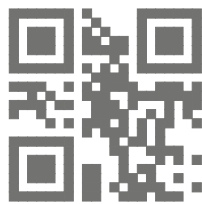


*Wie Sie mit Methoden des
Systems Engineering eine
lückenlose Rückverfolgbarkeit
in PLM und ALM sicherstellen*

Traceability over Lifetime





Warm up

Vielleicht kennen Sie das: Ein Kind, neugierig auf die Welt, löchert seine Eltern bis ins Endlose, entweder mit einem unschuldigen „Warum?“, oder aber einem „Und dann?“.

Das Kind ist damit ein prima Beispiel dafür, was Traceability bedeutet: Ausgehend von einem Produkt, das vor ihm steht, möchte es einfach über alles Bescheid wissen: Bestandteile und deren Funktionen, und auch die Gründe, warum manche Dinge so sind wie sie sind – bis es an dem Punkt ankommt, an dem alle wesentlichen Fragen beantwortet sind. Richten Sie sich darauf ein, dass das im Diskurs mit Ihrem Kind seehr lange dauern wird.

In umgekehrter Richtung bedeutet Traceability, ausgehend von einer Anforderung begründen zu können, wie diese Anforderung in einem Produkt konkret realisiert wird – auch hier bis zu dem Punkt, an dem die Antwort (=Implementierung der Anforderung) eindeutig gegeben ist.

In beiden Richtungen, sowohl vorwärts als auch rückwärts, kann es je nach Produkt und Kontext erforderlich sein, diese Traceability „over Lifetime“ zu gewährleisten, also vom Anforderungsmanagement, über die Entwicklung, die Produktion, die Anwendung im Feld, bis zur Entsorgung.

Durch dieses Bedürfnis nach Transparenz in alle Richtungen kann Traceability sehr schnell sehr ressourcenintensiv werden und erfordert eine hohe Disziplin. Wir erleben dennoch, dass das Thema Traceability in jüngster Vergangenheit an Bedeutung gewonnen hat und verzeichnen eine verstärkte Sensibilität unserer Kunden, die sich auf zwei wesentliche Einflussfaktoren zurückführen lässt:

Zunehmender Anteil an Software in Produkten, welche die Gesamtkomplexität erhöht. Daraus folgt, dass die Nachvollziehbarkeit, wie sich ein Produkt in jedem Betriebszustand verhält, ebenfalls immer anspruchsvoller wird.

Verschärfte regulatorische und normative Forderungen nach eben jener Nachvollziehbarkeit. Teilweise sind diese direkte Folge des steigenden Software-Anteils, da Normengeber erkennen, dass bestehende Regelungen nicht mehr der Komplexität softwaregetriebener Produkte gerecht werden.

➤ Mit diesem Papier wollen wir Ihnen eine Orientierung geben, worauf es bei der Traceability ankommt, und wie Sie mit Methoden des Systems Engineering eine **lückenlose Rückverfolgbarkeit in Ihren PLM- und ALM-Systemen** sicherstellen können.



Kernfähigkeiten einer durchgängigen Traceability

Bevor wir uns den Lösungsansätzen zum Herstellen einer durchgehenden Traceability für softwaregetriebene Produkte widmen, schauen wir zunächst auf die hierfür notwendigen Kernfähigkeiten. Mit Fähigkeit ist gemeint, was Ihr Unternehmen grundsätzlich, also ohne Berücksichtigung eines konkreten Lösungsansatzes, können sollte, um eine Traceability zu erreichen (siehe Bild unten auf dieser Seite).

Aber nicht nur ganz global „was“ das Unternehmen können muss, sondern auch „wann“, beziehungsweise über welchen Abschnitt des gesamten Lifecycles es dies muss, sollte Berücksichtigung finden. Also ob Rückverfolgbarkeit in der Anforderungsanalyse bzw. im Requirements Engineering, der eigentlichen Entwicklung, im Rahmen der Wertschöpfungs-

prozesse, oder sogar bis hinein in die Phase der tatsächlichen Nutzung des Produkts benötigt wird.

Für die Traceability ist es ein großer Unterschied, ob zum Nachweis der Anforderungserfüllung ...

... **lediglich der Produktentwurf und die Befähigung der Produktionsprozesse** abgesichert werden (Beispiel einfache Massenartikel wie z.B. Kochgeschirr, bei denen in der Produktion allenfalls Stichproben geprüft werden, die aber nicht auf die einzelne Produktinstanz rückverfolgbar sein müssen),

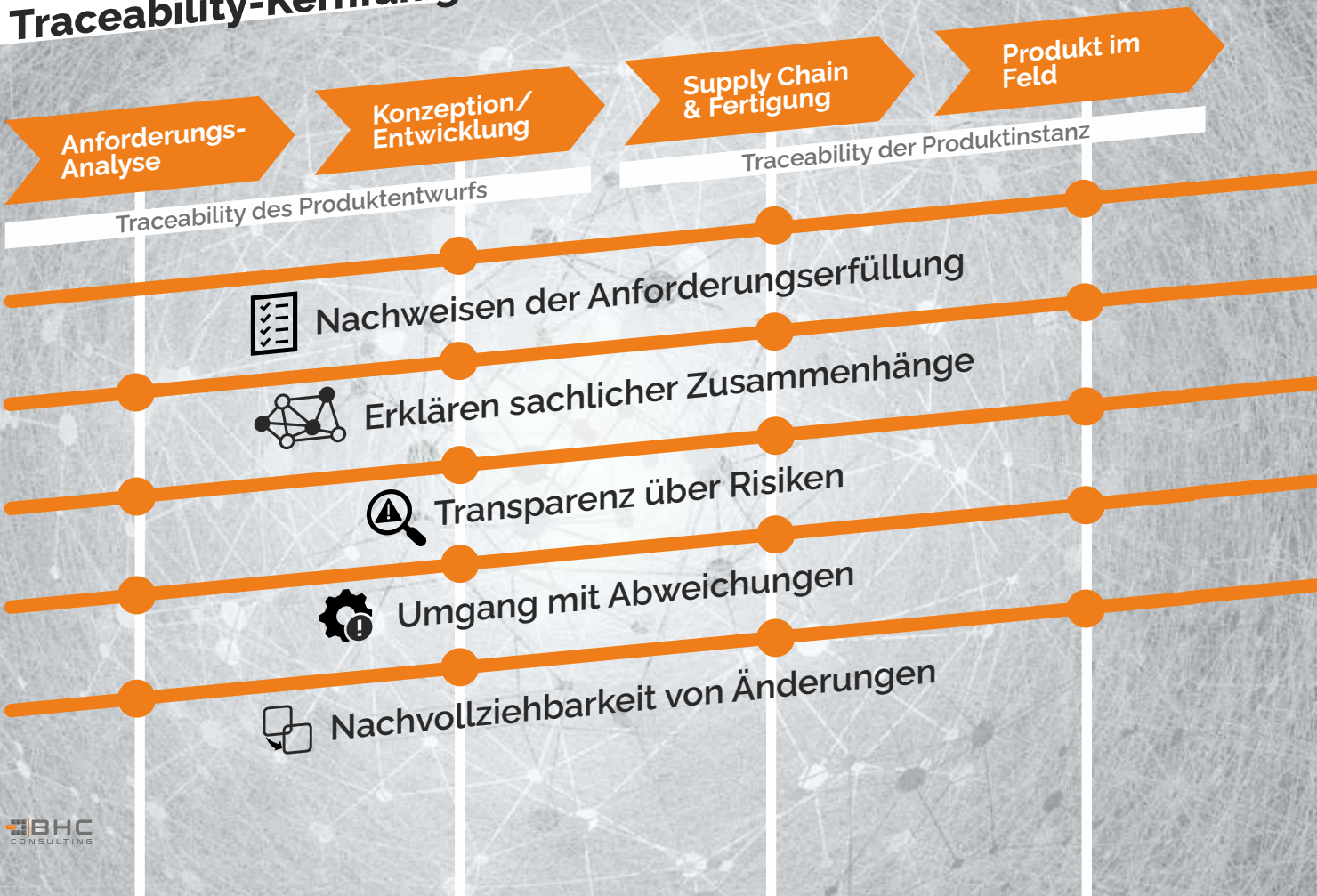
... eine **rückverfolgbare 100%-Prüfung** auf jede gefertigte Produktinstanz erfolgt (bspw. hochwertige E-Bikes, bei denen über eine Seriennummer für jedes einzelne Fahr-

rad ein Prüfprotokoll dokumentiert wird),

...oder ob **jede Produktinstanz auch bis hinein ins Feld auf Konformität überwacht wird** (bspw. Medizintechnikgeräte oder Bauteile in der Kraftwerkstechnik, für die ein Hersteller auch für den Einsatz im Feld Verantwortung übernimmt).

Je nachdem, in welcher Breite und Tiefe Traceability gefordert oder gewünscht ist, müssen diese Fähigkeiten in entsprechendem Maß ausgeprägt sein. Hintergrund dafür können sowohl regulatorische Aspekte sein, die Sie als Hersteller erfüllen müssen, als auch wirtschaftliche oder qualitative Erwägungen.

Traceability-Kernfähigkeiten entlang des Product Lifecycles





Nachweisen der Anforderungserfüllung

Die Fähigkeit, als Ingenieur die Erfüllung gestellter Anforderungen nachweisen zu können, sollte selbstverständlich sein. Nicht umsonst gibt es schon seit jeher in der Produktentwicklung Lasten- und Pflichtenhefte, sowie (wenigstens rudimentäre) Mechanismen in den Entwicklungsabteilungen, die auf die Anforderungserfüllung achten.

Als wesentlicher Baustein dient dabei immer ein Testmanagement, durch das die gestellten Anforderungen auch empirisch belegt werden. Das erstreckt sich mindestens auf die Entwicklungs-

absicherung anhand von Prototypen (Traceability des Produktentwurfs), kann aber auch darüber hinausgehen. Stichprobenartige Kontrollen oder in manchen Fällen sogar 100%-Kontrollen der tatsächlich gefertigten Produkte (Traceability der Produktinstanz) sind im Rahmen von Qualitätssicherungsmaßnahmen keine Seltenheit.

