

Daniel Litwin 0:03

Welcome to the sunrise podcast powered by sunrise labs.

I'm your host Daniel Litwin the voice of b2b. The fundamental question that comes with designing new med tech devices is this one. What are the user needs and requirements? The question feels simple enough, but it's far from simple. With that question comes others. Who is the user? What makes a good requirement? How do you actually test these requirements for a medtech Designer like Niklas new ski loss director of electrical engineering for sunrise labs, what he cares about the most is user needs that are unambiguous and testable. On today's healthcare podcast, let's new ski loss digs into this multi layered question of delivering on user needs. And defining needs versus requirements versus actual mandated FDA requirements. He breaks down the importance of atomists at the way user needs affect everyone from the manufacturers, to the patients, and his process for getting everyone involved on the same page. So that potentially life saving product can make its way efficiently to market. Alright, Nick, welcome to the podcast. How are you doing today? I'm well, Daniel, how are you? I'm great. Thank you so much for joining us. And I'm excited to dig into something that we haven't touched on on our healthcare podcast before. And that's really looking at user needs and requirements and healthcare design. You know, I think when it comes to the healthcare industry, a lot of these kinds of processes, you know, whether it's implementing new technology, whether it's designing physical spaces, whether it's designing the technology, right, anything that has to do with implementing something new into the healthcare space often comes a little slower, it can be difficult, because not only are you dealing with a lot of regulations, but you know, you do have people's lives literally in your hands. So you want to make sure everything is to the specifications that are required. And that gets us to user needs and requirements, right. As you know, a company that provides these services, you obviously want to be in tune with what your users really need. So to dive in, firstly, I'd like to look at what even is the user right, I think that is the biggest issue is defining who the user is here. Are we talking about the patients? Are we talking about the physicians, you know, who here is actually using the space? So tell me a little bit about when it comes to meeting these requirements for the user? How do you even define who the user is? Yeah, that's a great question. So I think really different people define it in different ways. And that can be frustrating, it can be a great challenge when trying to work with our customers, because they will often have a different idea about how to even define the term. And what I mean by that is, is I think everyone agrees that a patient is going to be a user of a medical device. Most medical devices, for a surgical tool, for example, the patient may be asleep, they may not be technically a user, though, of course, we still need to consider their needs during the surgery. I think everyone also agrees that the clinician or the physician who is using the device counts as a user.

Nick LL 3:26

I think the agreement sort of ends there. And then you have different companies and different people who will consider different levels of user within their user needs statements, including service people, service technicians, or even people on the manufacturing line. And sometimes for big, complex assemblies, you may need people to get into the device and install things or move things around as part of the manufacturing process. And in those cases, their safety is actually quite important. And their ability to do their job and to complete production of the device is quite important. So considering them and user needs can be helpful. But they're not an end user. They're a skills technician. So sometimes people when I say people, sometimes companies who are deciding how to define their user needs wanting specifically excluded needs relating to manufacturer or service technicians, because they are expected to be skilled, and to have instructions on how to operate the system. And there, it's somewhat out of the scope of what the FDA is expecting when they refer to defining your user needs. And the view of some of these companies is that it's not strictly required by the FDA, then they don't want to include it because they don't want to risk being audited on it. And that's a reasonable approach. I can't really disagree with that approach. But I think it makes my job more difficult as a product designer, because I need to consider not just the needs of the patient and the physician, but also the technicians are also beyond that, you know, marketing needs are often kept separate from user needs, but really the two are overlapped heavily, right? So you might have a

market need to sell the device in a certain market or to sell it at a certain price point. And you may say, Well, those are really user needs, they don't deal with the safety of the user, or the ability of the user to use the products. But they do if a user can't purchase the products, because they either can't afford it, or they don't live in a region where it's sold, then they can't benefit from it. So, you know, do you put those in a marketing needs statement, or you put them in a user need statements? Again, from a product development standpoint, I don't really care where those statements come from, I need to consider all of them. So it's easier for me to just think of them all as user needs, right?

