Traceability – meeting regulatory requirements

With Sven Wittorf, Prof. Dr. Christian Johner

Transcript

00:00:05 Speaker 1

Medical Device Insights, a podcast by the Johner Institute for medical device manufacturers, authorities and notified bodies.

00:00:18 Speaker 2

If you have an auditor who is not very familiar with your particular product class, then this auditor will probably be more formal.

00:00:28 Speaker 2

And 1 of these

00:00:29 Speaker 2

Checks that are more on the formal side are traceability checks, for example between the requirement and the associated tests.

00:00:38 Speaker 2

However, these traces are not so easy to draw and how to do it, which tools can be used and which regulatory requirements should be observed, we discuss in today's podcast.

00:00:52 Speaker 2

For today's podcast, I have invited Sven Widdorf, the managing director of our sister company Metzotum.

00:00:59 Speaker 2

Sven also spends a large part of his time modeling facts, developing tools for them and in this context, of course, it is also very much about the topic of traceability.

00:01:11 Speaker 2

Hello Sven, nice to have you here.

00:01:13 Speaker 1

Hello Christian, thank you for letting me be there.

00:01:16 Speaker 2

Yes, when we talk about traceability, we should perhaps first clarify what traceability actually is.

00:01:22 Speaker 2

How would you define something like that?

00:01:24 Speaker 1

So from a software point of view, it's clearly the tracking between

00:01:28 Speaker 1

Requirements that can be on different levels.

00:01:31 Speaker 1

I think we'll come to that later and especially between the requirement and the associated tests and just the proof I have one or more tests for each requirement and which requirement is covered by which test and maybe even the bit further, yes, these tests have actually been successfully executed.

00:01:49 Speaker 1

You shouldn't confuse the two, if you don't come out of the software and only see 13 485, then traceability is

00:01:56 Speaker 1

also often understood as and not misunderstood, as traceability of my products in the market.

00:02:02 Speaker 1

In other words, traceability in a classic Q.

00:02:04 Speaker 1

M.

00:02:05 Speaker 1

The meaning is actually traceability of products in the market.

00:02:08 Speaker 1

But I don't think we want to talk about that today.

00:02:09 Speaker 2

Or absolutely, yes.

00:02:12 Speaker 2

If we are talking about traceability and some still differentiate the definition according to vertical and horizontal traceability, does it make sense, do you do it that way?

00:02:21 Speaker 1

That definitely makes sense.

00:02:23 Speaker 1

So, if we look like this in the V.

00:02:23 Speaker 1

model, which should be a concept for everyone, at least as a documentation model, not so much as a process model, then the vertical traces on the left side of the V.

00:02:35 Speaker 1

model, where you just descend deeper and deeper implementation, from the very top stakeholder requirement to system requirements, software, mechanics, hardware, if we don't just talk about software or components and so on,

00:02:50 Speaker 1

There's a hierarchy in it that you have to have clear for yourself.

00:02:53 Speaker 1

We also refer to this as vertical traceability.

00:02:56 Speaker 1

So, which system requirement follows from which stakeholder requirement and horizontally is the test level that stands against it.

00:03:04 Speaker 1

So just like it is in the V.

00:03:05 Speaker 1

model, you also have different tests, test types and test goals.

00:03:11 Speaker 1

on the different levels.

00:03:12 Speaker 2

You just mentioned the 13 for 85, in the context that you shouldn't confuse it with the tracing of products in the market.

00:03:22 Speaker 2

Does the 13 for 85 still have such kind of traceability requirements in the sense we are discussing right now and what other regulatory requirements for traceability are there?

00:03:34 Speaker 1

So the 13 for 85 says yes, I have to survey the requirements of my customers

00:03:40 Speaker 1

und zwar stated and not stated by the customer.

00:03:43 Speaker 1

I always find that very nice.

00:03:44 Speaker 1

So even the requirements that the customer didn't actually say, I have to write them down if they are relevant.

00:03:49 Speaker 1

And I think what stands above all is the demand, the legal requirement, that I have to build a high-performance product.

