thing that comes to mind for my world is formative studies.

And. Oftentimes, what I find is companies that don't incorporate UCD. Well, we'll forget that formatives are coming up and the amount of effort it takes to prepare and plan for it. And that also takes coordination with team members and having this plan and the strategy ahead of time saying, okay, we're going to have two formatives and then a summative.

We can work in the planning for those formative studies and allow for teams to put effort forth in. Maybe we dedicate a sprint to create a branch of software so that we have a prototype to test

informative. And by integrating that strategy, that human factors sort of strategy from a higher level into the whole sprint planning and the bigger vision helps us stay focused.

It helps us meet our goals and admittedly, it helps us bring visibility to our plan. In general, I think we could potentially end up getting lost in the details or. Our team members, our counterparts in software or electrical or mechanical might not be as focused on when a formative is happening. If we don't communicate it in some way and being involved in sprint planning, being involved in helping create some of the backlog.

Helps bring that visibility right. To the forefront so that people can think about what features need to be made and what feedback are we going to be gathering so that maybe we wait to implement something until a little bit later after we've learned about it. I know I'm rambling a little bit, but it's, there's a lot to talk about here.

And I'm like, okay. Many things I could talk about from integrating with human factors, but I think from the bigger level, the strategic level, having those big. Milestones for planning for formative really helps on a lot of levels.

Dave Hibbard: [00:11:57] Yeah. To sort of give an example of what Kelly's talking about there. So in a lot of cases, if you're doing a user interface, you may have, um, a lot of structure that sits behind databases and, uh, API as to other devices or cloud structures.

Um, but in the short term, you want to get feedback on. Just the user interface that the, uh, the customer is going to interact with. So you may break off a part of the team, or you may, uh, develop a specific set of stories instead of a sprint, just to do the user interface without actually connecting that user interface to the underneath structure.

And that would allow the user centered design team to go off and do a formative and gain feedback. Well, the rest of the team kind of continues on and works on some of that infrastructure in the Interline sprint in between while the formative is going on.

Kelly Catale: [00:12:46] Much better said than, than I said, but I it's a... man, I had something I wanted to say and now I, I completely lost it. Oh, what I was going to say is that it's can be really disruptive if you don't plan for these kinds of activities, building that, that interface to be able to test. And then what happens if you don't plan for it is you either have a bunch of extra effort or the formative has to get delayed because something's not ready.

It just allows for a lot smoother planning and execution in the end for my experience.

Dave Hibbard: [00:13:21] Yeah. I think that, I think that's true. Another sort of. Topic. I wanted to bring up there. It's just, you know, I think a mistake that a lot of people make is they do one of two things. Either one, they, they plan their formatives much too far apart. Um, so they look at the formative as a, almost a verification activity that what I've done is correct rather than getting feedback upfront. And that's really, I think kind of against the whole agile concept and the, really the concept of. You know, ULI and human factors in general. Um, and then the other mistake I think that people make is not to include user centered design, user centered design or UX design while they are developing.

So you can't, it's not practical to a formative after every sprint. If you're doing this every two or three weeks, you can't go out and visit 10 users and get feedback, but you have people that understand and have studied that environment, um, you know, between the product owner and, um, The focus on user centered design.

So, you know, getting the feedback from them at the, every sprint is key to making sure that you're, you keep moving forward and you're not going off on a tangent to what ability, the product you want.

Nicole Kelley: [00:14:27] Yeah, I would definitely say making sure that, like you said, UX designers being involved in that sprint process definitely helps facilitate an almost quick in your feedback loop.

Even if you're not doing a formal or summative study, you can definitely get feedback quicker and more efficient. If you can add in the UX or even the UI at some points in small bite-size prototypes. So that during development. You can test it and check in with your product owner and make sure that you're on the right track.

Yeah. So that by then there's more ample opportunity to make changes instead of just having everything done at once and then having to go back and revisit something where that can cost a lot of time for the developers, because once the UI designer or visual designer is done, handing that off, it's sometimes a project on its own.