Eine häufig zu beobachtende Schwachstelle ist jedoch die Zuordenbarkeit von Testfällen zu Anforderungen und umgekehrt. Während das Durchführen von Tests zur Absicherung Stand der Technik ist, ist die Transparenz darüber, inwiefern die gesetzten Anforderun-

gen von Tests und Absicherungsmaßnahmen abgedeckt sind, manchmal sehr getrübt. Die Traceability ist dann nicht durchgängig. Gründe für diese Schwachstelle sind vielfältig: Angefangen bei nicht einzeln adressierbaren Anforderungen, die in Powerpoints und Word-Dokumenten stecken, über ein nicht Ende-zu-Ende („E2E“) durchgängiges Management von Produktvarianten bis hin zu nicht miteinander integrierten IT-Tools, in denen all diese verschiedenen Aspekte gespeichert werden, womöglich in nicht konsistent zueinander passenden Versionsständen und unterschiedlichen Datenmodellen.



Erklären sachlicher Zusammenhänge

In der Praxis ist die Entwicklung eines Produktes kein streng geradliniger Prozess, sondern eine Vorgehensweise, bei der man sich iterativ den Top-Level-Anforderungen stufenweise weiter annähert. Das galt schon für klassische Wasserfall-Vorgehensmodelle, für agile Arbeitsweisen gilt das freilich noch viel mehr.

Während jedoch ohne Anspruch auf durchgängige Traceability dies oft Vorgänge sind, die nur von den beteiligten Ingenieuren aufgrund ihres Wissens nachvollzogen werden können, ist es mit Anspruch auf Traceability nun elementar, dass dieses Wissen auch in einer auswertbaren Form so granular wie für die geforderte Tiefe der Nachvollziehbarkeit nötig dokumentiert wird.

Vielleicht erinnern Sie sich an das Fallbeispiel ‚Elektrisches Speichenschloss eines E-Bikes‘ aus unserem [Whitepaper Softwaregetriebene Produkte: Organisational Readiness und Methodische Grundlagen](#)? Dort ging es darum, wie sich die verschiedenen Entwicklungsdomänen Elektronik, Mechanik und Software synchronisieren können, um im Ergebnis ein konsistentes und funktionierendes Produkt auf den Weg zu bringen. Im Fallbeispiel entriegelt sich das elektromagnetische Speichenschloss bei Annäherung des Besitzers automatisch.

Spinnen wir dieses Beispiel unter dem

Aspekt der Traceability-Fähigkeiten „Nachweis der Anforderungserfüllung“ und „Erklären sachlicher Zusammenhänge“ weiter.

In einer Welt ohne Traceability-Erfordernis würde es ausreichen, wenn das Ergebnis der Entwicklung dokumentiert ist: also bspw. eine Stückliste, in der alle Elemente, aus denen das Produkt schlussendlich besteht, aufgeführt sind. Baut man diese Stückliste zusammen, erhält man ein funktionierendes Produkt. Die Stückliste gibt aber keine Antwort darauf, warum und wie das Produkt funktioniert. Eine einzelne Stücklistenposition (z.B. der Motor des elektrischen Speichenschlosses) „weiß“ nicht, welchen Zweck sie erfüllt. Bei einfachen Produkten und Bauteilen mag sich der Zweck einem Wissenden zwar auf den ersten Blick erschließen. Umso komplexer das Produkt und seine Zusammenhänge, desto weniger zugänglich wird diese Klarheit allerdings. Oder hätten Sie als Außenstehender mit Blick auf die Stückliste auf Anhieb sagen können, dass die Bluetooth-Antenne auch der Näherungserkennung des Fahrers (bzw. seines Handys) und damit der Entriegelung des Speichenschlosses dient? Oder dass ein im Speichenschloss verbauter Magnetfeldsensor Bewegung erkennt und somit eine Verriegelung während der Fahrt verhindert? Eine durchgängige Traceability kann darauf eine Antwort liefern.

Seite 6

Seite 6

Seite 7

...Lösungsideen ab Seite 8

...Implementierung ab Seite 14



Transparenz über Risiken

Nichts ist hundertprozentig. Dieser Binsenweisheit, die man im Alltag oft als allgemeines Lebensrisiko durchgehen lassen kann, kommt im Ingenieurwesen eine besondere Bedeutung zu. Dass jede Anforderung unter allen Umständen immer erfüllt ist, und sich keine Fehler einschleichen, lässt sich schlicht nicht mit absoluter Sicherheit erfüllen.

Ob dieser Umstand nun besonders kritisch ist, oder nicht, lässt sich nicht pauschal beantworten. Es hängt vom Einzelfall ab. Dass die Anforderung „Schutz des Fahrrads gegen unberechtigte Mitnahme“ durch ein einfaches Speichenschloss bei ausreichend krimineller Energie des Übeltäters nicht wirklich erfüllt

werden kann, wiegt weniger schwer, als die Anforderung, dass das Produkt Leib und Leben seines Benutzers nicht gefährden soll. Das Durchführen von Risikoanalysen, die Identifizierung von Fehlermöglichkeiten, deren Bewertung hinsichtlich Auswirkungen und das Definieren von Gegenmaßnahmen gehören daher zur zwingenden Praxis jeder Entwicklung.

Traceability kommt dabei vor allem in zweierlei Hinsicht eine wichtige Bedeutung zu:

Nachvollziehbare funktionale Zusammenhänge helfen, Risiken zu identifizieren und diese zu bewerten

Nachvollziehbar dokumentierte Risikobewertungen helfen, die eigene Sorgfaltspflicht nachzuweisen.

Setzen Sie sich in die Lage des verantwortlichen E-Bike-Konstrukteurs, sollte eines Tages die Staatsanwaltschaft anklopfen. In den Bergen ist ein Fahrer Ihres Rades bei Tempo 50 bergab verunglückt, weil passiert ist, was nicht passieren soll: Das elektrische Speichenschloss hat sich mitten in der Fahrt verriegelt. Können Sie auf Anhieb versichern nach bester Ingenieurspraxis alle Risiken bewertet und entsprechend gehandelt zu haben? Wenn ihre Prozesse in Bezug auf Traceability gut ausgeprägt sind, wird das kein Problem sein.

Setzen Sie sich in die Lage des verantwortlichen E-Bike-Konstrukteurs, sollte eines Tages die Staatsanwaltschaft anklopfen. In den Bergen ist ein Fahrer Ihres Rades bei Tempo 50 bergab verunglückt, weil passiert ist, was nicht passieren soll: Das elektrische Speichenschloss hat sich mitten in der Fahrt verriegelt. Können Sie auf Anhieb versichern nach bester Ingenieurspraxis alle Risiken bewertet und entsprechend gehandelt zu haben? Wenn ihre Prozesse in Bezug auf Traceability gut ausgeprägt sind, wird das kein Problem sein.



Umgang mit Abweichungen

Risikoanalysen werden durchgeführt, noch bevor ein Produkt überhaupt auf den Markt kommt. Trotz aller Sorgfalt in dieser Phase ist es dennoch wahrscheinlich, dass das fertige Produkt nicht vollständig fehlerfrei ist. Geschuldet ist dies auch der sehr hohen Komplexität softwaregetriebener Produkte. Aus Sicht einer durchgehenden Traceability ist deshalb nicht nur relevant, wie mit Risiken präventiv, also bevor das Produkt zum Kunden kommt, umgegangen wird, sondern auch, wie neue Erkenntnisse über Fehler und damit Risiken in angemessener Weise bewertet werden, und zu welchen Maßnahmen dies führt.

Ein weiteres Beispiel: das E-Bike-Display eines einzelnen Kunden ist ausgefallen. Das kann auf einen dummen Zufall, Kundenverschulden (z.B. Sturz) oder aber auch einen systematischen Fehler in der Leiterplattenfertigung oder gar einen Software-Fehler zurückzuführen sein. In einem Einzelfall wird man dem Kunden eher auf Kulanz ein neues Display anbieten und nicht unbedingt auf systematische Ursachenforschung gehen. Spätestens wenn sich dieser Fehler statistisch häuft, sollte das aus wirtschaftlichem Eigeninteresse des Herstellers aber nachgeholt werden.

Ganz anders könnte es dagegen beim Fall des in den Bergen verunglückten Radfahrers aussehen: Sie konnten vielleicht nachweisen, dass Sie im Rahmen der Entwicklung alle nötige Sorgfalt haben walten lassen, indem Sie entsprechende Risikoanalysen durchgeführt hatten. Stellen Sie sich aber vor, die Ermittler finden heraus, dass dies dennoch schon der dritte ähnlich gelagerte Fall aus der Praxis ist, Sie also irgendein Risiko übersehen haben müssen. Besonders ungünstig wäre dann, wenn dieses wiederholte Vorkommnis (=Abweichung vom Sollzustand) Ihrem Unternehmen zwar bekannt geworden ist, mangels entsprechender Prozesse aber keine Gegenmaßnahmen (z.B. weitere Analysen oder Rückrufe) eingeleitet wurden.

Traceability in diesem Fall bedeutet also auch, dass Abweichungen (Fehler, Reklamationen, etc.) erfasst, und vor allem nachvollziehbar behandelt werden.



Nachvollziehbarkeit von Änderungen

Verfolgen wir das Fallbeispiel noch etwas weiter. Durch tieferegehende Analysen wird herausgefunden, dass im elektrischen Speichenschloss aus eigener Produktion zur Fahrterkennung ein mangelhafter Magnetfeldsensor verbaut war. Dieser sollte eigentlich Bewegung des Rades detektieren, und ein Aktivieren des Schlosses während der Fahrt verhindern. Außerdem kommt heraus, dass in einer früheren Ausführung des E-Bikes ein Sensor eines anderen Herstellers eingesetzt wurde, und alle bislang bekannten Fehlfunktionen des Schlosses betreffen nur diesen neuen Sensor.

Der zuständige Ingenieur, der seinerzeit die Änderung des Sensors verantwortet hat, hat das Unternehmen leider verlassen und kann zur internen Aufarbeitung nicht mehr beitragen – die Hoffnung der versammelten Entwicklungs-

mannschaft liegt daher auf einer sauberen Änderungsdokumentation.

Glücklicherweise haben Sie in Ihrem Unternehmen Traceability schon immer ernst genommen, und siehe da: aus der Dokumentation geht hervor, dass der ursprüngliche Sensor nicht mehr lieferbar ist und der Ersatz auf dem Papier eine identische Spezifikation hat. Die durchgeführten Qualitätssicherungsmaßnahmen und Tests verliefen alle positiv, sodass einem Einsatz zumindest aus damaliger Sicht nichts entgegensprach. Ihren Sorgfaltspflichten sind Sie also nachweisbar nachgekommen.

Das mag dem verunfallten Fahrradfahrer zwar nicht helfen, der Fall der Fälle ist schließlich eingetreten und als Hersteller werden Sie zivilrechtlich dennoch in der Haftung stehen. Aus straf- und ordnungsrechtlicher Sicht wird man aber nicht in gleichem

Maße belangt werden, und auch auf Ansprüche gegen Versicherungen und Lieferanten dürfte sich eine eindeutige Nachweisführung positiv auswirken.