00:03:57 Speaker 1

This means that I have to prove that the product performs well.

00:04:01 Speaker 1

This means that I first have to write down what performance actually is.

00:04:06 Speaker 1

And there is just the 13 485 specified, the ,n

00:04:10 Speaker 1

a bit deeper, I would say, in the chapter where it's about development and product development and in there is actually the 62 304 standard, which drills this whole topic of product development for the topic of software again and it goes into much more detail into the topic of architecture, tracing, testing and also testing, I'll say, lower software level at component level.

00:04:33 Speaker 1

So we say that's actually quite a hierarchy.

00:04:35 Speaker 1

At the top is the legal requirement, 13 485 makes the

00:04:39 Speaker 1

Processes around it and 62 304 sounds the same, you can put that very nicely next to each other, even the two chapters on development and you can see that they are almost structurally identical, only 13 for 85 is generic and 62 304 deals specifically with software development.

00:04:54 Speaker 2

Could you say that the 62 304 requires both vertical and horizontal traceability?

00:05:00 Speaker 1

Ah, that's a difficult question.

00:05:02 Speaker 1

I don't want to, I don't want to gossip about the 62 304, but in the end, from our point of view, there is

00:05:07 Speaker 1

little, so the answer is yes and no, in any case, of course, horizontal traceability, as far as requirements are concerned, but so this vertical traceability between the different levels and also the distinction between what levels there are and how they differ and why they make sense and what happens between the levels, I find very, very weak in actually all normative or at least normative requirements, the

00:05:34 Speaker 1

harmonised.

00:05:35 Speaker 1

I explain this a bit by the fact that it doesn't really matter to the regulator whether the product is developed efficiently afterwards, whether customer needs are really satisfied, the main thing is that it is safe and efficient.

00:05:48 Speaker 1

Maybe it's a bit provocative, but the norm itself, the

00:05:53 Speaker 1

62304 talks about software requirements and system requirements.

00:05:57 Speaker 1

In fact, it says that system and software requirements can also be the same if it is, if the system is pure software, but somehow doesn't say anywhere, it doesn't give any help in which they differ.

00:06:08 Speaker 1

And we actually see that in practice, that this also causes a lot of problems for companies.

00:06:12 Speaker 1

Well, I, I just have, I have to be in the V.

00:06:15 Speaker 1

Model make several levels of requirements and then I just write several levels without really having a clear rule as to what belongs where.

00:06:23 Speaker 1

how does that differ, then I often actually get problems at the test level, because I cover several levels or at least imaginary levels with the same tests in the end.

00:06:33 Speaker 1

And unfortunately, unfortunately I have to say, the standard gives little help.

00:06:37 Speaker 2

Yes, on the contrary, it may even be confusing at some points, if we think back to the traces of the units, again very high up to the requirements, then it almost skips a level.

00:06:47 Speaker 1

Yes, there are also in this this listing, I need to raise the requirements if

00:06:52 Speaker 1

what types of requirements there are.

00:06:54 Speaker 1

There is functional requirement and then there is data definition and then there is interfaces and from our point of view these are actually at least 2 different levels that are being talked about.

00:07:05 Speaker 2

Yes, while we're talking about this software requirement, maybe we can zoom in a bit and think about how the software requirements can be traced to the architecture.

00:07:17 Speaker 2

What would be your tips, is it even possible?

00:07:20 Speaker 2

So that you can

00:07:21 Speaker 2

a software request for an architecture item feature can somehow trace or how should this be done in concrete terms?

00:07:29 Speaker 1

It would, it requires 2 things first.

00:07:30 Speaker 1

One is that I uniquely identify my software requirement.

00:07:34 Speaker 1

I think it's very simple, I number them consecutively or I have them in columns in Excel or I have them or in rows, or rather, it's very simple.

00:07:43 Speaker 1

The second is, I think, much more difficult, that I mean my architecture decisions in individual

00:07:51 Speaker 1

I can certainly do that.

00:07:54 Speaker 1

So that, there, there is called, if I think of it again in hardware, I have components and they just get an I.

