

Traceability - die regulatorischen Anforderungen erfüllen

Mit Sven Wittorf, Prof. Dr. Christian Johner

Transkript

00:00:05 Sprecher 1

Medical Device Insights, ein Podcast des Johner Instituts für Medizinproduktehersteller, Behörden und benannte Stellen.

00:00:18 Sprecher 2

Wenn Sie einen Auditor haben, der sich mit Ihrer speziellen Produktklasse nicht besonders gut auskennt, dann wird dieser Auditor oder diese Auditorin wahrscheinlich eher formal prüfen.

00:00:28 Sprecher 2

Und 1 dieser

00:00:29 Sprecher 2

Prüfungen, die eher auf der formalen Seite einzuordnen sind, sind Überprüfungen der Traceability, beispielsweise zwischen Anforderung und dazugehörigen Tests.

00:00:38 Sprecher 2

Diese Traces sind aber gar nicht so einfach zu ziehen und wie man das macht, welche Werkzeuge man dazu einsetzen kann und welche regulatorischen Anforderungen beachtet werden sollten, das besprechen wir heute im Podcast.

00:00:52 Sprecher 2

Für den heutigen Podcast habe ich mir dazu den Sven Widdorf eingeladen, den Geschäftsführer unserer Schwesterfirma Metzotum.

00:00:59 Sprecher 2

Sven verbringt ,n großen Teil seiner Zeit auch damit, Sachverhalte zu modellieren, Werkzeuge dafür zu entwickeln und in diesem Kontext geht es natürlich auch ganz insbesondere um das Thema Traceability.

00:01:11 Sprecher 2

Hallo Sven, schön, dass du da bist.

00:01:13 Sprecher 1

Hallo Christian, danke, dass ich da sein darf.

00:01:16 Sprecher 2

Ja, wenn wir über Traceability sprechen, sollten wir vielleicht erstmal klären, was Traceability überhaupt ist.

00:01:22 Sprecher 2

Wie würdest du sowas definieren?

00:01:24 Sprecher 1

Also aus der Softwaresicht ist es ganz klar die Nachverfolgung zwischen

00:01:28 Sprecher 1

Anforderungen, die auf verschiedenen Ebenen sein können.

00:01:31 Sprecher 1

Ich denk, da kommen wir später zu und insbesondere auch zwischen Anforderung und dazugehörigen Tests und eben der Nachweis habe ich für jede Anforderung ein oder mehrere Tests und welche Anforderung wird durch welchen Test abgedeckt und vielleicht auch noch den Stück weiter, ja, sind diese Tests eigentlich auch erfolgreich ausgeführt worden.

00:01:49 Sprecher 1

Man darf das nicht verwechseln, wenn man jetzt nicht aus der Software kommt und nur die 13 485 sieht, dann ist Traceability

00:01:56 Sprecher 1

auch häufig verstanden als und auch nicht falsch verstanden, als Traceability meiner Produkte im Markt.

00:02:02 Sprecher 1

Also Rückverfolgbarkeit in einem klassischen Q.

00:02:04 Sprecher 1

M.

00:02:05 Sprecher 1

Sinne ist tatsächlich Rückverfolgbarkeit der Produkte im Markt.

00:02:08 Sprecher 1

Aber ich glaub, darüber wollen wir heute nicht sprechen.

00:02:09 Sprecher 2

Oder absolut, ja.

00:02:12 Sprecher 2

Wenn wir gerade über Traceability uns unterhalten und die Definition unterscheiden manche noch nach vertikaler und horizontaler Traceability, macht es Sinn, machst du das auch so?

00:02:21 Sprecher 1

Das macht auf jeden Fall Sinn.

00:02:23 Sprecher 1

Also, wenn wir so im V.

00:02:23 Sprecher 1

Modell denken, was jedem ,n Begriff sein sollte, zumindest als Dokumentationsmodell, gar nicht so sehr als Vorgehensmodell, dann sind die verticalen Traces auf dem auf der linken Seite des des V.