And making sure that you're in line with your developers, you software developers can really help speed up that process so you can attack it in more bite sized pieces.

Alex Therrien: [00:15:21] Attacking definition of the features, most of what you're going to find as problems or most of what you're going to find is misinterpretations.

When translating an intent into code. We can flesh those out a lot sooner and make sure that it's meeting the owners, product owners, vision to meeting the demands of business, meeting the mental models of the end users in the design efforts that we're, we're trying to do upfront and eliminate a lot of the ambiguity for how to interpret a user story, where to take it with how we've written it with the context.

And yeah, it just, it reduces a lot of, you know, potential confusion. Throughout the development process and leveraging that leveraging diversity of interpretation upfront in, in looking at very different options around each of the features. You can get a lot more uncovered quickly. And even if you're just ending with a rough prototype to say and make these changes with a, with a well-defined development and design handoff, um, you know, between two team members who know each other, well, you can solve the rest with whiteboards.

It just works a lot more fast, quickly and more efficiently. So as we talk about this iterative process and learning through B building, um, let's, let's dig into that a little bit more and talk about what that looks like in terms of, you know, when agile is doing a pure software learn and build type approach, Dave what's, what do you expect from the software side of.

Of testing and evaluation that's going on over there. And, um, maybe Kelly, you can compliment it with some of the ways that you've, you've used user-centered techniques and human factors techniques to, to compliment that. Well, Yeah. So I would say a couple of things. Um, so one just kind of going a little bit back to the previous topic as a program manager, one of the greatest aspects of having a good involvement, heavy involvement from UCD is that it gives me.

Dave Hibbard: [00:17:31] An expert in that area. Um, so, you know, I have a software lead as my, my architecture expert expert in my systems lead is my requirements and risk apps for it. And my UCD lead is my UI expert. And I think as a lot of people realize when you have something that's visual, be it, you know, the, the enclosure of the device or in this case, particularly the UI, the device.

Everybody has an opinion about it, and everybody thinks their opinion is right. And it can cause a lot of diversity within the UI. So having, you know, to be able to point back to one group or one person that

says, look, this is the person that, that understands the UI. This is the person that's studied the user interactions.

Uh, I think helps to cut down on a lot of unnecessary churn, which is. For a program management standpoint, fantastic opinion is never part of the conversation. I have never experienced this before in my life. I don't know what you're speaking about. So, you know, you're asking a little bit about, um, You know, just what are some of the ideas in terms of making sure that the software group from an agile standpoint is performing well?

So one is to make sure that the stories are complete and I'm not open to a lot of interpretation. So, um, Yeah, I suppose in the pure sense of, of agile, the stories might be a little bit vague, but I think as you work with a user centered design group, um, you need to make sure that while you're in the sprint, the stories that you are implementing are well-defined and have good acceptance criteria that way you've, um, not.

In essence, saddled the software engineer with having to think about the design as they go through trying to implement something, they have something in front of them, they implement it, they focus on the architecture. They focus on implementing the best code that they can. Uh, they focus on reviewing each other's code.

Uh, but they're not necessarily focused on, um, is this a good UI or is this not a good UI? You know, that responsibility lives with the UCD group. Um, and then if. It turns out that the UCI that you created, that UI, that you created in that sprint, it is not the best men. That's what agile is all about. You get to reconsider and you get to evaluate and you get to try again.

Um, and that's, that's. I think that process really lends itself well, um, it, it keeps everyone focused on their areas of expertise, uh, and kind of, um, I think stops some of the confusion within some of the team and team members.

Kelly Catale: [00:20:01] Alex, you had asked me a question about how does it relate to gathering feedback from a human factors perspective?

Can you maybe elaborate.

Alex Therrien: [00:20:11] Hey was, it was even a little bit more about how it compliments well or how you can seamlessly utilize those opportunities to, to improve what we're doing across the board.

Kelly Catale: [00:20:23] Well, one thing that I've found really valuable, just being involved in the agile process for development teams is being involved.