00:07:59 Speaker 1

D.

00:07:59 Speaker 1

and then that's a part of my system development, my architecture.

00:08:03 Speaker 1

But that, the real tracing on it is not so easy.

00:08:08 Speaker 1

So there are several considerations to.

00:08:09 Speaker 1

The first is, for me, the architecture of a system is the answer to the sum of all requirements.

00:08:16 Speaker 1

And if I go now

00:08:18 Speaker 1

And no, I'll call it a 100% traceability demand.

00:08:21 Speaker 1

So every architecture item that I somehow identify has to be traced to every system requirement and vice versa, then I get massively into trouble with it.

00:08:31 Speaker 1

So this, I always call it 100% traceability, I definitely need it between requirement and tests, no question about it, at the system level, i.e. the level I use to prove performance.

00:08:44 Speaker 1

But when I do this at the architecture level, I see several problems.

00:08:47 Speaker 1

The first is, there are non-functional requirements, performance requirements, for example, on which architectural item, on which line of code, on which.

00:08:57 Speaker 1

Resistance I trace a performance requirement.

00:09:00 Speaker 1

The system has to deliver a result within, I don't know, 10 milliseconds.

00:09:03 Speaker 1

That, that doesn't work, because the architecture as such, the overall architecture, satisfies such requirements and vice versa, even if I have an old architecture database, for example, and if I want or have to trace it, I say, oh, I have to trace everything and then I look for all the requirements that have something to do with persistence, then it is completely pointless from my point of view, because it just doesn't help.

00:09:26 Speaker 1

That is, the the.

00:09:27 Speaker 1

I would definitely not call for one hundred percent traceability at this point.

00:09:31 Speaker 1

Certainly it makes sense, for example, if I have any requirements to, I have to implement a certain algorithm and have the direct trace into the architecture, in this, in this component it is implemented.

00:09:43 Speaker 1

That makes perfect sense, but definitely not such a dogmatic one, in the end I have to create a trace report that proves that every component has a motivation that is directly derived from a

00:09:54 Speaker 1

individual identifiable requirements.

00:09:57 Speaker 1

And I also think that would be the second aspect, that this does not do justice to this whole activity of architecture.

00:10:05 Speaker 1

So it's almost insulting.

00:10:07 Speaker 1

So that would mean that I can prove the quality of an architecture through simple traceability.

00:10:12 Speaker 1

That, that doesn't work.

00:10:14 Speaker 1

So for me, that's actually the main value creation that is in an entire development is that I think about an architecture and I'm not going to be able to create the

00:10:23 Speaker 1

Goodness to somehow prove this architecture by traces on demand.

00:10:28 Speaker 1

And then I have a 100% green tick somewhere and then it's like, yes, that's great architecture.

00:10:33 Speaker 1

I don't think that makes sense.

00:10:36 Speaker 1

It is also not a valid statement.

00:10:38 Speaker 1

Well, I only see whether the architecture works at the end, when I have built the product according to this architecture.

00:10:42 Speaker 1

And then I have my performance requirements again.

00:10:46 Speaker 1

Through which I can then prove the whole thing.

00:10:48 Speaker 1

This also means that such a trace would itself be a review of the requirements or the architecture.

00:10:53 Speaker 1

Excuse me, by a peer, by a second architect is actually nothing more than I call it a qualified assumption that the architecture will be good.

00:11:02 Speaker 1

and the real proof, which is then also relevant from a regulatory point of view, I then provide by actually testing the requirements that led to this architecture on the finished product.

00:11:14 Speaker 1

And until I have done that, the product does not go out to the patient.

00:11:18 Speaker 1

This means that until then there is no danger at all, and from a regulatory point of view, I would say that this traceability requirement for architecture or the proof of the quality of the architecture says

00:11:29 Speaker 1

I can do the 62, 3 or 4 myself through traceability, I can do it through traceability.

00:11:34 Speaker 1

Our recommendation would always be to do this with a proper peer review, but in my view that is also a quality management requirement, i.e. a requirement for the.