Man könnte nun meinen, Traceability wäre nur dazu da, um im Fall der Fälle nachweisen zu können, alles richtig gemacht zu haben. Das ist eine Sichtweise, eine andere Perspektive kann aber auch diese sein: Traceability hilft dabei, die Qualität Ihres Produktes zu erhöhen, gerade weil der Fokus auf Transparenz und Nachvollziehbarkeit liegt und somit zu einem besseren Verständnis des Produktes und seiner Schwachstellen führt. Oder anders formuliert: Während der geschilderte Fall des während der Fahrt aktivierten Speichenschlosses leider **trotz** aller Traceability passiert ist, haben Sie gleichzeitig **wegen** Ihrer Sorgfalt viele andere Fehler und Schäden von vornherein verhindert, die sonst zu noch viel größerem Schlamassel geführt hätten.

Quick-Check: Wie gut ist die Traceability in meinem Unternehmen?

Note
Relevanz

	Note	Relevanz
Sind allen Beteiligten die Anforderungen und ihr Geltungsbereich bekannt? Kann für jedes gefertigte Produkt nachgewiesen werden, dass diese erfüllt und abgesichert sind?		
Kann für jede Anforderung nachvollzogen werden wie diese implementiert wird – bei softwaregetriebenen Produkten bei Bedarf auch bis auf die Ebene des Quellcodes?		
Werden Risiken des Produkts systematisch identifiziert? Können für die Risiken Bewertungen und risikominimierende Maßnahmen aufgezeigt werden?		
Werden Fehler, Reklamationen und Abweichungen systematisch erfasst? Kann gezeigt werden, wie diese regelmäßig im Rahmen von Produktverbesserungen berücksichtigt sind?		
Können Änderungsstände eindeutig identifiziert werden? Ist für jede Änderung nachvollziehbar, warum diese durchgeführt wurde, und welche Auswirkungen sie hat?		
Ist die Traceability insgesamt prozesssicher (das heißt unter anderem, nicht abhängig vom Kopfwissen Einzelner)?		

Machen Sie den Selbsttest: Entscheiden Sie, wie wichtig die verschiedenen Aspekte in Ihrem Unternehmen sind, und vergeben Sie eine Bewertung (zum Beispiel Schulnoten von 1-6).

Lösungsideen

Traceability für Ihren Product Lifecycle

1 | Produktarchitektur à la Systems Engineering

Wie aber installiert man eine durchgängige Traceability im Unternehmen? Wie so oft, liegt die Antwort in mehreren Dimensionen, die nicht vollständig losgelöst voneinander betrachtet werden können: Prozess, Methode und Tool.

In Bezug auf die Methode bietet es sich an, Konzepte aus dem Systems Engineering einzusetzen. Bewährt hat es sich, entlang des V-Modells die Sicht auf das Produkt in wenigstens folgende Schichten aufzuteilen:

- Anforderungen 
- Funktionale Beschreibung und 
- Implementierung 

Ausgehend von den Anforderungen wird das Produkt im Rahmen des Entwicklungs-

prozesses dabei stetig weiter ausgeprägt. Ein iteratives Vorgehen ist dabei durchaus üblich. Der Erfolgsfaktor für eine durchgängige Traceability ist nun, in jeder Schicht die wesentlichen Informationen einzeln zu identifizieren, als solche zu erfassen und ihre Beziehungen und Abhängigkeiten zueinander zu dokumentieren.

Im Fall des elektrischen Speichenschlosses könnte das am Beispiel der automatischen Entriegelungs-Funktion auszugswise wie im Bild unten auf der Seite aussehen.

In dieser Dokumentationsform wird ersichtlich...

... welchen Zweck ein einzelnes Bauteil erfüllt. Über die modellierten Verbindungen lässt sich zurückverfolgen, dass die Bluetooth-Antenne auch dazu dient, die Anwesenheit des Besitzers zu erkennen, um so die Ver-

und Entriegelung anzusteuern.

... wie eine Anforderung konkret implementiert wird, und zwar sowohl aus einer funktionalen Betrachtung, als auch ganz konkret in der tatsächlichen Implementierung. Im Beispiel wird die Anforderung „Komfort-Diebstahlschutz“ realisiert durch ein Zusammenwirken der Komponenten Fahrradcomputer und elektrisches Speichenschloss.

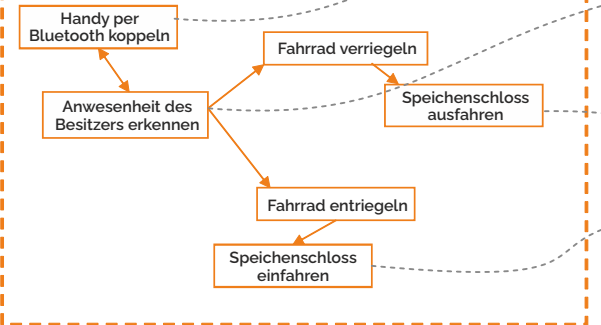
Das ist natürlich nur ein Ausschnitt aus einer vollständigen Dokumentation eines komplexen Produkts. In der Realität wird eine Anforderung oft durch eine größere Zahl von Funktionen realisiert, und auch die Detaillierung kann noch deutlich tiefer gehen.

Anforderung

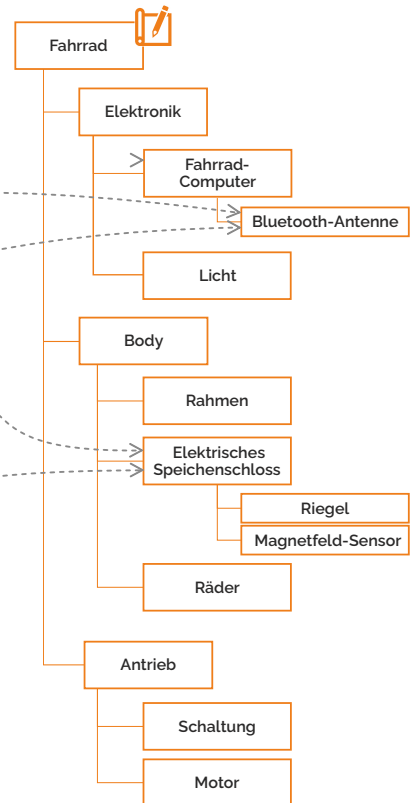
R
Komfort-Diebstahl-Schutz
Das Fahrrad soll automatisch ver-/entriegelt werden, wenn sich der Besitzer entfernt/nähert.


Funktionale Beschreibung


Funktion: Automatisch ver-/entriegeln



Implementierung



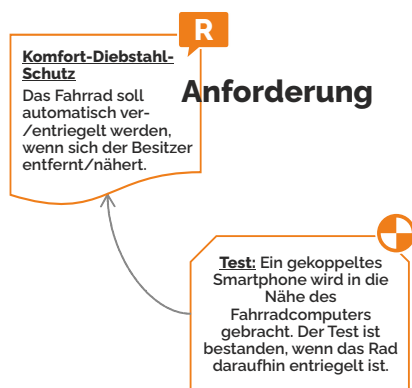
 Erklären sachlicher Zusammenhänge

 Mit Methoden des Systems Engineering kann eine Traceability in Bezug auf funktionale und sachliche Zusammenhänge eindeutig hergestellt werden.

2 | Verknüpfen Sie Tests mit den Elementen Ihrer Produktarchitektur

Kommen wir zum nächsten Punkt, dem Nachweis der Anforderungserfüllung. Im Sinne der Traceability ist dabei als erstes wichtig, überhaupt zu wissen, woher eine konkrete Anforderung eigentlich stammt – also zum Beispiel, ob es sich um eine Kundenanforderung oder um eine gesetzliche Vorgabe handelt. Es ist schließlich ein Unterschied, ob der Nachweis gegenüber einer prüfenden Behörde (z.B. durch Prüfprotokolle) oder gegenüber einem kritischen Kunden (v.a. durch ein gutes Produkt) erbracht werden muss.

In jedem Fall bedarf es aber einer Absicherungsstrategie, die in der Regel durch Tests gestützt wird. Für die Traceability wiederum ist wichtig, die Testdefinitionen eindeutig den Objekten zuzuordnen, für die sie eine Aussage treffen. Das ist leichter gesagt als getan. Stellen wir uns vor, wir wollen den Komfort-Diebstahl-Schutz testen. Aus Sicht der Anforderung könnte eine Testdefinition wie folgt lauten und wäre mit der zugehörigen Anforderung zu verknüpfen:



Eine solche Anforderung im Produkt zu realisieren ist allerdings nicht ganz trivial. Es erfordert oft ein Zusammenwirken verschiedener Einzelfunktionen, die wiederum von mehreren Bauteilen implementiert werden (siehe Darstellung auf Seite 8). Um daher die Anforderung nicht ausschließlich als Ganzes testen zu müssen, sondern arbeitsteilig auch einzelne Lösungsbestandteile getrennt voneinander bewerten zu können, wird es Tests für weitere Teilaspekte geben – beispielsweise einen Test, der sich allein auf die Frage konzentriert, ob der Riegel des Speichen-

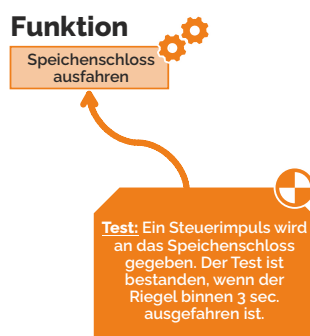
schlosses korrekt ausfährt, wenn er angesteuert wird.