00:02:35 Sprecher 1

Modells, wo man eben immer tiefer Implementierung absteigt, von ganz oben Stakeholder Anforderung über Systemanforderungen, Software, Mechanik, Hardware, wenn wir nicht nur über Software reden oder auch Komponenten und so weiter,

00:02:50 Sprecher 1

Da steckt ,ne Hierarchie drin, die man auch sauber für sich klar haben muss.

00:02:53 Sprecher 1

Das bezeichnen wir auch als vertikale Traceability.

00:02:56 Sprecher 1

Also, welche Systemanforderung folgt aus welcher Stakeholder-Anforderung und horizontal ist eben die die Testebene, die dagegen steht.

00:03:04 Sprecher 1

Also genau wie es im V.

00:03:05 Sprecher 1

Modell auch gedacht ist und hat man eben auch unterschiedliche Tests, Testarten und Testziele.

00:03:11 Sprecher 1

auf den verschiedenen Ebenen.

00:03:12 Sprecher 2

Du hast gerade eben schon die 13 für 85 erwähnt gehabt, in dem Kontext gesagt, dass man das jetzt nicht verwechseln sollte mit der mit dem Tracing der Produkte im Markt.

00:03:22 Sprecher 2

Hat die 13 für 85 trotzdem so Art Traceability Anforderungen in dem Sinn, wie wir es gerade besprechen und welche anderen regulatorischen Anforderungen an die Traceability gibt es da?

00:03:34 Sprecher 1

Also die 13 für 85 sagt ja, ich muss die Anforderungen meiner Kunden erheben

00:03:40 Sprecher 1

und zwar stated and not stated by the customer.

00:03:43 Sprecher 1

Das finde ich immer sehr schön.

00:03:44 Sprecher 1

Also auch die Anforderungen, die der Kunde eigentlich gar nicht gesagt hat, die muss ich aufschreiben, falls sie relevant sind.

00:03:49 Sprecher 1

Und ich glaub, das was über allem steht, ist die Forderung, die gesetzliche Forderung, dass ich ein ein leistungsfähiges Produkt bauen muss.

00:03:57 Sprecher 1

Das heißt, ich muss nachweisen, dass das Produkt seine Leistungsfähigkeit erbringt.

00:04:01 Sprecher 1

Das bedeutet, dass ich überhaupt erst mal aufschreiben muss, was die Leistungsfähigkeit eigentlich ist.

00:04:06 Sprecher 1

Und da ist eben die die 13 485 präzisiert, das ,n

00:04:10 Sprecher 1

bisschen tiefer, würde ich sagen, in dem in dem Kapitel, wo es eben um Entwicklung geht und Produktentwicklung und da rein hängt sich eigentlich die 62 304 Norm, die dieses ganze Thematik Produktentwicklung für das Thema Software noch mal aufbohrt und die geht also deutlich detaillierter auch in das Thema Architektur rein, Tracing, Testen und auch Testen auf, ich sag mal, niederer Softwareebene auf Komponentenebene.

00:04:33 Sprecher 1

Also wir sagen, das ist eigentlich ,ne ganz schöne Hierarchie.

00:04:35 Sprecher 1

Oben steht die gesetzliche Anforderung, 13 485 macht die

00:04:39 Sprecher 1

Prozesse drumherum und 62 304 klingt sich eben, das kann man sehr schön nebeneinander stellen, sogar die beiden Kapitel zur Entwicklung und man sieht, die sind fast strukturgleich, nur 13 für 85 ist generisch und 62 304 geht konkret auf Softwareentwicklung ein.

00:04:54 Sprecher 2

Könnte man sagen, dass die 62 304 sowohl die vertikale als auch die horizontale Traceability verlangt?

00:05:00 Sprecher 1

Ah, das ist ,ne schwierige Frage.