00:11:43 Speaker 1

Process and not a requirement for the safety of the product, because I will not bring the product to a patient until I have tested the performance against the level above.

00:11:53 Speaker 2

What would be the consequence of this for the manufacturers?

00:11:56 Speaker 2

Would you recommend that they then completely dispense with this topic of traceability between software requirements and software architecture?

00:12:03 Speaker 1

So, what's really exciting is what works and I'm also fascinated by why.

00:12:07 Speaker 1

I haven't quite figured it out yet, that you, that I result from the architecture yes

00:12:12 Speaker 1

other requirements.

00:12:15 Speaker 1

So architecture is ultimately a decision that I make about how to build my system, for example.

00:12:20 Speaker 1

I have a list of requirements for a system and I say the system works when I have this component, A.

00:12:26 Speaker 1

B.

00:12:26 Speaker 1

C., and who interact with each other in this and that way.

00:12:29 Speaker 1

And that, in turn, leads to the fact that I have requirements for these components, now from point of view.

00:12:35 Speaker 1

In this way, the architecture that the

00:12:39 Speaker 1

the next level of requirements and very often it works very well that I do the tracing via this architecture level between the requirements myself.

00:12:48 Speaker 2

And that brings us directly to the next question, namely the question of tools.

00:12:54 Speaker 2

In your opinion, what would be characteristics, skills that tools should have in order to achieve good tracing, whether horizontal or vertical tracing?

00:13:07 Speaker 1

So there are

00:13:09 Speaker 1

So, we see 2 types of tools coming from 2 directions.

00:13:14 Speaker 1

And ultimately, there are, let's say, 3 areas that these tools cover or 3 challenges.

00:13:19 Speaker 1

One is what you just mentioned, which is that I have to prove that my product is safe and capable as it is.

00:13:27 Speaker 1

That means I have that the output of it goes into the product file.

00:13:31 Speaker 1

Then there is the area, I also have to prove that I have lived a compliant process during that time.

00:13:36 Speaker 1

If so, then I want to have everything from one tool.

00:13:38 Speaker 1

This is more audit, I'll call it audit resistance, which means that I have to prove with the output of this tool, yes, I have done a review, yes, I have released it and that must also be somehow feasible with a time stamp and possibly even with electronic signatures.

00:13:55 Speaker 1

And the third now

00:13:56 Speaker 1

very close to that of software development is the topic of collaboration.

00:14:01 Speaker 1

So I want to be able to say in the tool, who does what, how long do I need, I might want sprint planning if I work agile, with burndown charts and so on.

00:14:10 Speaker 1

And now there are actually 2 corners from which such tools can come.

00:14:14 Speaker 1

Some tools really come out of development relatively hard, some of them are in the I.

00:14:19 Speaker 1

en.

00:14:19 Speaker 1

somehow integrated, they are very good in the topic of collaboration, they are also good at it in principle

00:14:25 Speaker 1

Status transitions and so on.

00:14:27 Speaker 1

This means that I can already prove a certain process conformity with them, but they are often not good when it comes to extracting a document in the end that can really put it in a file.

00:14:40 Speaker 1

So that's an urgent recommendation that we have when you work with tools and use them to develop medical technology.

00:14:44 Speaker 1

Make sure you get a PDF out, so to speak.

00:14:47 Speaker 1

The manufacturer is happy when he gets an order for 15 years.

00:14:51 Speaker 1

But if I know what will happen in 10 years with the manufacturer, I know that I can still read the data.

00:14:55 Speaker 1

That means I need some kind of reporting that I can do in a neutral tool and there is P at the moment.

00:15:00 Speaker 1

D.

00:15:00 Speaker 1

F., I would say, the tool of choice.

00:15:02 Speaker 1

So even X.

00:15:03 Speaker 1

M.

00:15:03 Speaker 1

L.

00:15:03 Speaker 1

I usually get it out, but I have to format it again and if I then use a very strict Q.

00:15:08 Speaker 1

M.

00:15:08 Speaker 1

or even just a semi-strict Q.