Eine wesentliche Frage ist deshalb, mit welchem Objekt dieser spezialisierte Test verknüpft wird, und eigentlich sollte ein Test immer in Bezug zu einer Anforderung stehen. Im gezeigten Beispiel käme dafür nur die Anforderung des Diebstahl-Schutzes als Ganzes in Frage. Das wiederum wäre aber auch ungünstig, denn der Test repräsentiert dann nicht die ganze Anforderung und wäre so in seiner Aussagekraft an dieser Stelle viel zu spezifisch:

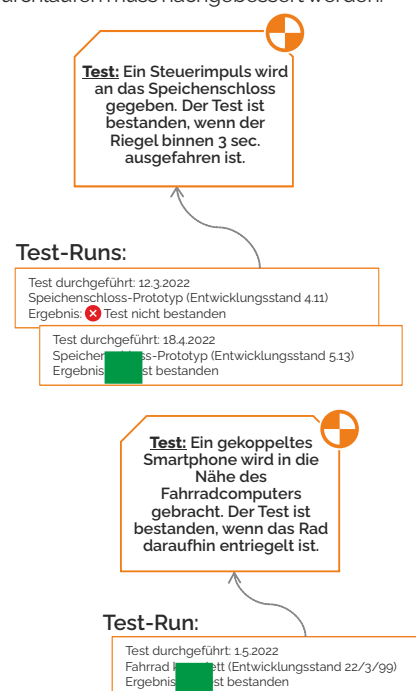


Besser wäre es, den Test mit der Funktion „Speichenschloss ausfahren“ zu verknüpfen. Ist der Test erfolgreich, wird eine Aussage genau über diese Funktion getroffen. Nicht mehr, und nicht weniger:

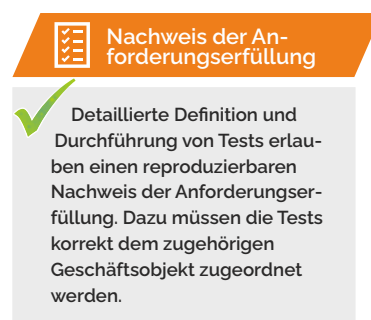
... besser so:

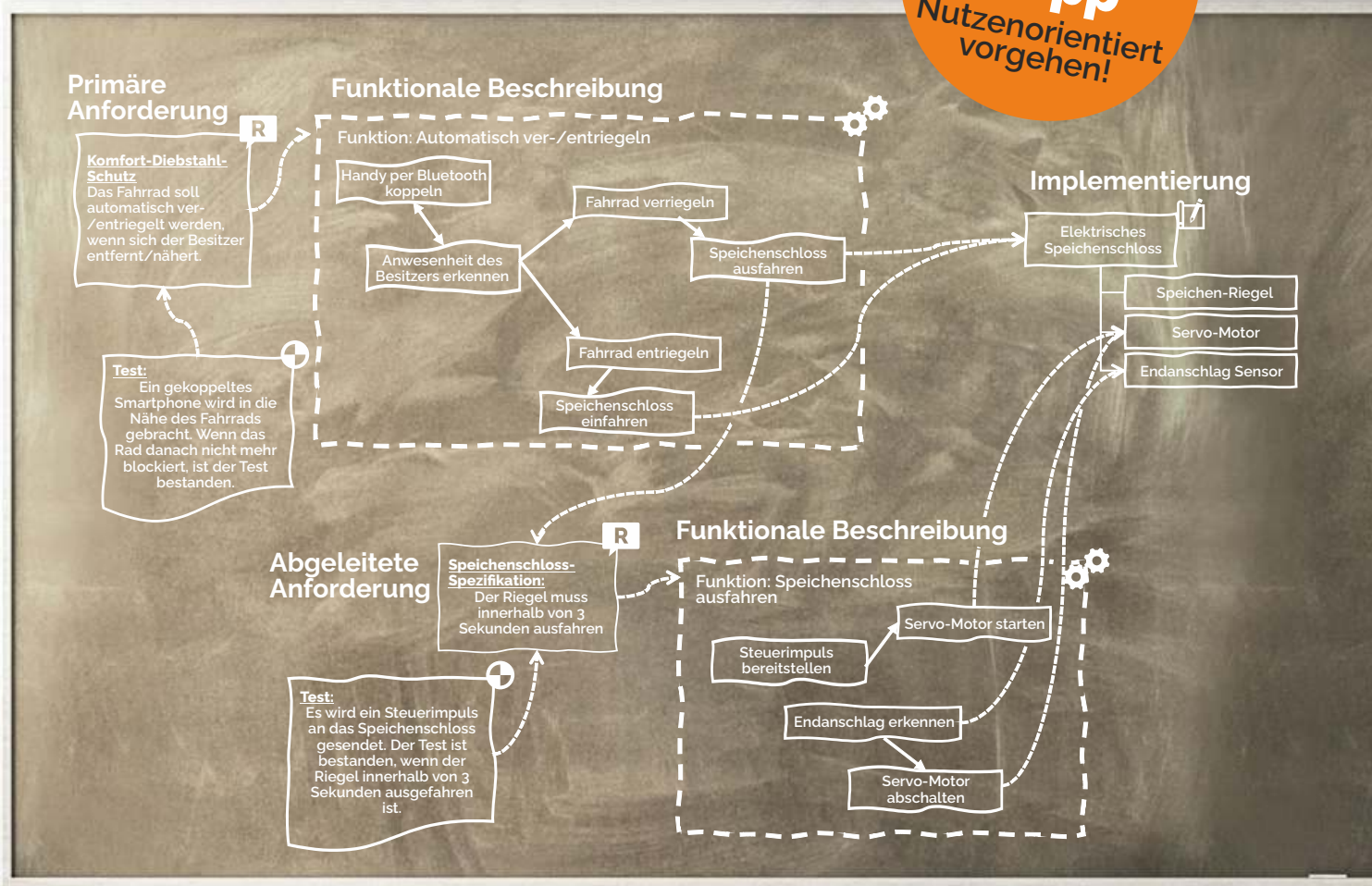


Wenn nun jede Testausführung (Test-Runs, Prüfprotokolle) sauber und in Bezug auf die Testbeschreibung dokumentiert wird, kann der Nachweis jeder Anforderung nachvollziehbar erbracht werden. Denn die Testbeschreibungen selbst sind ja mit den Anforderungen verknüpft, und ein erfolgreich durchgeführter Test erlaubt Rückschlüsse auf die Anforderungserfüllung. Bei nicht erfolgreichen Testdurchläufen muss nachgebessert werden.



Der Test-Run muss dabei Bezug darauf nehmen, mit welchen konkreten Änderungen bzw. Entwicklungsständen der einzelnen Produkt-Bestandteile er durchgeführt worden ist. Ansonsten ginge ein wichtiger Teil der Nachvollziehbarkeit und damit der Aussagekraft bezüglich der Anforderungserfüllung verloren.





Tests als Anforderungen?

Die auf der vorherigen Seite beschriebene Vorgehensweise wäre zumindest mit einem konservativen Blick auf Systems Engineering (z.B. à la A-SPICE) nicht ganz sauber – schließlich sollte ein Test immer aus einer Anforderung abgeleitet werden. Umgekehrt ausgedrückt bekommt der Test in unserem Beispiel damit den Rang einer Anforderung, da er zu der Stelle wird, an der die zu erfüllenden Eigenschaften beschrieben werden. Konservativ betrachtet „richtiger“ wäre es, Anforderungen und funktionale Beschreibung zu kaskadieren. Das heißt: aus den funktionalen Beschreibungen, die aus den primären Anforderungen hervorgehen, weitere Anforder-

ungen auf einer tieferen Detailebene abzuleiten. Aus diesen wiederum kann dann der beschriebene Testfall für den Test des Speichenschlosses abgeleitet werden (siehe Tafelbild oben).

Nachteil dabei: Der Aufwand, diese Kaskade zu modellieren, kann schnell den Nutzen übersteigen. Gerade in der agilen SW-Entwicklung gehen Trends daher in die Richtung „test-driven development“ oder „requirements as code“.

Aus Traceability-Gesichtspunkten ist ohnehin viel wichtiger, eine *sinnvolle* Granularität zu finden, bei der ein Test eindeutig einem zu testenden Sachverhalt zuzuordnen ist – ob dies eine Anforderung, oder eine Funktionsbeschreibung ist, ist für die Nachvollziehbarkeit letztlich weniger entscheidend.

3 | Risiken in der Produktarchitektur erkennbar machen

Methoden für die Identifizierung und Bewertung von Risiken und Fehlermöglichkeiten gibt es diverse, auf diese wollen wir an dieser Stelle nicht weiter im Detail eingehen. Um im Sinne der Traceability Transparenz über Risiken zu schaffen, sind unabhängig von der Art der Erhebung folgende zwei Fragen wichtig:

Dokumentation der identifizierten Risiken:

Wo und was sind die Risiken?

Dokumentation der Mitigation: Was wurde für eine Minderung der Risiken getan?

Im einfachsten Fall könnte es ausreichen, die Risiken, ihre Bewertung und die ergriffenen Mitigation-Maßnahmen in Listenform schlicht aufzuzählen. Die elegantere Variante ist freilich, die Risiken mit jenen Modellelementen zu verknüpfen, mit denen das Risiko in Verbindung steht, z.B. im Rahmen der funktionalen Beschreibung. Die modellierte Funktionsbeschreibung ist ohnehin ein guter Ausgangspunkt für ein umfassendes Risikomanagement, schließlich haben Fehler ihre Ursache meist im ungewollten Nicht-Funktionieren einer gewollten Funktion. Man kann sich umgekehrt also auch fragen: Welche Risiken können dazu führen, dass eine Funktion „kaputt“ geht?

Die Traceability-Fähigkeit „Erklären der sachlichen Zusammenhänge“ hilft außerdem dabei, die Auswirkung einmal identifizierter Fehlermöglichkeiten besser bewerten zu können.