When we're talking about defining the stories that go into the sprints. If there's some input that I can offer, or if we're defining acceptance criteria and maybe there's something related to for that story. Maybe there's a use error that I know is related to this design. Maybe this certain feature is mitigating a potential user.

We can. Uh, to some language, to the acceptance criteria to say, Hey, let's make sure that this user has mitigated because ultimately it comes down to, from a human factors perspective, it comes down to mitigating use related risk. And I'd say that's one key component with how that aligns with my contributions.

Otherwise, there are other ways that I can gather quick feedback at the end, when we have a, at the end of a sprint, we might have something to show to either as Nicole mentioned to the product owner,

or might even be able to do some internal quick. Checking. I know that I've brought stuff to folks internally and just said, Hey, can you try this?

Or, Hey, can you tell me what the, what you think about this? Not a formal gathering, like usability test kind of format, but a quick touch point of someone who doesn't know the product at all. Can you give me some feedback on it or we can turn that around really quickly to the developers?

Nicole Kelley: [00:21:52] I definitely think that's. Not thought of a lot. You can get some great feedback by your peers, uh, just by making sure you can get, and it may not be up to the level of a formative, uh, you know, some of those really what's the word I'm looking for. So it was like really defined problems that you may need to solve right off the bat.

But if it's something where it's

Kelly Catale: [00:22:14] yeah, yeah.

Nicole Kelley: [00:22:15] You know, like, I mean, it's just, it's so beneficial. We'll have that fast checking to make sure that the interaction that you're creating is going to give you the result you're looking for. Because sometimes even though you're developing, it was a very, you know, thoughtful process.

You can miss some small things just because. You're in the middle of work. So it's really good to check in, even with some people on your team that you're working on the same project, or you're not working on the same project. So you can get an even broader look at testing quick test, just to get some more feedback, more feedback, the better, especially in terms of

Kelly Catale: [00:22:48] yes.

Alex Therrien: [00:22:49] Yeah. I love the, the idea that we are always pushing, which is how do we lower the barrier for touch points with the user? And there's, there's always an amount of expense. There's always an amount of planning and commitment that goes into, you know, getting a really deep look, but whatever we can do to find less formal, more, more frequent touch points, you know, either through local people who might be closer to these or group or.

Um, or trying to find lower cost ways of, of approaching the same problem of, of getting more looks at the problem because no one gets it right on the first time. And, you know, I think that's been one of the key successes for our team integrating with, you know, agile focused development is bringing, trying to get more iterations in so that we're succeeding and we've even done it on the hardware side.

You know, as we're, as we're trying to find. Fits the human problems that that, or solutions to problems were fit to human as an issue. You know, we're, we're looking at that. We're building prototypes quickly, roughly. And what is the minimum viable learning experience that we can drive, you know, and it, isn't just a software.

It's the most commonly applied form of agile, but, you know, we've had success with longer sprints that allow for more hardware development. Um, Bringing electrical engineering, mechanical engineering, and industrial design into the picture to be successful with the application of it. Um, Dave, when we, when we talk about that, what's, what's some of the learning process that we had going beyond software with this, you know, you're, you're probably one of the high art folks here for what, what we do with agile.

Dave Hibbard: [00:24:37] Yeah. So there's definitely a couple of things to consider. Um, So, so one is that the process most likely is foreign to them. So people working on the hardware, electric engineers,

mechanical engineers, um, probably. May have heard of it, but always considered a software tool, not a process that could be applied to other disciplines.

Um, so one of the, one of the things you need to do is you need to make sure you break down the stories into small, uh, bite-sized chunks, if you will. So, you know, on the software side, we'd like to see stories that are in the range of one to three days. Um, and we usually end up doing. For a team of maybe three to four software engineers, three week sprints.

If the team is larger, uh, you know, maybe six, seven, eight software engineers. And we would most likely have to have a dedicated scrum master. And the sprints probably would be shorter. Um, more along the lines of two weeks, uh, in the case of hardware. Um, again, you want to try and get those tasks, um, as small as you can.