00:15:09 Speaker 1

M., who then wants to have all this validated, very time-consuming.

00:15:12 Speaker 1

And this is where the tools that come from the requirements engineering corner come from.

00:15:17 Speaker 1

Yes, they come from above, so to speak.

00:15:18 Speaker 1

They can do it very well.

00:15:19 Speaker 1

They are, because they think, document-oriented.

00:15:21 Speaker 1

They also think, also in output, which I generate.

00:15:26 Speaker 1

And often they are not so close to the code and the others, who are very close to the code, are then not so close to the development.

00:15:34 Speaker 1

And my recommendation when you decide to use tools is always the first question: What, what, what do I need?

00:15:39 Speaker 1

Do I need some kind of lid tool to organize my documentation for me?

00:15:44 Speaker 1

then I look more at this level of requirements engineering or do I need a tool where the developers are really working in the code on a daily basis?

00:15:51 Speaker 1

Then we are in the tools that are really very close to the code and they come from the manufacturers of the software products.

00:15:57 Speaker 1

So that's a, that's Team Foundation Server and all these stories that run in the Microsoft environment, for example.

00:16:04 Speaker 2

Where would you locate your products, i.e. the method products such as Riskpack and Bedpack?

00:16:10 Speaker 1

So we have exactly this one,

00:16:12 Speaker 1

We don't build everything from Scratch.

00:16:14 Speaker 1

We are based on a tool.

00:16:15 Speaker 1

This is the Polarion tool, which has since been taken over by Siemens.

00:16:19 Speaker 1

Exactly because we hold this very, very powerful in the area of documentation and.

00:16:25 Speaker 1

where you can also map variants, you can actually make any branches of documents that are very close to the code.

00:16:32 Speaker 1

Someone who has never developed software before can also work there.

00:16:35 Speaker 1

So it's very word-like in parts, but it still has a database behind it that enables all these traceability analyses and so on.

00:16:44 Speaker 1

And it also works: It can also go very deep into code, but on a level that you don't really need from a regulatory point of view.

00:16:52 Speaker 1

So very few of our customers use

00:16:55 Speaker 1

Polarion itself, to go into the code, but sometimes have tools like Jira, just to name 1, where they do all their team management, the topic of collaboration, and what you do then is that you have to find a level and that often works very well, in which these tools can then be found.

00:17:12 Speaker 1

That is, I have a layer of information that I can synchronize bidirectionally.

00:17:16 Speaker 1

This is actually very, very easy and very good.

00:17:20 Speaker 1

And then, for example, the bugs found in Jura go directly back into Polarion so that they are ready for documentation and further processing.

00:17:28 Speaker 1

you have a bit of the best of both worlds and you don't take away people's usual tools, which they have been working with for a very long time.

00:17:36 Speaker 1

What, what, I think, is the most wrong approach you can make in tooling, because maybe a subordinate clause of whether a tool is successful or not, that's our experience in the more than 10 years I know in which we've been doing this now, depends very little on the features or

00:17:49 Speaker 1

the compliant output, but first and foremost on whether I manage to get people on board to work with this tool.

00:17:55 Speaker 1

And that, I think, is what, what is underestimated in practice.

00:17:58 Speaker 2

Yes, Sven, that's actually a beautiful final word.

00:18:01 Speaker 2

I think we've covered quite a lot now in this podcast, but of course there will be more questions.

00:18:07 Speaker 2

Maybe a tip on this, Sven and I have written an article, a technical article on our website, which we will link to below.

00:18:17 Speaker 2

And I think you won't have any difficulties with it if we recommend that all those who are interested in tools, the handling of tools, the mapping of traceability with tools then think of you.

00:18:30 Speaker 2

And if you agree, Sven, we would also link your contact details below.

00:18:35 Speaker 1

Yes, of course.

00:18:36 Speaker 1

I'm looking forward to it.

00:18:37 Speaker 2

Yes, then all I can do is say thank you very much, Sven.

00:18:40 Speaker 2

See you very soon.

00:18:41 Speaker 1

Thank you and goodbye.