Stellen wir uns gedanklich vor, ein für die Elektronik verantwortlicher Ingenieur identifiziert zur Funktion „Handy per Bluetooth koppeln“ das Risiko des ungewollten Verbindungsabbruchs. Daraus ergeben sich wenigstens vier Folgefragen:

- Welche Umstände führen dazu?**
- Welche Auswirkungen kann das haben?**
- Wie wahrscheinlich ist das?**
- Was tun wir dagegen?**

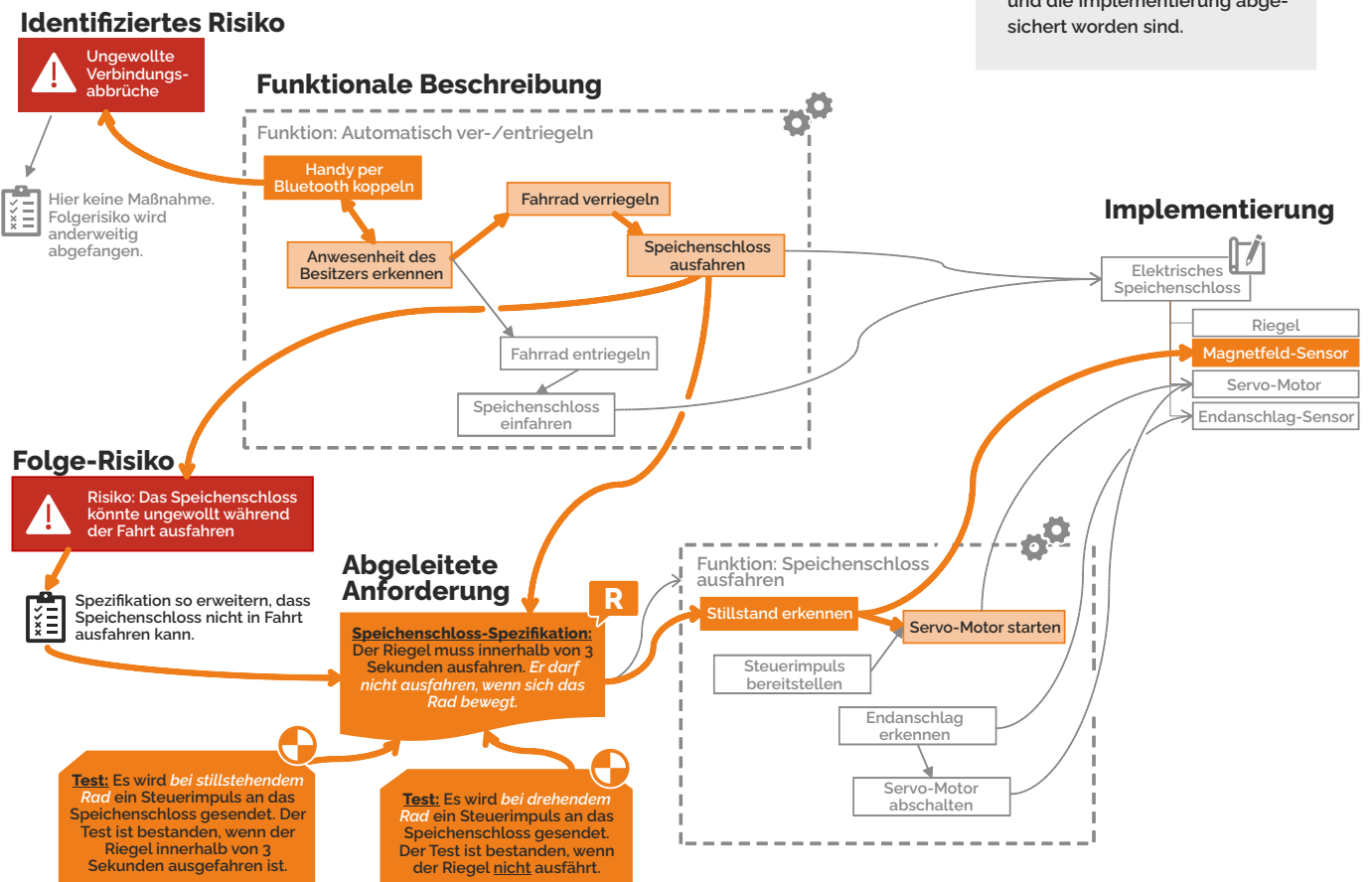
In unserem Beispiel könnte über die Funktionsbeschreibung als Auswirkung recht rasch als weiteres Risiko erkannt werden, dass ein ungewollter Verbindungsabbruch auch zum

Ausfahren des Speichenschlosses führen kann, was während der Fahrt wiederum fatal ist.

Für die weitere Betrachtung ist es letztlich auch egal, ob das zuletzt genannte Risiko aufgrund der Impact-Analyse ausgehend vom Bluetooth-Verbindungsabbruch oder anderweitig identifiziert worden ist. Der nächste Schritt ist in jedem Fall, Mitigation-Maßnahmen zur Risikominimierung zu treffen – oder bewusst zu entscheiden, dass eine Maßnahme aufgrund der Betrachtung aller Umstände und aufgrund eines geringen Risikos nicht notwendig ist. Im Sinne der Traceability ist auch hierüber Buch zu führen.

Transparenz über Risiken

Durch diese Art der Dokumentation kann nachgewiesen werden, dass Risiken identifiziert, Maßnahmen abgeleitet und die Implementierung abgesichert worden sind.



✓ 4 | Abweichungen systematisch erfassen und verfolgen

Es bietet sich an, sämtliche Vorkommnisse, die dem Hersteller in Bezug auf sein Produkt bekannt werden, strukturiert zu erfassen. Das können simple How-To-Anfragen sein, aber auch Fehlermeldungen, Defekte oder Reklamationen. Strukturiert heißt, diese Vorkommnisse zielgerichtet in Bezug auf Anforderungen, funktionale Beschreibung oder Elemente der Implementierung zu dokumentieren.

Ein Kunde stellt an der Hotline eine Frage, weil er mit der Bedienung des Akku-Ladegeräts nicht zurechtkommt? Verknüpfen Sie diese Anfrage mit der Funktion „Akku aufladen“. Ihnen wird gemeldet, dass ein Fahrrad-Display plötzlich ausgefallen ist? Ist das ein Einzelfall, dann seien Sie kulant, schicken dem Kunden ein Neues zu, aber vor allem: dokumentieren Sie es in Ihren Tools in Bezug auf das Fahrrad-Display. Ihnen wird bekannt, dass ein Fahrer Ihres Rades verun-

glückt ist? Erfassen Sie auch dieses Vorkommnis und weisen ihm eine hohe Kritikalität zu („Leib und Leben“).

Ausschlaggebend ist, diese Vorkommnisse nicht nur zu erfassen, sondern auch zu überwachen und angemessen zu reagieren. Bei 10 000 verkauften Fahrrädern bleibt es bei Einzelfällen mit defekten Fahrrad-Displays? Sollten Sie weiter beobachten, können Sie aber erstmal zurückstellen. Sie registrieren 75 Anfragen bezüglich der Bedienung des Ladegeräts? Leiten Sie Verbesserungsmaßnahmen hinsichtlich der Usability für die Weiterentwicklung des Ladegeräts ab und nehmen Sie diese in die Spezifikation mit auf. Sie haben den leisen Verdacht, dass Ihr Fahrrad eine erhöhte Gefahr für die Gesundheit des Fahrers darstellen könnte? Dann gehen Sie dem umgehend nach und informieren Sie im Bedarfsfall die für Sie zuständigen Aufsichtsbehörden.

Traceability kann natürlich nicht verhindern, dass es Abweichungen geben wird. Aber abgesehen davon, dass das damit verbundene Frontloading eine von vornherein sorgfältigere Arbeitsweise fördert, lässt Traceability Sie auch ruhiger schlafen, weil Sie die Einhaltung all Ihrer Sorgfaltspflichten lückenlos nachweisen können.



Umgang mit Abweichungen



Zur Traceability gehört auch, transparent mit Abweichungen umzugehen. Die systematische Erfassung und Bewertung in Bezug auf das jeweils verursachende Objekt erlaubt eine eindeutige Nachvollziehbarkeit.

✓ 5 | Änderungen nachvollziehbar durchführen

Produkte unterliegen ständiger Veränderung, insbesondere in einer schnelllebigen Zeit wie heute. Die Gründe von Änderungen können vielfältig sein: Angefangen bei neuen Kundenanforderungen, strengeren gesetzlichen Anforderungen, über Aspekte wie Prozess- und Kostenoptimierung, bis hin zu notwendigen Anpassungen in der Supply Chain und der Fertigung, wie in unserem E-Bike-

Beispiel mit dem geänderten Magnetfeldsensor.

In allen Fällen gilt: Jede Änderung an irgendeiner Stelle kann Auswirkungen an anderer Stelle haben. Aus der Änderung einer Anforderung (zum Beispiel aufgrund neuer Kundenwünsche oder geänderter Normen) folgt regelmäßig, dass Funktionsarchitektur und/oder Produktimplementierung angepasst werden müssen (Vor-

wärts-Durchlauf des V-Modells). Andersherum muss eine Anpassung an den Supply Chain-Prozessen mindestens dahingehend geprüft werden, ob die ursprünglichen Anforderungen bspw. bei geänderten Produktionsbedingungen immer noch eingehalten werden können (aus Sicht des V-Modells also rückwärtige Betrachtung).

Im Rahmen der Traceability-Kernfähigkeit „Nachvollziehbarkeit von Änderungen“ muss daher für jede Änderung transparent sein:

Was wird geändert?

Warum wird geändert?

Welche Auswirkungen hat die Änderung?
(Bewertung)

Welche weiteren Änderungen zieht die ursprüngliche Änderung nach sich?

Wer war an der Änderung beteiligt?

(auch: Wer hat freigegeben?)

Übertragen auf unser Beispiel des nicht mehr lieferbaren Magnetfeldsensors könnte sich ein solcher Änderungsvorgang wie folgt darstellen:

❶ In einem Änderungsantrag, ausgelöst durch den Einkauf, wird dokumentiert, dass der Magnetfeldsensor gegen ein vergleichba-

Änderungsantrag #98765

Grund: Magnetfeldsensor von Hersteller XYZ nicht mehr lieferbar.

Objekte: Sachnummer 12345 MAGNETFELDENSOR

Bewertung: Der Sensor dient der Stillstandserkennung des Hinterrads. Der Ersatzsensor hat identische Technische Daten, es ist daher davon auszugehen, daß eine Radbewegung weiter zuverlässig erkannt wird. Dies wird durch Tests abgesichert.

Die Änderung wird freigegeben
 nicht freigegeben

Max Muster

res Bauteil ausgetauscht wird (werden soll).