So, you know, instead of saying design a circuit board, it might be. Design the power input to the, to the circuit board, you know, design the display, um, interface, those types of things, you know, design the touch interface. And one of the things that we've really tried to do on the software side, as we develop things in bite-size chunks, we review them.

Um, so we actually have our source control database set. So you can't. Uh, promote code into the source control database, unless you've had at least two people review and approve that software. Um, so you can follow a similar process on the hardware side, where again, if you're, um, biting off a schematic or something into a smaller units, you can review that particular. Section of the schematic before you either check that into your source control or fusing all Tam or whatever the case may be. Um, certainly before you go to any kind of layout, uh, you would want to do review, uh, on that as well. And you can apply that to the, to the mechanical side in a similar fashion.

The other, the other thing that I would bring up there is that, um, You need to consider how you are going to integrate with the tools that you have. Uh, so for instance, if you're using a tool like SolidWorks and PDM and they have their own workflows and their own way of controlling, uh, the drawings, so you need to be able to make sure that you are agile process and task management process, uh, coincides, and really gets along with the process of controlling 3d models, drawings, et cetera.

Alex Therrien: [00:27:05] One of the other parts that I think, especially from an industrial design workflow that we have to think about when we're integrating with agile is the creative tension that will exist between a design process, which I think is very compatible with agile in the fact that it looks from a user perspective, you know, how are we defining the product?

But the design process, a lot of times is looking. From sort of bigger picture, less fidelity passes at defining the product and sort of, you know, eliminating possibilities and then getting into the details. And so looking holistically at the experience of the product and then getting into the sub components were part of agile is breaking things into bite sized pieces.

And so part of the process of especially industrial design is, is bringing in. A way of incorporating those kinds of, uh, really thinking of the product and early definition of the product to allow them the space earlier on to, to get through that, that end of the funnel before getting too into breaking apart the product into individualized component experiences, because that's, that's usually where you end up with less consistency and less, um, coherent design experiences.

And, uh, I, I think just as you were talking about solid works and that workflow and how you're able to, to incorporate the way that mechanical engineering tools work, that, that sort of big picture view. It is a key part of, of any designed experience and flowing into there. Okay. So let's, let's bring it down to some brass tacks here, as we think about the practical reality that we can help people with here, what

are some of the things that we can help share from our experience that people can take and apply to increase their success as a team, combining these disciplines, um, you know, what, what do.

What are some of you think are key areas of focus for, for that kind of, uh, successful integration?

Dave Hibbard: [00:29:17] Yeah, it says a few, few areas that come to mind. Um, one would be just, you know, making sure your stories are complete, making sure they're there right size, you know, they're not, uh, over scoped. They're not too ambiguous have acceptance criteria.

Uh, you know, don't. You know, give the team something to work with. Don't just give them a blob and throw it over the fence and be like, okay, see what you come out with, you know, let's, you know, work together and, and, and lead that team. Um, so that's one, I think, you know, putting too much into a sprint, I think is a common mistake.

So you you're like, I gotta get this done. I gotta get that done. I gotta get this, you know, this. Feature in before the 1st of March or whatever the date may be. And you ended up putting a lot of stuff into the sprint. I think it causes a quite a bit of consternation within the team as you, you know, you're not going to get that stuff done.

When you, when you plan it, you come to three weeks later, it's not done. Everyone's like, ah, we feel like we failed. Um, and then you repeat so. Making sure that you're learning from the sprints that you've done in the past. Um, you know, looking at the concept of velocity or some other metric to know how much you can get done, uh, in a similar length sprint with a similar team and, you know, making sure that you don't over plan, I think is also fairly important.

Uh, Nicole look like you had a couple of things you wanted to throw in there.

Nicole Kelley: [00:30:39] Yeah, I did have a question. I was curious on what your thoughts would be and how beneficial you think it is to have you CD a part of that early process in, you know, device development or project development. Uh, definitely in sprint planning meetings as well, because sometimes I've often heard that even though it is, there are, you know, linear approaches to both design and product development and software and stuff like that.