00:05:02 Sprecher 1

Ich möchte, ich möchte ja nicht lästern über die 62 304, aber letztlich gibt es aus unserer Sicht

00:05:07 Sprecher 1

wenig, also die Antwort ist Jein, auf jeden Fall natürlich die horizontale Traceability, was Anforderungen angeht, aber so diese vertikale Traceability zwischen den verschiedenen Ebenen und auch die Unterscheidung, was es für Ebenen gibt und wie die sich unterscheiden und warum die sinnvoll sind und was zwischen den Ebenen geschieht, die finde ich sehr, sehr schwach in eigentlich allen normativen oder zumindest ja normativen Anforderungen, die

00:05:34 Sprecher 1

harmonisiert sind.

00:05:35 Sprecher 1

Ich erkläre mir das so ,n bisschen dadurch, dass es eigentlich dem Regulierer ziemlich egal ist, ob das Produkt nachher effizient entwickelt wird, ob ob wirklich die Kundenbedürfnisse befriedigt werden, Hauptsache es ist sicher und leistungsfähig.

00:05:48 Sprecher 1

Ist vielleicht ,n bisschen provokant, aber die Norm selber, die

00:05:53 Sprecher 1

62304 redet von Softwareanforderungen und Systemanforderungen.

00:05:57 Sprecher 1

Tatsächlich sagt, dass System und Softwareanforderungen auch gleich sein können, wenn es ist, falls das System ,ne reine Software ist, sagt aber irgendwie nirgendwo, der gibt keinerlei Hilfestellung, worin die sich unterscheiden.

00:06:08 Sprecher 1

Und das sehen wir tatsächlich in der Praxis, dass das auch bei Firmen sehr viel Probleme macht.

00:06:12 Sprecher 1

Also, ich, ich habe eben ja, ich muss ja im V.

00:06:15 Sprecher 1

Modell mehrere Ebenen von Anforderungen machen und dann schreib ich halt mal mehrere Ebenen, ohne wirklich ,ne klare Regel zu haben, was gehört wohin.

00:06:23 Sprecher 1

worin unterscheidet sich das, dann krieg ich auch auf der Testebene oft tatsächlich Probleme, weil ich mehrere Ebenen oder zumindest gedachte Ebenen am Ende mit den gleichen Tests abdecke.

00:06:33 Sprecher 1

Und leider, leider muss ich sagen, gibt die Norm da wenig Hilfe.

00:06:37 Sprecher 2

Ja, im Gegenteil, sie verwirrt vielleicht sogar an mancher Stelle, wenn wir gerade an die Traces zurückdenken von den Units, wieder ganz hoch auf die Anforderungen, dann überspringt es fast ,ne Ebene.

00:06:47 Sprecher 1

Ja, es gibt auch in dieser dieser Auflistung, ich muss die Anforderungen erheben, wenn

00:06:52 Sprecher 1

welche Arten von Anforderungen es gibt.

00:06:54 Sprecher 1

Da steht eben funktionale Anforderung und dann steht da Data Definition und dann steht da Interfaces und das sind eigentlich aus unserer Sicht auch Minimum 2 unterschiedliche Ebenen, über da über die da gesprochen wird.

00:07:05 Sprecher 2

Ja, wenn wir gerade über diese Softwareanforderung sprechen, können wir vielleicht aber ein bisschen reinzoomen und überlegen, wie können denn die Softwareanforderungen zur Architektur hin getraced werden.

00:07:17 Sprecher 2

Was wären so deine Tipps, geht es überhaupt?

00:07:20 Sprecher 2

Also, dass man

00:07:21 Sprecher 2

eine Softwareanforderung auf ein Architektur Item Merkmal irgendwie tracen kann oder wie soll das ganz konkret geschehen?

00:07:29 Sprecher 1

Es würde ja, es bedingt ja erstmal 2 Dinge.