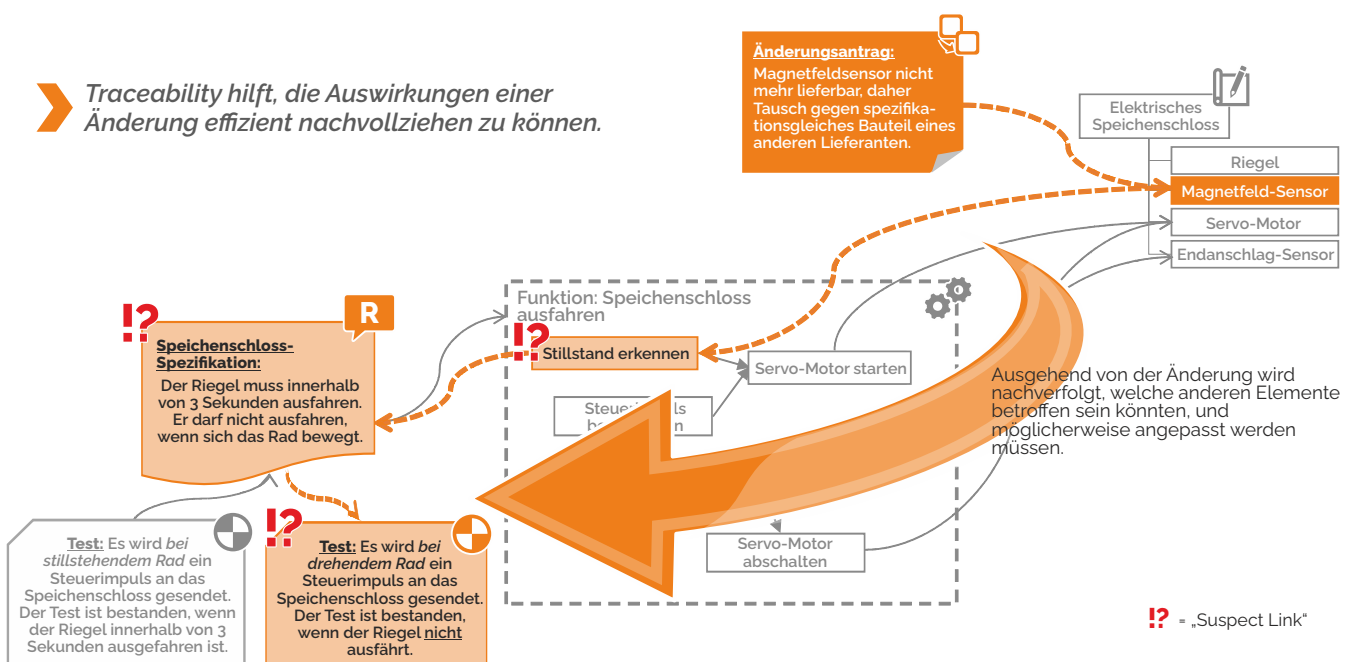
☛ **Vor Freigabe des Änderungsantrages werden die Auswirkungen identifiziert und bewertet.** Hierbei hilft die Traceability-Kernfähigkeit „Erklären der sachlichen Zusammenhänge“:

⇒ Der geänderte Sensor kann unmittelbare Auswirkung auf die Funktion „Stillstand erkennen“ und damit auf die Einhaltung der Anforderung

Die Entwicklungsstände müssen also in irgendeiner Form historisiert werden. In der Praxis lassen sich dafür vor allem zwei Methoden finden:

Arbeiten mit zeitlichen Gültigkeiten (Zeitscheiben, Gültigkeitsintervalle): In der Dokumentation wird jedes Element (z.B. ein Lastenelement oder eine Stücklistenposition) mit einer

hen können muss, desto mehr bietet sich jedoch an, mit Methoden des Konfigurationsmanagements zu arbeiten. Zwar können bei sorgfältiger Prozessgestaltung beide Wege eine eindeutige Traceability sicherstellen – die flexiblere und prozesssicherere Nutzbarkeit kann jedoch schnell den Aufwand für die Einführung eines Konfigurationsmanagements rechtfertigen.



derung „Speichenschloss-Spezifikation“ haben (siehe Bild oben).

⇒ Um abzuschließen, dass die Spezifikation weiterhin erfüllt wird, muss der zugehörige Test mit dem neuen Sensor zur Absicherung erneut durchgeführt werden.

☛ **Der Vorgang wird zusammenhängend dokumentiert und freigegeben.**

Da Änderungen nicht nur in der Entwicklungsphase, sondern auch wenn das Produkt bereits produziert und verkauft wird, erfolgen, ist es für die Traceability enorm wichtig, nachvollziehen zu können, auf welchem Entwicklungsstand sich ein gebautes Produkt befindet. Anders formuliert: Es muss identifiziert werden können, ob sich laut Soll-Zustand im Fahrrad die alte oder neue Version des Magnetfeldsensors befindet, und ob und wie dieser Zustand abgesichert wurde.

zeitlichen Gültigkeit versehen. Basierend auf bspw. eines Produktionsdatums können dann die zum Zeitpunkt der Produktion gültigen Stände ermittelt werden.

oder

Arbeiten mit Baselines und Snapshots (Konfigurationsmanagement): Zusammengehörige Elemente werden in einer Baseline oder einem Snapshot versionsgenau festgeschrieben. Zusammen mit dem gefertigten Produkt wird dokumentiert, auf genau welchem dieser „Snapshots“ die Fertigung basiert.

Die erste Methode ist vor allem im Rahmen der Dokumentation in klassischen PDM-Systemen anzutreffen, also bei Produkten, die historisch betrachtet einen eher geringen Komplexitätsgrad haben. Umso komplexer, und insbesondere mit umso mehr Softwarekomplexität die Produktentwicklung umge-

Unabhängig davon, mit welchen der beiden Methoden ihr Unternehmen arbeitet, jeder geänderte Zustand, sei es ein neuer Snapshot einer Konfiguration oder ein ab einem Zeitpunkt geändertes einzelnes Artefakt: In jedem Fall müssen Sie die Änderung eindeutig mit dem oben beschriebenen Änderungsantrag verknüpfen.

Nachvollziehbarkeit von Änderungen

✓ Ein integriertes Änderungsmanagement ist fundamentale Voraussetzung für die Traceability. Die Änderungsstände aller Eigenschaften, die für die Nachvollziehbarkeit relevant sind, müssen hierüber getrackt werden können.

Der Traceability-Backbone: Implementierung in Ihrer Tool-Landschaft

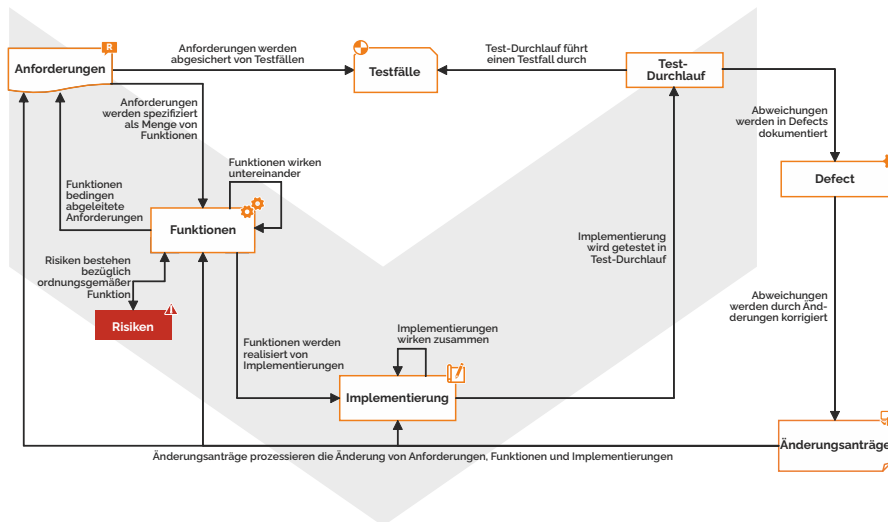
Das Informationsmodell: Grundlage für Prozeßverständnis...

Bis hierhin haben wir uns mit den methodischen Lösungsansätzen zum Herstellen einer durchgehenden Traceability in Bezug auf die genannten Kernfähigkeiten beschäftigt. Bleibt die Frage: Wie lassen sich diese methodischen Ansätze in einer IT-Bebauung abbilden?

muss anders behandelt werden, als eine SW-Anforderung. Und wenn Sie dem Rechnung tragen wollen, müssen diese als separate Geschäftsobjekte sichtbar werden.

Gleiches gilt für Hierarchien. Im hier links oben dargestellten Modell kann aus einer

Unternehmen ist es gut möglich, dass Ihr individuelles Informationsmodell am Ende der Übung sehr umfangreich und überladen wirkt – sie aber eigentlich nichts einfacher darstellen können, weil Ihre Unternehmensrealität nun mal so komplex ist. Das wäre dann der richtige Moment, sich auch über eine Konsolidierung von Prozessen und Methoden insgesamt Gedanken zu machen. Auch wenn diese Konsolidierung hier nicht weiter Thema sein soll, **die Traceability dankt es Ihnen.**



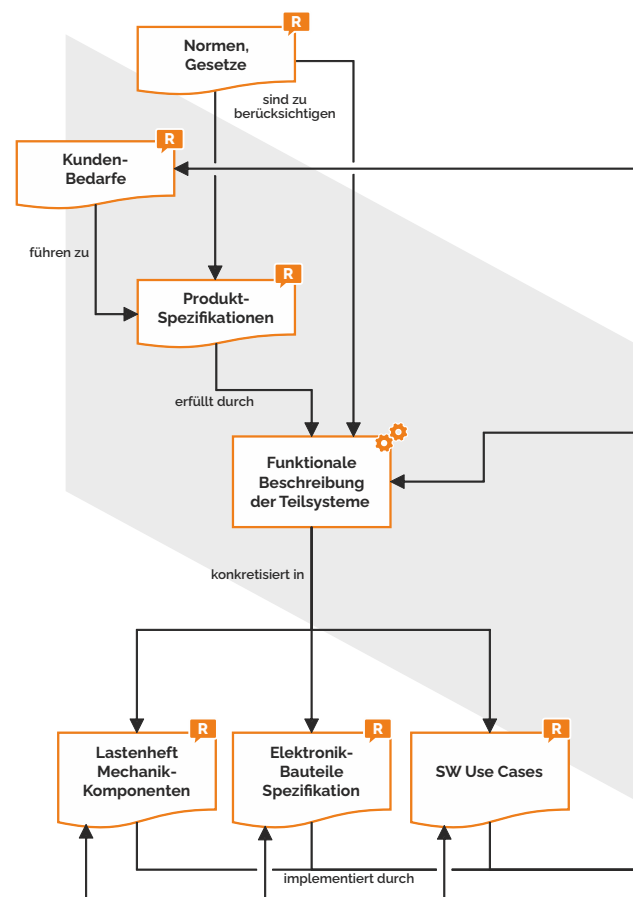
Ihr Informationsmodell sollte nicht zu abstrakt sein, sondern auf die Besonderheiten in Ihrer Prozeß- und Methodenwelt eingehen!

Der erste Schritt ist, die relevanten Geschäftsobjekte zu identifizieren und ihre Abhängigkeiten in einem Schaubild („Informationsmodell“) darzustellen.