Daniel Litwin 5:46

It's just so many different challenges. Because when you are designing these products, like you said, You're not just looking at, how is this going to perform technically the best? It's also how can I make sure to source this with products that are at a price point that people can actually purchase them and use them right and that you're not putting too much money into the the design of it into the actual materials that go into designing the product. And then at that point, you have to sell it at a higher product. or excuse me, you have to sell it at a higher price point and then make a profit. And then at that point, no one's buying it, right. It's all these little things that how do we actually get this product in the hands of the user, and every step in between is technically a user need. Because if the user, whether that's the clinician or whether that's the patient who is receiving the device, can't access it, it doesn't work for them. And that is because of a manufacturing issue, a distribution issue, a price point issue or a technical issue, all of that falls under user I think no user need.

Nick LL 6:50

Yeah, I definitely agree. I think it does all fall under what I would consider user needs. I think it can be a touchy area, though. And that's where we get into trouble. Because, you know, there are guidances out there. And there are rules out there, particularly with the newer versions of, of the regulations 1491, for example, that preclude us from considering cost when we assess risk in certain ways. And so it's it, people are very afraid, companies are very afraid of making the statement, well, you know, the cost target is part of our needs. Because if we don't make the cost target, and we can't sell the product, technically, that's, that's, that's touchy, it's a touchy area. However, it's very true area, the The fact of the matter is that if your product is too expensive, you're only going to get so much reimbursement for it. If it's too expensive, you can't sell it, which means the cut of the customer can't buy it, which means they can't extract any benefit from it. So So we certainly have to consider those factors in the

Daniel Litwin 7:52

in the design of the products. And then you know, beyond that, when you look at the actual phrasing of the word user needs or user requirements, the word requirements often feels like okay, if you don't meet this point, there's going to be some issue, right? Someone is going to bring the hammer down and say, Hey, you didn't meet this user requirements. But I think you and I both know here that it can be very rare for these requirements to actually even be testable. And a user need is more of something that is not a true requirement. But it's more of like you said it yourself, a user need statement, write something that they express what we need out of this and you try and meet it to the best of your ability, it's not necessarily the requirement that someone comes down and says, Hey, you didn't meet this requirement. This needs to be you know, this can't hit the market. So how do you try and balance that side of it both meeting, you know, what the FDA is looking for in their actual requirements, and then also looking at what the user requires of you. But more on on a trusting level? Right. It's more of just a out of respect for the product and respect for getting it in the hands of those who need it, not necessarily some some mandated law?

Nick LL 9:10

Yeah, I think there were a few a few sort of issues and questions for up there. So I agree. I think there's a lot of muddling between the the concept of user needs statements versus user requirements. A lot of our customers will come to us, you know, companies big and small. And they will give us what they call the user requirements specification. And it's filled with with these phrases with these

requirements that are written as if they were requirements, but the people who wrote them clearly wrote them as user needs statements. They are not they don't meet some basic requirements of being a requirements. For example, they copy testable, which is really important to requirement because if I can't test to it, it means I can't show that my product actually meets the requirements. So I can't prove to you that my product does what I told you it would do. So testability is extremely important in a formal requirements on. So normally what I what I try to encourage and push with the customers that come in is this view that the user requirements and they use air quotes around requirements, there are really more of user needs statements. And what we're going to do is create system requirements that tell what the product is going to do. And then we we test to those we verify against the system requirements. And that is our way of showing them that the product does what we told them it would do, they still have to validate the product, which means they have to show that it meets the user's needs. But that might be a combination of usability studies and tests to show that the user requirements are met to within their comfort level of it. But that's a little bit less of a quantitative process, then the system requirements verification, where we'll say that the product does something specific, it might measure temperature to within one degree Celsius. And so we have to come up with the tests that puts it in the mode where it's measuring temperature, show that it measured it show that the specifications are that meets that specification, and document that and that's our proof that our product does what we said it would do. Right. And I think at the core

Daniel Litwin 11:19

of that issue is, is the requirement even a good system requirement? Right? Like how do you align that between someone like yourself, who is actually designing the products then with the manufacturers and with the users themselves? Because it sounds like oftentimes you have clients that will send you over their user requirements. But they're, they're not verifiable, they're not testable. And that makes it really difficult for them to even know if what you're delivering is what they wanted in the first place. It's just this, this nebulous soup that you can't really do much with. So tell me a bit about that side of it. How do you get everyone on the same page of what is a good requirement in the first place? You know, what should we actually be setting standards for what which of these requirements are testable? You know, it break that down a bit for me.