Um, but sometimes it may be beneficial for some UCD aspects or in my case, UX UI designers to start almost a sprint ahead to give the developers or the software engineers, something to work with as well as working with them. Um, in other cases, what I mean is having a design at a point where we can hand off some materials.

We're still working, but they're still working, but at least having that buffer room before they know they have to start working, if that makes sense. What are your thoughts on that?

Dave Hibbard: [00:31:36] Yeah. So at sometimes what we usually try and do is we usually in, uh, during the. Design phase. So we have what we call phase one, which we do, right?

A lot of the plans, we also write, um, architectures and requirements documents. Um, at that time it was also when we want to do the first pass, that information architecture, wireframes that type of stuff. And maybe, you know, if we can fit it in even a formative, um, on that information, then we can look at some of the styling of the UI, some of the information layout of the UI, what kind of features would be available.

You know, easily and what kind of features are maybe hidden under a menu, that type of thing. And then I think you're absolutely correct as we go through and do the sprints, um, you know, in a, in an

ideal world, um, A feature that's going to be implemented in sprint three is gone through the, the UI UX process in sprint two.

And, you know, from when I run a project, I like to include those types of tasks as part of the sprint process. So, you know, I'll have a story on there that says, you know, develop the menu system or develop the settings window, uh, from a, from a UX standpoint so that when we come to the next sprint and I always say, implement that, uh, settings feature, then we know exactly what that's going to look like.

Nicole Kelley: [00:32:50] Yeah, I think that works really well. And I think it's really good to have that type of information beforehand and definitely a good handoff for the developers. It creates a good relationship too. I feel working on a team, uh, because you definitely have that good buffer space and you're not too stressed out or worried that you're not going to have something well-defined enough for a developer when your hands are tied as well. So I really appreciate that process that we do here at sunrise.

Dave Hibbard: [00:33:14] Yeah, I think, you know, speaking of the relationship, it really gives the software engineer someone to go to. Right. So they, they know, they're like, okay, I'm going to be working on implementing blank feature, but they know when I come to the next sprint and that first feature was done and I want to work on the next one that, you know, whoever's doing that UX work is, has already gotten that to them.

And, you know, has a good start on what it's gonna look like. And. Uh, I was going to end up, so, you know, that relationship, I think ends up being key.

Nicole Kelley: [00:33:41] Yeah. I think that definitely ties into the second point we kind of had, which is, you know, making sure we're holding effective sprint reviews and making sure that everybody's voice is heard within a sprint review or a sprint planning meeting to make sure that everybody's heard.

And everybody knows what they're doing.

Kelly Catale: [00:33:55] In addition to sprint reviews and sprint planning also I'd say the best experiences I've had with agile development is when we have regular scrum. Meetings as well, not just a once or twice a week, but every day, because it gives an opportunity for the team to gather and ask quick questions or not even asking a question but more, Hey, Nicole, I'm going to find you after the meeting because I'm having trouble with whatever kind of opens up the door for that communication.

And visibility within the team, rather than waiting until the end of a sprint to make it known that there was an issue or, uh, having someone have to discover it by reading a comment in a ticket that they may or may not see in a timely way. So the scrum meetings also, I would say, are really, really important and.

Especially helpful for keeping everyone engaged with the UCD team too. And letting allowing us to share where we are in our tasks, which might be a bit more nebulous to some folks than potentially a quick, uh, software development feature effort or something like that.

Alex Therrien: [00:35:06] Yeah. The communication that takes place there.

So the, the formality of the agile workflow of, Hey, scrums need to happen. It should be daily or every two days, just to make sure that everybody's got a touch point. They're checking in, they're talking about what their progress is and giving updates on their velocity. It's another way of lowering the barrier to communication.

Any sort of over the wall is the death knell of good design and, you know, embedding with the team. And you know, one of the, the parts that Nicole and Dave you were talking about earlier is the relationships. Good design in an agile environment, happens from democratizing the understanding of design and by investing in those relationships, our peers start to learn how to appreciate what the intention is, what the philosophy is underlying, the, the design of the product, and they can take that and run with it, and they can also find logical inconsistencies.