00:07:30 Sprecher 1

Das eine ist, dass ich meine Softwareanforderung eindeutig identifiziere.

00:07:34 Sprecher 1

Ich glaub, das ist sehr einfach, die nummeriere ich durch oder die habe ich in in Excel in Spalten oder die habe ich oder in Zeilen, besser gesagt, das ist sehr einfach.

00:07:43 Sprecher 1

Das zweite ist, glaube ich, deutlich schwieriger, dass ich meine meine Architektur Entscheidungen in einzelne

00:07:51 Sprecher 1

identifizierbare Elemente runterbreche, das kann ich durchaus machen.

00:07:54 Sprecher 1

Also das, da, da nennt sich, wenn ich es auch wieder in Hardware denke, habe ich Bauteile und die geben kriegen einfach ,ne I.

00:07:59 Sprecher 1

D.

00:07:59 Sprecher 1

und dann ist das ,n Teil meiner meiner Systementwicklung, meiner Architektur.

00:08:03 Sprecher 1

Das, das wirkliche Tracen da drauf ist aber nicht so ganz einfach.

00:08:08 Sprecher 1

Also gibt es mehrere Überlegungen zu.

00:08:09 Sprecher 1

Das erste ist, für mich ist die Architektur eines Systems die Antwort auf die Summe aller Anforderungen.

00:08:16 Sprecher 1

Und wenn ich jetzt hingehe

00:08:18 Sprecher 1

Und ,ne, ich nenne es jetzt mal ,n 100% Traceability fordere.

00:08:21 Sprecher 1

Also jedes Architektur-Item, was ich irgendwie identifiziere, muss auf jedes, auf jede Systemanforderung getraced sein und umgedreht, dann krieg ich damit massiv Schwierigkeiten.

00:08:31 Sprecher 1

Also diese, ich nenne es immer die 100% Traceability, die brauche ich definitiv zwischen Anforderung und Tests, gar keine Frage, auf der Systemebene, also die Ebene, die ich dafür nutze, die Leistungsfähigkeit nachzuweisen.

00:08:44 Sprecher 1

Aber wenn ich das auf die Architekturebene mache, sehe ich mehrere Probleme.

00:08:47 Sprecher 1

Das erste ist, es gibt nicht funktionale Anforderungen, Performanceanforderungen, zum Beispiel auf welches Architektur-Item, auf welchen auf welche Zeile Code, auf welchen.

00:08:57 Sprecher 1

Widerstand trace ich ,ne Performanceanforderung.

00:09:00 Sprecher 1

Das System muss innerhalb von, keine Ahnung, 10 Millisekunden ein Ergebnis liefern.

00:09:03 Sprecher 1

Das, das geht nicht, weil die die Architektur als solche, die Gesamtarchitektur, solche Anforderungen befriedigt und umgedreht auch, wenn ich ,ne Architekturalem Datenbank zum Beispiel habe und wenn ich

das tracen möchte oder muss, sage ich, oh, ich muss ja alles tracen und dann suche ich mir alle Anforderungen, die irgendwas mit Persistierung zu tun haben, dann ist es aus meiner Sicht vollkommen sinnfrei, weil es hilft einfach nicht.

00:09:26 Sprecher 1

Das heißt, die die.

00:09:27 Sprecher 1

Forderung an der Stelle auf hundertprozentige Traceability würde ich auf keinen Fall aussprechen.

00:09:31 Sprecher 1

Sicherlich ist es sinnvoll, zum Beispiel, wenn ich irgendwelche Anforderungen an, ich muss einen bestimmten Algorithmus umsetzen und den direkten Trace in die Architektur habe, in diesem, in dieser Komponente ist es umgesetzt.

00:09:43 Sprecher 1

Das macht durchaus Sinn, aber auf keinen Fall so ,ne dogmatische, am Ende muss ich einen Trace Report erstellen, der nachweist, dass jede Komponente irgendwie ,ne Motivation hat, die sich unmittelbar aus einer

00:09:54 Sprecher 1

einzelnen identifizierbaren Anforderungen ableitet.