Nick LL 12:11

So there, there are two aspects to that. The first is about how we define requirements around our product, or product development efforts. And that is we call that translation from the user needs two requirements, we take the user needs as given to us by our customers or by our partners who gather user needs from from users, and through market analyses. And those may come to us in the form of a user of a set of user requirements or user requirements specification. But at that point, we're treating those as user needs statements, we're not treating those as verifiable requirements, then we translate those into system requirements, usually in collaboration with marketing with the customers, in order to come up with those verifiable requirements. And then we verified to those to prove that it does what we said it would do, that we're actually people generally agree on pretty well, the part that people don't agree on well, is the second part of your question, which is about what makes a good system requirements. And we actually just had a meeting with five or six of the systems engineers and project managers within sunrise, to talk about how we, we manage this, this process of what we consider a good requirement and how we can better train our engineers to understand how to both read and write requirements. And all of us could not agree on what the most important parts of a requirements are. To me, as someone who has been an engineer, you know, since birth, all I really care about is producing something that is good producing something that does what I said it would do, and that does what it needs to do in order to be successful in order to help people. I'm in the business of designing medical devices, because I want to help people. And I want to make sure that the devices I design are best able to do that. So a lot of guidelines around requirements writing are aimed toward that, but don't really hit the mark, in my opinion. So there are requirements like, or there's expectations, like requirements should be unambiguous, they should be correct. They shouldn't be consistent. They should be traceable, they should be unique, they should be atomic. And of those, I think, by far the most important things to me are unambiguous and testable or verifiable. If a

requirement is ambiguous, then toothcomb people reading it might read in different ways. It's very difficult to say that whether that what the designer has designed actually meets what the author of the requirements meant, and that can be a huge problem and is intrinsic in language. Human language is full of ambiguities. There there we have homophones we have words that sound the same. We have words that are the same but mean different things. If I use the word love, I could be talking about romance or score in tennis, it makes it very difficult to write on ambiguous statements about what a product will or will not do. But that's that, in my opinion, is the ultimate goal of the requirements. Others like Adam Issett II, so atomic requirements are is one that only requires one feature or one aspect. So if you have temperature measurements, some people will be very adamant about making the the range of temperature measurement, a separate requirement from the accuracy of the temperature measurement, I really couldn't care much less about that, I'm fine with combining requirements, it doesn't really matter to my ability to design a product that meets those, it eases the job of the testers to make requirements atomic, because they can test just one aspect. And then if that aspect is not met, if the product does not meet that aspect, then you only have to go back and retest that one aspect. So it's it makes testing a little bit more streamlined if your requirements are atomic. But I really, from a design standpoint, I really don't care that much, right?

Daniel Litwin 16:10

Well, because if your requirements aren't atomic, and you're sort of asking for I want to make sure this does said thing, but said thing requires three different aspects to work in conjunction with each other and work well. You know, you test it, it doesn't work, then all of a sudden, you're like, Okay, well, which of these pieces isn't the one that's working, which requirement was not met. And then at that point, you you just don't know where to go next. And then you have an unhappy user? on your end, you know, you're probably stuck saying, Okay, well, they didn't really even give me enough direction to begin with here. So it must make the whole situation really complicated. Which brings me to my next point. And that is, you know, I'd love to hear an example from you of a complicated situation where you were given a specific user need or requirement, maybe it wasn't in you don't need to name any companies, if you don't want to really just sort of describing the situation and how you found the best way to sift through what they laid out and find, okay, this is the requirement that they're really asking for here, you know, this is, this is a good requirement, I'm going to get back to them and say, Hey, I don't think this is really what you're asking for I recommend blank, kind of walk me through one of those examples.

Nick LL 17:16

Yeah, that's a good question. So I would say that it's not usually complicated. It's not that the user needs statements, or the provided user requirements are complicated. It's that they're usually very nebulous for their inexact, again, it comes back to ambiguity. So one that comes up all the time, I would say every single user requirement specification, I've seen put in front of me had a requirement that said, the device shall be easy to use. And that's, that's great. You know, of course, we want the device to be easy to use, we know that that takes to, to whether or not the user will buy it, it relates to whether or not you'll get one star or five star on Amazon. And that's the difference between your company going under, and your company thriving. Hack four stars to five stars, and Amazon's is the difference between your company failing and succeeding. So that's a, it's a really tough requirements to try and address to pare down to system requirements. Because what makes a device easy to use can be can vary wildly from whether you're talking about an MRI machine, to a Fitbit, right there, the the needs, there are very, very different. I would say that usually it comes down to some very basic concepts. And that's what we'll start to dig into with the customers is but when you say it shall be easy to use, what are you envisioning will be difficult to use about it. And often we find some consistent themes like readability of text readability of text, sorry, the text in terms of its height, and its contrast, particularly when you're talking about devices with small displays, or even apps, which we do a lot of these days, getting contrast between your different elements between your different texts on the screen, can be very, very important to the usability of a product, and is worth taking the time to figure out ahead of time what the guidelines are, in fact, there are some guidances on that, that one topic and trying to design those in from the beginning, so that you don't get any big surprises when you go