And it's yet another way of lowering the barrier to improving the product and getting more iterations. And anything around this is good communication. And that, that's kind of the key to all those little points that I was just talking about is that we're getting FaceTime with people. We're an integrated part of the team and we're, we're helping to make sure that everybody feels like they're accountable for a great experience because no one holds the keys to those, that kind of creativity.

Dave Hibbard: [00:36:28] Uh, another place where the relationship really matters is it's, it's easy for. Easy or maybe even common for, um, the architecture to really prevent a good UI and, uh, you know, uh, the UI to maybe prevent a good architecture. Uh, and you know, those two teams really need to work together or. Elson, you can be, you can be putting false constraints on constraints that really don't need to be there in the way that the UI is going to flow and work.

Um, based on the way the architecture was created. So they both need to be considered as you're developing the software. Um, you know, both the UI side or the infrastructure and database side. That's my least favorite moment is when you get to the, you know, third to last sprint or second to last sprint and you're sitting there and you're like, okay, well, the right decision here is that the architecture is locked, man.

Alex Therrien: [00:37:21] Things can blow up, man. Don't, don't go back there. And, uh, yeah, those kinds of conversations really hurt. And yeah, I love the fact that, um, you know, as a, as a, you know, an aside here we've been pushing a lot of. Um, you know, systems engineering here is big and we have systems architectures. We've been doing a lot of parallel path of, of UX architectures and then coming together right at the front and saying, okay, let's compare notes.

What, what adds up, what doesn't add up and, and sort of negotiating those things out by having walked through the user experience story in our, you know, as a, as a logical exercise, to make sure that, you know, foreseeable issues from a human experience standpoint, don't make it. Uh, the architecture doesn't exclude them or what an assumption is around the human story of it.

Doesn't block a robust and stable architecture from taking place because all of those things really do add up to the user experience. If you have a Bluetooth connection that keeps dropping every 30 seconds, like that's, that's not a screen, but it's like one of the worst experiences that you can be chasing, like.

Every third word is dropping out and your phone call, that's awful. No one wants that. Uh, another, another issue with that, um, is in a similar vein is making sure that you handle. Air handling early on in the project. Um, it's particularly the higher class of, uh, product that you're making. So if you're working on a class two or class three, there may be a lot of conditions that the device needs to be able to handle, and they may have to guide the users through resolving those particular air conditions.

Dave Hibbard: [00:38:58] And if you don't consider those early as part of the initial information architecture, initial UX design, uh, it becomes. Very, uh, awkward to the user, I think to experience that error situation and then try and handle it with a UI wasn't really ever intended to do that. So, you know, it was part of having, you know, the flow that the normally user would follow.

You also need to really heavily invest in considering what are the, what are the error paths that you don't really want the user going down to, but, you know, eventually that they're going to get there. Yes. And treating those front, you know, Nicole has been doing a lot of great work lately on the systemizing of the way our designs come together and thinking about their, their consistency at the earliest stages, so that when we apply design solutions to problems, they're driven throughout the product.

Alex Therrien: [00:39:50] And as we learn, then they become better. And, um, I I've really appreciated your work in that space. Nicole, it's been, it's been great to see that start to percolate everywhere.

Nicole Kelley: [00:40:02] UX is such a hot topic nowadays. I mean, if you don't have a UX engineer or UX designer, a human factors engineer early on in the team, you know, they're really trying to push that because like you said, Alex, learning those.

And, you know, learning the user at the beginning and figuring out those architectures and making sure that you're not missing a system that needs to be fully thought out, can really start breaking it and increase your scope. Tenfold. If by, as we said, the third, fourth to last sprint, and you need to finish this design and now you need to rethink almost everything because you weren't iterating enough in the beginning or having enough.

No users set user centered thought in the beginning to really dive into those problems that some people maybe weren't even thinking about back then, I think it's really important. Yeah.