00:09:57 Sprecher 1

Und ich glaube auch, das wäre so der zweite Aspekt, dass das dieser ganzen Tätigkeit der Architektur ja gar nicht genüge tut.

00:10:05 Sprecher 1

Also ist ja fast beleidigend.

00:10:07 Sprecher 1

Also das würde bedeuten, dass ich durch eine einfache Traceability die Güte einer Architektur nachweisen kann.

00:10:12 Sprecher 1

Das, das geht nicht.

00:10:14 Sprecher 1

Also für mich ist das eigentlich die die Hauptwertschöpfung, die in einer gesamten Entwicklung steckt, ist, dass ich mir eine Architektur überlege und ich werde mitnichten in der Lage sein, die

00:10:23 Sprecher 1

Güte, diese Architektur irgendwie durch Traces auf Anforderung nachzuweisen.

00:10:28 Sprecher 1

Und dann hab ich so ,n 100% irgendwo so ,n grünen Haken und dann heißt es, ja, das ist ,ne super Architektur.

00:10:33 Sprecher 1

Das ist, glaub ich, das ist nicht sinnvoll.

00:10:36 Sprecher 1

Es ist auch keine valide Aussage.

00:10:38 Sprecher 1

Also, ob die Architektur funktioniert, seh ich erst am Ende, wenn ich das Produkt nach dieser Architektur gebaut habe.

00:10:42 Sprecher 1

Und dann hab ich ja wieder meine Anforderung an der Leistungsfähigkeit.

00:10:46 Sprecher 1

Über die ich das Ganze dann nachweisen kann.

00:10:48 Sprecher 1

Das heißt auch, auch so ein Trace wäre selbst ein Review der Anforderungen oder der Architektur.

00:10:53 Sprecher 1

Entschuldigung, durch einen Peer, durch einen zweiten Architekten ist eigentlich nichts mehr als ich nenne das mal eine qualifizierte Vermutung dass die Architektur gut sein.

00:11:02 Sprecher 1

wird und den wirklichen Nachweis, der dann auch regulatorisch relevant ist, den erbringe ich dann dadurch, dass ich die die Anforderungen, die zu dieser Architektur geführt haben, am fertigen Produkt tatsächlich teste.

00:11:14 Sprecher 1

Und bevor ich das nicht gemacht hab, geht das Produkt ja auch nicht raus an den Patienten.

00:11:18 Sprecher 1

Das heißt, bis dahin ist gar keine, ist gar keine Gefahr in Verzug und aus regulatorischer Sicht würde ich sagen, diese Traceability Anforderung zur Architektur oder die den Nachweis der Güte der Architektur, da sagt

00:11:29 Sprecher 1

Die 62 3 oder 4 selber kann ich durch Traceability kann ich durch Traceability machen.

00:11:34 Sprecher 1

Unsere Empfehlung wäre immer macht das mit einem ordentlichen Peer Review, aber auch das ist aus meiner Sicht ne ne Qualitätsmanagement Forderung, also ne Forderung an den.

00:11:43 Sprecher 1

Prozess und nicht ,ne Forderung an die Sicherheit des Produktes, weil ich werde das Produkt nicht an einen Patienten bringen, bevor ich nicht die Leistungsfähigkeit und zwar gegen die Ebene drüber getes-

tet habe.

00:11:53 Sprecher 2

Was wär dann die Konsequenz dessen für die Hersteller?

00:11:56 Sprecher 2

Würdest du empfehlen, dass sie dann auf dieses Thema Traceability zwischen Softwareanforderungen und Softwarearchitektur ganz verzichten?

00:12:03 Sprecher 1

Also, was ganz spannend ist, ist was funktioniert und mich fasziniert das auch noch, warum.