through usability studies. There are a lot of other aspects that can tie into that, particularly when you're talking about displays your viewing angles, which can be you know, a big problem. Everyone knows about that with TVs, you set the wrong angle, you can't see anything. Oh, imagine that you're relying on that device for your life or your life, you really want to be able to see it. So that can be a pretty important thing about being easy to use, right? That's

Daniel Litwin 19:41

the that's the bare minimum, right? You want to be able to at least see everything that's going on. Right, right.

Nick LL 19:47

That brings up a good point to actually go bare minimum. So when we talk about requirements, and you mentioned this before, we meet requirements, so the specifications of the device and what it actually does have to exceed the requirements fit better than there are requirements, but the requirements are what this thing must do in order for it to help people in order for it to be sellable in order to be safe. And that's, that's an important distinction to make. Because if you start requiring things that are really arbitrary, then you end up with bloating your effort enormously. We see this all the time with user interfaces. So we get these user interface specifications that specify, you know, border width, and, and very specific colors on everything and pixel offsets within a box. And making that, that, you know, making that level of detail work can be very difficult depending on what platform you're developing for. And certainly, if you're working on something that's cross platform, if NASA work in Linux, and Windows or iOS and Android, it can be very difficult to make the make a consistent look and feel across both of those. And you're wasting a lot of time pushing pixels around to do things that the the apple look and feel doesn't want to follow, which ends up costing a lot of money and not really providing a better user experience. I mean, people who use Apple products want that Apple language in their user interface. And people who use Android products, expect the Android language, interface language in their interface. And, and by forcing something else, you create extra work, and you make a product that doesn't, doesn't really speak to the users as well.

Daniel Litwin 21:27

Right? Yeah, I mean, it's, it's just so many different aspects that you need to balance to make sure you deliver on on those user needs a, I definitely feel your pain, just you describing it. Yeah, but you know, at the same time, it can be exciting to you know, when you have user needs and requirements that are laid out nicely, then you have a clear path of what you need to accomplish as the designer, it makes it makes reaching that goal a challenge, right, but at the same time, it not a nuisance, and you can get there and deliver on something that is you may be going to change an aspect of the industry for the better, you know, it could have really, really large positive consequences. So that's, that's just must be an exciting aspect of what you do as well. It is, yeah, we

Nick LL 22:16

work on some very, some very cutting edge type technologies. And it's pretty impressive to watch these things for and watch these high tech fully custom, you know, life support devices that people are going to be relying on for their lives, to roll out the door. And to know that they see the videos of people using these devices and, and to see the reactions and to see how important it is to them. Because it improves the quality of the life of life and in some cases, supports their life is just it's really heartwarming. And it's, it's why we all come into work every day, I think.

Daniel Litwin 22:54

But you know, it's also important to remember, you don't get there unless the requirements that are laid out in front of you are verifiable, are unambiguous, right? And so it's really important for everyone to be on that same page and for the users to communicate what they need a little better. And then to make sure that the teams that are in place who are designing this know exactly what to look for, right, are well versed on what makes a good system requirement. You know, when we're looking at what they're asking for, how do we know what to then relate back to them? Hey, could you be a little less

ambiguous with this? Is there a more atomic part of this that you need us to test? Right? So yeah, it's an exciting time, I think we're starting to see so much great med tech enter the industry. And I'm really excited to see how you and your team continue to help set a standard for, you know, what is a really great user need, what is a great user requirement? And how can everyone put their heads together and get on the same page to deliver on technology that, you know, is really needed in this industry. So again, thank you so much for joining us on this podcast. Nick. Really enjoyed the insight.

Nick LL 24:03

Thanks for having me.