Alex Therrien: [00:40:49] Yeah. And I think that comes back to a point that we were talking about a little earlier and maybe as a, as one of the key takeaways that we can wrap up on is allowing the time upfront, like you were saying the coal to, to do some investigation of the design, but also thinking about it more globally.

Like I had mentioned earlier, Because those global investigations of the entire product is where that consistency starts to STEM from. And thinking about the architecture of the design experience before getting into the smaller chunks, that's where you end up having train wrecks at the end of a, of a classic agile program, because what is consistent within a feature sometimes doesn't scale well to being consistent for other features.

And that's where the architectural robustness. From a system standpoint, as well as from a user experience standpoint can get out of sync and you know, all the points that I'd like to make as we wrap up, allowing the time and for that global look to, to drive consistency throughout an interface and a designed to experience.

That's, that's a big one for, for what I, I think is important for the integration. And what are some other ones that you all would like to make sure that people were walking away with? Yeah. One thought that has kind of popped into my head is, you know, have a little fun with it. Right. So I would, you know, for one of my projects, I'd love to, uh, have Alex create a persona of one of our users and, you know, have a Nicole just sketch it up and dry it out there and I'll stick a, stick it on everyone's cube.

Dave Hibbard: [00:42:21] So they can always kind of keep that person in mind. You know, it's, it's just, uh, build the team, build the relationships. And I think things will go well.

Kelly Catale: [00:42:31] I think that's a wonderful idea

Alex Therrien: [00:42:34] staying present in everybody's mind, you know, it's it's, you know, whatever we can do to get rent free space.

Kelly Catale: [00:42:40] I think just being invested early in the planning and recognizing that there is value in having integration with your UCD.

Team members is key. If we haven't beaten that into the ground enough today, hopefully you can take that away and say, yes, there is a lot of value. It is, it is doable. And it is incredibly valuable for the team

Alex Therrien: [00:43:05] earlier episodes. One of the points that you've made in the past Kelly is, is being strategic about that.

And so upfront, if we know that there's. A technology or an aspect or a clinical condition that is like high-risk for our medical device, the features that traced to that, putting them up front in the build because we need more time with those to get them right. That's been such a key point that you've made over time.

That it's a great success factor for the way that we do it. And it's probably something that other people can leverage. Yeah. And a couple of, couple more things I would throw out there is, you know, there's obviously lots of ways to approach product development. You can consider the, the monkey's taping approach and, you know, if you get lucky, then you may get the fastest and the cheapest to the finish line to get exactly what you wanted.

Dave Hibbard: [00:43:55] But most likely it's going to end up in. You know, not creating a novel, it's going to create, treat some disaster that needs to be cleaned up. So, you know, by mitigating risks and bringing in UCD early and often, then you can, you know, help guarantee the success of your product without, uh, you know, going off the rails, so to speak.

Alex Therrien: [00:44:17] I'll I'll make that check an extra 500 bucks at the end of this. So, you know, just, just so you know,

Kelly Catale: [00:44:25] I'd say we covered a lot today in our conversation. This was really fantastic. And yeah, Dave, thank you for joining us. You provided a lot of insight and it's clear that you have a lot of experience in this, and we're excited that you were able to share your thoughts with us.

Dave Hibbard: [00:44:42] Thanks for having me on. It's actually been a pleasure, great conversation.

Alex Therrien: [00:44:47] It's been a ton of fun, getting hanging out with you again. And, um, you know, it's, uh, it's to our benefit to have somebody as experienced and agile as you and, and you know, who seen enough in the trenches with user centered design approaches that we're getting this kind of integration.

So, yeah, it's great to have you as a partner and. Yeah, thanks everybody for joining us today. Bye.

Kelly Catale: [00:45:08] Thanks Dave.

Nicole Kelley: [00:45:10] Make sure you're following or subscribe to our podcast channel to get the latest updates on our new episodes. You can find us on our website@sunriselabs.com or Spotify, Apple podcasts, Google podcasts.

Thanks for listening. And we hope you tune in next time.