00:12:07 Sprecher 1

Ich hab es noch nicht ganz rausgefunden, dass man, dass ich aus der Architektur ergeben sich ja

00:12:12 Sprecher 1

weitere Anforderungen.

00:12:15 Sprecher 1

Also ,ne Architektur ist letztlich ,ne Entscheidung, die ich treffe, wie ich mein System aufbaue, beispielsweise.

00:12:20 Sprecher 1

Ich habe eine Anforderungsliste an ein System und ich sage, das System funktioniert, wenn ich diese Komponente habe, A.

00:12:26 Sprecher 1

B.

00:12:26 Sprecher 1

C., und die auf diese und diese Art und Weise miteinander interagieren.

00:12:29 Sprecher 1

Und das führt wiederum dazu, dass ich an diese Komponenten Anforderungen habe, jetzt aus Sicht.

00:12:35 Sprecher 1

Damit strukturiert mir eigentlich die Architektur, die die

00:12:39 Sprecher 1

die die nächste Anforderungsebene und und sehr oft funktioniert das sehr gut, dass ich da eben das Tracing über diese über diese Architekturebene zwischen den Anforderungen selbst erledige.

00:12:48 Sprecher 2

Und das wird uns direkt zur nächsten Frage, nämlich der Frage nach den Werkzeugen.

00:12:54 Sprecher 2

Was wären deiner Meinung nach Eigenschaften, Fähigkeiten, über die Werkzeuge verfügen sollten, da-

mit man damit ein gutes Tracing, sei es jetzt ein horizontales oder vertikales Tracing, hinbekommt?

00:13:07 Sprecher 1

Also es gibt

00:13:09 Sprecher 1

Also, wir sehen so 2 Arten von Tools, die aus 2 Richtungen kommen.

00:13:14 Sprecher 1

Und letztlich gibt es, ich sag mal, 3 Bereiche, die diese Tools abdecken oder 3 Herausforderungen.

00:13:19 Sprecher 1

Das eine ist das, was du gerade erwähnt hast, nämlich, ich muss nachweisen, dass mein Produkt so wie es ist, sicher und leistungsfähig ist.

00:13:27 Sprecher 1

Das heißt, ich habe Das, dass der Output dessen kommt in die Produktakte.

00:13:31 Sprecher 1

Dann gibt es den Bereich, ich muss auch nachweisen, dass ich einen konformen Prozess gelebt habe in der Zeit.

00:13:36 Sprecher 1

Wenn dann will ich ja alles aus einem Tool haben.

00:13:38 Sprecher 1

Das ist mehr Audit, ich nenne es mal Auditfestigkeit, das heißt, ich muss mit dem Output dieses Tools auch nachweisen, ja, ich habe ein Review gemacht, ja, ich habe es freigegeben und das muss auch mit Zeitstempel und gegebenenfalls sogar mit elektronischen Signaturen irgendwie machbar sein.

00:13:55 Sprecher 1

Und das Dritte jetzt

00:13:56 Sprecher 1

sehr nah an der an der Softwareentwicklung ist ja das Thema Kollaboration.

00:14:01 Sprecher 1

Also ich will in dem Tool sagen können, wer macht was, wie lange brauche ich, ich will vielleicht ,ne Sprintplanung, wenn ich agil arbeite, mit Burndown Charts und so weiter.

00:14:10 Sprecher 1

Und jetzt gibt es eigentlich 2 Ecken, aus denen solche Tools kommen können.

00:14:14 Sprecher 1

Die einen Tools kommen wirklich relativ hart aus der Entwicklung raus, die sind teilweise in der I.

00:14:19 Sprecher 1

de.

00:14:19 Sprecher 1

irgendwie integriert, die die sind sehr gut in dem Thema Kollaboration, die sind prinzipiell auch gut in so

00:14:25 Sprecher 1

Statusübergängen und so weiter.