Auf einer sehr hohen Flugebene könnte dieses Schaubild ungefähr so aussehen, wie links oben auf dieser Seite dargestellt. Auf diesem Abstraktionsgrad dürfte sich die Darstellung von Unternehmen zu Unternehmen wenig unterscheiden. Allerdings ist Anforderung nicht gleich Anforderung, und Implementierung ist nicht gleich Implementierung. Ob es nun gute, in der Sache liegende Gründe dafür gibt, oder es an der gewachsenen Unternehmensrealität liegt – das tatsächliche Informationsmodell wird diffiziler aussehen, und sollte auch bei gleichartigen Objekttypen (z.B. Anforderungen) die unterschiedlichen Ausprägungen herausarbeiten, die in Ihrem Unternehmen entscheidend sind. Eine Anforderung an eine mechanische Komponente

Funktions-Spezifikation „rekursiv“ eine weitere Anforderung abgeleitet werden. Dadurch lässt sich prinzipiell jede Art von Hierarchie darstellen. Das Modell ist also sehr generisch, für die praktische Anwendung allerdings wenig spezifisch. Praktikabler ist es, die Hierarchien, soweit sie durch die Entwicklungsmethodik, Ihre Produkt- oder Unternehmensstruktur vorgegeben sind, konkret zu benennen. Also beispielsweise: Aus einem Kundenwunsch folgt eine Systemanforderung, aus den Systemanforderungen folgen Anforderungen an mechatronische Komponenten sowie die darin enthaltene Software. Benennen Sie, was in Ihrem Unternehmen wichtig ist. In gewachsenen

So abstrakt wie möglich, ...



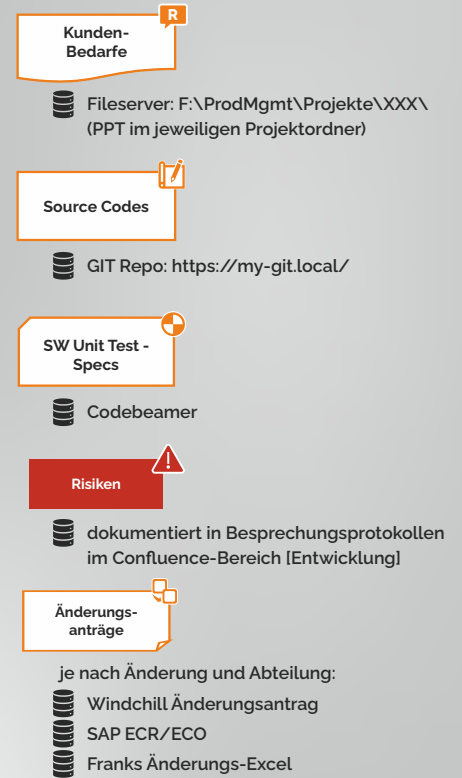
... und Mapping auf Ihre IT-Tool-Landschaft

Der zweite, aus IT-Sicht entscheidende Schritt, ist es, die einzelnen Objekte des Informationsmodells auf die Tool-Landschaft zu mappen, also zu dokumentieren, welche Informationen führend in welchem Tool zu Hause sind. Hilfreich ist dabei, nicht nur das Tool anzugeben, sondern auch, wie die zugehörigen Objekte im Datenmodell des Tools bezeichnet werden. Zwischen Geschäftsobjekt Ihres Informationsmodells und den Datenobjekten der Tools kann dabei auch eine n:m-Beziehung bestehen. Etwa weil sich ein Geschäftsobjekt auf verschiedene Datenobjekte verteilt (bspw. das Geschäftsobjekt „Baugruppe Endprodukt“ auf den SAP Materialstamm und die zugehörigen Stücklisten), oder weil mehrere fachlich separierte Geschäftsobjekte durch dieselben Datenob-

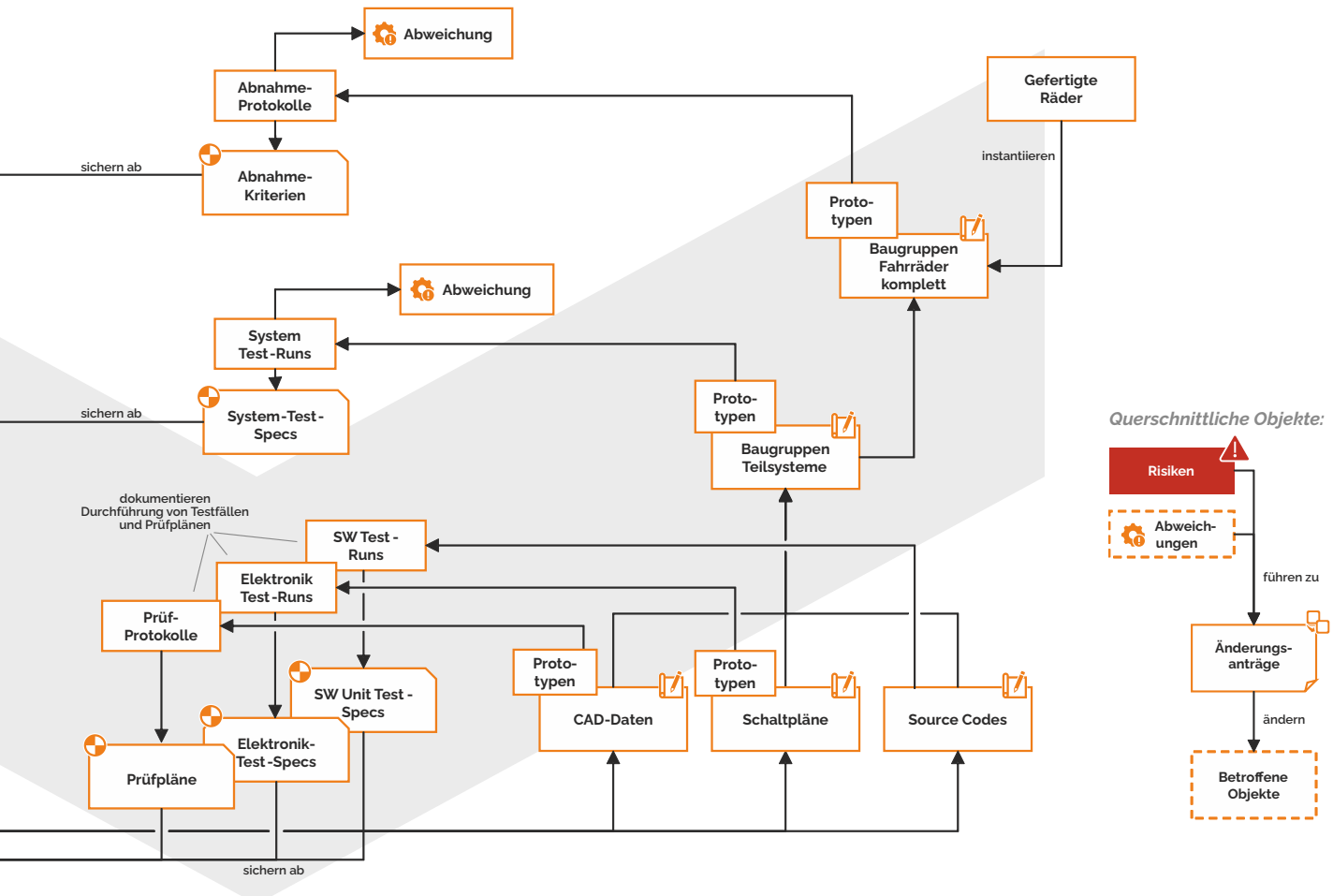
jekte abgebildet werden (bspw. das Geschäftsobjekt „Baugruppe Teilsystem“ ebenfalls durch SAP Materialstamm und Stücklisten).

Übrigens: Sie haben richtig gelesen. In unserem Beispiel rechts auf dieser Seite mappen die Kunden-Bedarfe auf Powerpoint-Dateien auf dem Fileserver, und manch eine Änderung wird in „Franks Änderungs-Excel“ gemanaged. Wenn das in Ist-Prozessen so gehandhabt wird, dann ist das das, was aus dem Informationsmodell hervorgehen soll. Ob das nun als gut oder schlecht zu bewerten ist, wird dann Gegenstand des nächsten Schrittes und der Optimierungs-Aktivitäten.

Tool-Mapping in einem exemplarischen Ist-Zustand:



... so genau wie nötig:



Knüpfen Sie die Verbindungen entlang Ihres Informationsmodells

Mit dem Informationsmodell in der Tasche geht es nun an die Detailarbeit. Insbesondere dann, wenn Sie auf gewachsenen und nicht frisch renovierten Prozessen aufsetzen, oder eine heterogene Tool-Landschaft vorfinden. Dann nämlich ist es gar nicht so unwahrscheinlich, dass es eben alles andere als trivial ist, von einem konkreten Geschäftsobjekt zum anderen zu finden. Auch wenn es sich mit Abstand betrachtet zunächst simpel anhört: die Frage, welche Anforderung oder Spezifikation von exakt welchem Testfall abgesichert wird, hat schon so manchen Qualitätsverantwortlichen das ein oder andere graue Haar beschert, vor allem wenn viele Änderungen im Spiel sind. Meist ist diese Verbindung zwar „irgendwie“ vorhanden, aber die Interpretation erfordert entweder Kopfwissen einzelner Prozessbeteiligter, oder umfangreiche Detektivarbeit, wenn sie denn überhaupt ohne Weiteres erschließbar ist.

Nehmen wir an, Sie beschreiben die Spezifikation Ihrer Teilsysteme in Form von in Word oder einem ähnlichen Textverarbeitungsprogramm erstellten Lastenheften. Dann könnte es irgendwo auf Ihrem Fileserver eine Datei in der Art

`E1_Speichenschloss_SPEC_Stand_14.7.22.pdf`

geben. Und weitere, ähnlich benannte Dateien mit anderen Ständen, z.B. vom 19.6.22 und 3.6.22 und so weiter. Die Risiken wiederum werden in Tagungsprotokollen der Risk-Assessment-Termine festgehalten.