00:14:27 Sprecher 1

Das heißt, ich kann damit auch schon so ,ne gewisse Prozesskonformität nachweisen, die sind aber oft nicht gut, wenn es darum geht, am Ende ein Dokument rauszuleiten, was auch wirklich in ,ne Akte packen kann.

00:14:40 Sprecher 1

Also das ist ,ne dringende Empfehlung, die wir haben, wenn man mit Tools arbeitet und damit Medizintechnik entwickelt.

00:14:44 Sprecher 1

Seht zu, dass ihr ,n PDF rauskriegt, sozusagen.

00:14:47 Sprecher 1

Der Hersteller freut sich, wenn er gleich für 15 Jahre ,ne Bestellung kriegt.

00:14:51 Sprecher 1

Aber weiß ich, was in 10 Jahren ist mit dem Hersteller, weiß ich, dass ich die Daten noch lesen kann.

00:14:55 Sprecher 1

Das heißt, ich brauch irgend ,ne Art Reporting, die ich in einem neutralen Tool machen kann und da ist im Moment eben P.

00:15:00 Sprecher 1

D.

00:15:00 Sprecher 1

F., würd ich mal sagen, das Tool der Wahl.

00:15:02 Sprecher 1

Also selbst X.

00:15:03 Sprecher 1

M.

00:15:03 Sprecher 1

L.

00:15:03 Sprecher 1

krieg ich meistens raus, aber das muss ich wieder formatieren und wenn ich dann ,n sehr strengen Q.

00:15:08 Sprecher 1

M.

00:15:08 Sprecher 1

hab oder auch nur ,n halbstrengen Q.

00:15:09 Sprecher 1

M., der will das dann alles validiert haben, sehr aufwendig.

00:15:12 Sprecher 1

Und aus dieser Richtung kommen so die Tools, die aus der Ecke Requirements Engineering kommen.

00:15:17 Sprecher 1

Ja, sie kommen sozusagen von oben.

00:15:18 Sprecher 1

Die können das sehr gut.

00:15:19 Sprecher 1

Die sind, die denken nämlich dokumentenorientiert.

00:15:21 Sprecher 1

Die denken ja auch, auch in in Output, den ich generiere.

00:15:26 Sprecher 1

Und oft sind die aber nicht so nah am Code und die anderen, die sehr nah am Code sind, sind dann nicht so nah an der an der Entwicklung.

00:15:34 Sprecher 1

Und da ist immer meine Empfehlung, wenn man sich für Tools entscheidet, die erste Frage: Was, was, was brauch ich denn?

00:15:39 Sprecher 1

Brauch ich so ,ne Art Deckel-Tool, was mir meine Dokumentation organisiert?

00:15:44 Sprecher 1

dann gucke ich eher in dieser Ebene Requirements Engineering oder brauche ich ,n Tool, wo wirklich tagtäglich permanent auch die Entwickler im Code mitarbeiten?

00:15:51 Sprecher 1

Dann sind wir so in den in den in den Tools, die wirklich sehr nah am Code sind und die kommen dann von den Herstellern der der Softwareprodukte.

00:15:57 Sprecher 1

Also das ist ein, das ist dann Team Foundation Server und diese ganzen Geschichten, die im im Micro-soft Umfeld beispielsweise laufen.

00:16:04 Sprecher 2

Wo würdest du deine Produkte, also die Methodeprodukte wie Riskpack und Bedpack verorten?

00:16:10 Sprecher 1

Also wir haben ja genau diesen,

00:16:12 Sprecher 1

Wir bauen nicht alles von Scratch.

00:16:14 Sprecher 1

Wir basieren auf einem Tool.

00:16:15 Sprecher 1

Das ist das Tool Polarion, was mittlerweile von Siemens übernommen wurde.

00:16:19 Sprecher 1

Genau aus dem Grund, weil wir das sehr, sehr mächtig halten in dem Bereich Dokumentation und.

00:16:25 Sprecher 1

wo man eben gerade auch ja Varianten mit abbilden kann, irgendwelche Branches tatsächlich von Dokumenten machen kann, die sehr codenah sind.

00:16:32 Sprecher 1

Da kann auch jemand mit arbeiten, der noch nie Software entwickelt hat.

00:16:35 Sprecher 1

Das ist also in Teilen sehr wordähnlich, hat aber trotzdem ,ne Datenbank dahinter, die eben diese ganzen Traceability Analysen und so weiter ermöglicht.

00:16:44 Sprecher 1

Und es geht auch Es kann auch sehr tief in Code reingehen, aber auf einer Ebene, die man eigentlich gar nicht regulatorisch gar nicht braucht.

00:16:52 Sprecher 1

Also sehr wenige unserer Kunden nutzen

00:16:55 Sprecher 1

Polarion selber, um bis in den Code reinzugehen, haben da teilweise aber Tools wie Jira, um einfach mal 1 zu nennen, wo sie ihre ganzes Teamverwaltung, das Thema Kollaboration, machen und was man dann macht, ist dass man eine Ebene finden muss und das geht oft sehr gut, in denen sich diese Tools dann finden.

00:17:12 Sprecher 1

Das heißt, ich habe eine eine Informationsebene, die ich bidirektional synchronisieren kann.

00:17:16 Sprecher 1

Das geht tatsächlich auch sehr, sehr einfach und sehr gut.

00:17:20 Sprecher 1

Und dann gehen beispielsweise die in Jura gefundenen Bugs, die gehen direkt wieder in Polarion, damit sie zur Dokumentation und weiteren Verarbeitung bereitstehen.

00:17:28 Sprecher 1

hat man so ein bisschen das Beste aus beiden Welten und man nimmt den Leuten nicht ihre gewohnten Werkzeuge weg, die mit denen sie eben schon sehr lange arbeiten.

00:17:36 Sprecher 1

Was, was, glaube ich, der falscheste Ansatz ist, den man beim Tooling machen kann, weil vielleicht ein Nebensatz, ob ein Tool erfolgreich ist oder nicht, das ist unsere Erfahrung in den weiß ich über 10 Jahren, in denen wir das jetzt machen, hängt sehr wenig an den Features oder an

00:17:49 Sprecher 1

dem konformen Output, sondern in allererster Linie daran, ob ich es schaffe, die Menschen mitzunehmen, mit diesem Tool zu arbeiten.

00:17:55 Sprecher 1

Und das ist, glaube ich, was, was unterschätzt wird in der Praxis.

00:17:58 Sprecher 2

Ja, Sven, das ist eigentlich schon ein wunderschönes Schlusswort.

00:18:01 Sprecher 2

Ich glaube, wir haben ziemlich viel abgedeckt jetzt in diesem Podcast, aber natürlich wird es noch weitere Fragen geben.

00:18:07 Sprecher 2

Vielleicht ,n Tipp dazu, Sven und ich haben einen Artikel geschrieben, einen Fachartikel bei uns auf der Webseite, den verlinken wir unten auch wieder.

00:18:17 Sprecher 2

Und ich denke, du wirst keine Schwierigkeiten damit haben, wenn wir empfehlen, dass all diejenigen, die sich für Werkzeuge, den Umgang mit Werkzeugen, das Abbilden der Traceability mit Werkzeugen dann an dich denken.

00:18:30 Sprecher 2

Und wenn du einverstanden bist, Sven, würden wir eure Kontaktdaten auch unten noch verlinken.

00:18:35 Sprecher 1

Ja, sehr gerne natürlich.

00:18:36 Sprecher 1

Ich freue mich.

00:18:37 Sprecher 2

Ja, dann bleib mir nur ganz herzlichen Dank sagen, Sven.

00:18:40 Sprecher 2

Bis sehr bald.

00:18:41 Sprecher 1

Dankeschön und Tschüss.