Weiter vorne in diesem Papier haben wir in einem der Beispiele ein Risiko identifiziert, dass das Speichenschloss während der Fahrt ungewollt ausfahren könnte. Die Mitigationsmaßnahme hierzu ist, eine Erweiterung der Spezifikation – im Beispiel die Stillstandserkennung, welche mittels des Magnetfeldsensors realisiert wird. Ebendiese erweiterte Spezifikation wird sich nun in einem der Spezifikationsstände wiederfinden. Traceability bedeutet aber, die Spur in beide Richtungen verfolgen zu können. Also vom identifizierten Risiko zur geänderten Spezifikation, und von der geänderten Spezifikation zum auslösenden Risiko. Einigermaßen realistisch in einer Bestandswelt, in der zwar durchgängig dokumentiert wird, die aber noch nicht auf Traceability optimiert ist, wäre, wenn als Ergänzung

zum Tagungsprotokoll des Risk-Assessments an der entsprechenden Stelle ein textueller Vermerk in der Art „berücksichtigt in Speichenschloss-Spec am 17.6.“ stünde und in der Änderungshistorie der Spezifikation des Speichenschlosses ein Eintrag à la „Änderung wg. Risk-Assessment vom 28.5. (Aktivierung in Fahrt)“ zu finden wäre. Wohlgermerkt, das ist der optimistische Fall, genauso gut werden sich in gewachsenen Prozessen Beispiele finden lassen, wo eine oder beide dieser Hinweise fehlen. Aber selbst wenn diese Dokumentation in der Form wie beschrieben vorhanden ist, wäre diese aus Traceability-Gesichtspunkten mangelhaft:

Traceability ausgehend vom Risiko:

⚡ Es gibt im Fallbeispiel keine Speichenschloss-Spezifikation mit Stand 17.6. – naheliegender ist, dass der darauffolgend freigegebene Stand vom 19.6.22 gemeint sein könnte, sicher ist das aber nicht. Genauso könnte der freigegebene Stand vom 19.6. auf einer Änderung vom 10.6. beruhen, und die Änderung vom 17.6. erst mit dem Stand vom 14.7.22 freigegeben worden sein.

⚡ Innerhalb der Dokumentation könnte es mehrere Änderungen geben. Aus den Hinweisen in der Änderungshistorie lässt sich aber nur erkennen, dass es eine Änderung aufgrund des Risikos aus dem Assessment vom 28.5. gab, aber nicht welche der vielen Änderungen genau hierauf zurückgehen.

Traceability ausgehend von der Spezifikation:

⚡ Auch umgekehrt gilt: Die Änderungen in den einzelnen Kapiteln oder Absätzen der PDF-Spezifikation lassen sich nicht mehr ohne Weiteres auf das identifizierte Risiko zurückführen (es sei denn, der zuständige Ingenieur erinnert sich...)

⚡ Am 28.5. gab es vielleicht sogar zwei Risk-Assessment-Termine von zwei unterschiedlichen Abteilungen. In welchem der beiden Protokolle nun das Risiko zu finden ist, erschließt sich dann nur durch Suchen. Zu allem Überfluss dokumentiert die eine Abteilung ihre Tagungsprotokolle möglicherweise auf dem Abteilungslaufwerk, die andere in einem Wiki-System (=noch mehr Suchaufwand).

Damit kommen wir zu unserer ersten praktischen Handlungsempfehlung, falls Sie Ihre Tool-Landschaft hinsichtlich Traceability fit machen wollen:

1 | Stabilisierung der Bestandswelt:

Prüfen Sie, ob Sie ausgehend von ihrem Informationsmodell in der tatsächlichen Toolkette auf der notwendigen Granularitätsebene eindeutig von einem Geschäftsobjekt zum nächsten und wieder zurück finden.

Falls ja → Dokumentieren Sie, wie. Es sollte kein Insider-Wissen notwendig sein, um die Spur verfolgen zu können

Falls nein → Leiten Sie Verbesserungsmaßnahmen ein. Als Sofortmaßnahme würde in diesem Beispiel bereits eine kleine Prozessanpassung einiges bringen, bspw. dass in der Änderungshistorie detailliertere Informationen dokumentiert werden müssen, die die Zuordenbarkeit erleichtern. Und dass Änderungsstände nicht mehr über ein Datum, sondern über einen eindeutigen Versionsstand identifiziert werden.

Die Stabilisierung sorgt im Idealfall dafür, dass Sie bereits ohne den Austausch von Tools einen deutlich verbesserten Traceability-Grad erreichen. Im Beispiel wäre es aber dennoch weiterhin notwendig, die Spur im Bedarfsfall manuell zu verfolgen. Sie steigern also die Effektivität in gewissem Maße, aber nicht zwangsläufig die Effizienz. Der Wunsch wird in vielen Fällen jedoch sein, die Nachverfolgbarkeit sozusagen „auf Knopfdruck“ zu erhalten, oder in Dashboards und Cockpits Auswertungen wie bspw. einen Abdeckungsgrad der Anforderungen zu ermitteln.

Womit wir zur zweiten Empfehlung kommen:

2 | Optimierung der Toolkette:

Prüfen Sie, ob die im Informationsmodell beschriebenen Objekte nicht nur in menschenlesbarer Form eindeutig identifizierbar sind, sondern auch maschinell auswertbar.

Sollte das nicht der Fall sein, wie auch hier am Beispiel der Word-geschriebenen Spezifikationen und der Risiken in Tagungsprotokollform, dann planen Sie für jedes betroffene Informationsobjekt die Migration auf ein hier-

für geeignetes Werkzeug. Das können gängige PLM- oder ALM-Tools wie Doors, Codebeamer oder Jira sein. Vielleicht ist für ein anderes Objekt bereits ein geeignetes Tool für ein anderes Objekt im Unternehmen im Einsatz, dann bietet sich eine Wiederverwendung möglicherweise an. Am Ende sollten sich die Objekte Ihres Informationsmodells nicht auf zu viele unterschiedliche Tools verteilen.

Ganz am Ziel einer automatisierbaren und damit effizienten Traceability sind Sie damit meist aber immer noch nicht. Jedenfalls dann nicht, wenn die Tools entlang des Informationsmodells nicht aus der Hand eines Vendors stammen, oder wenn sie zwar aus einer Hand kommen, Sie aber nicht die Standard-Templates und Best Practices des Vendors implementiert haben. Dann nämlich bräuchten Sie sich im Idealfall weder um Schnittstellen noch um den Bau von Reports und Auswertungen kümmern, da diese vom Tool-Lieferanten bereits durchgängig mitgeliefert sein könnten. In der Praxis wird dieser Fall aus ver-

schiedenen Gründen eher selten realisierbar sein, weshalb wir zu unserer dritten und letzten Empfehlung hinsichtlich Tool-Kette kommen:

3 | Implementierung des Traceability

Backbones in Ihren Tools: Realisieren lässt sich das beispielsweise als ein eigenes Stück Software (bspw. OpenCLM), das sich an alle Tools andockt, über das sich Ihr Informationsmodell verteilt. Da Sie in den vorherigen Schritten ja bereits dokumentiert haben, wie die Informationsobjekte eindeutig zusammenhängen, und dafür gesorgt haben, dass dies auch maschinell auswertbar ist, sollte dieser Schritt von der fachlichen Betrachtung her vergleichsweise einfach ausfallen. IT-seitig kann das aber noch mal beliebig komplex werden, und hängt davon ab, ob die verschiedenen Tools Standards wie OSLC unterstützen, alternativ welche proprietären Schnittstellen die Backbone-Software bereits aus dem Stand beherrscht, oder inwieweit noch Customizing erforderlich ist.

Fazit

Wir hoffen, wir konnten Ihnen mit diesem Papier den ein oder anderen Impuls auf Ihrem Weg zur durchgängigen „Traceability over Lifetime“ mitgeben. Die Konzeption und Implementierung sind kein Hexenwerk, können aber im Detail sowohl methodisch als auch technisch durchaus herausfordernd werden. Förderlich sind daher ein strukturiertes Vorgehen und umfangreiche Expertise sowohl auf der Methoden- als auch auf der Tool-Seite, um unnötige Schleifen möglichst zu vermeiden.

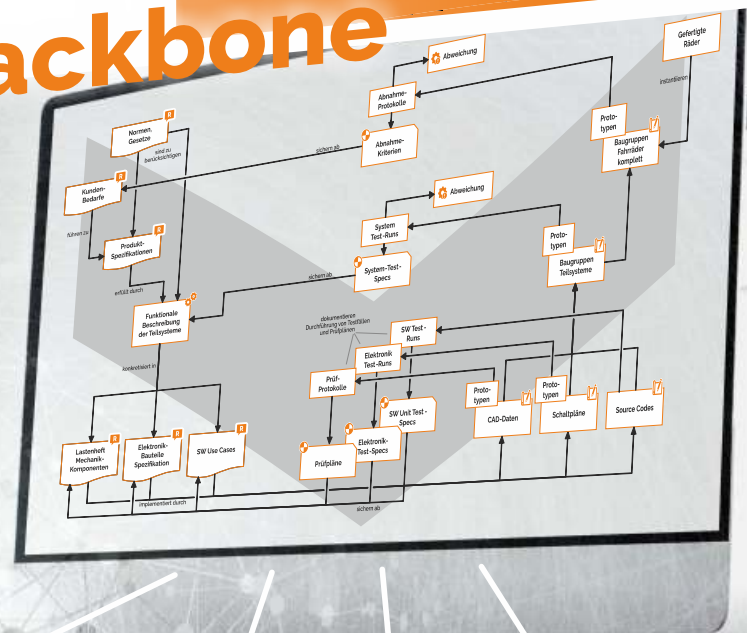
Ein Partner wie die **BHC Consulting** bringt diese Expertise von außen schnell und effektiv ins Unternehmen. Zusammen mit Ihnen analysieren unsere Systems Engineering-Experten Ihre Ist-Situation sowie Ihren Bedarf, und planen den Weg zur Umsetzung. Über den Unternehmensverbund der PROSTEP-Gruppe haben wir außerdem Zugriff auf Tool-Knowhow für alle relevanten Produkte.

Wir freuen uns auf das Gespräch mit Ihnen.

Traceability Backbone

Der Traceability-Backbone ...

- ... dockt an die verschiedenen Tools im Unternehmen an
- ... kennt die semantischen Zusammenhänge der Geschäftsobjekte
- ... stellt den Anwendern übersichtlich die Traceability over Lifetime dar



Codebeamer

SAP

GIT

Windchill

(Dargestellte Tools exemplarisch)

Notizen

A large grid of dotted lines for taking notes, consisting of 20 columns and 30 rows of small squares.

Impressum

„Traceability over Lifetime“ ist eine Veröffentlichung der BHC GmbH im Januar 2023.

Herausgeber:

BHC GmbH
Konrad-Zuse-Straße 5
71034 Böblingen

Verantwortlicher Redakteur:

Julian Weyer
BHC GmbH
Konrad-Zuse-Straße 5
71034 Böblingen

Druck:

Buch- und Offsetdruckerei Häuser GmbH & Co.KG,
50829 Köln-Bocklemünd

Bildnachweis:

Diese Veröffentlichung beinhaltet eigene Darstellungen, Illustrationen und Bilder, sowie von folgenden Künstlern:

Pixabay/Foundry
Pixabay/geralt
Pixabay/juergen-polle
Leonie Kolb
Pixabay/PCB-Tech
iStock.com/peshkov

Rechtlicher Hinweis:

Alle enthaltenen Informationen dienen der allgemeinen Information und stellen keine Beratung im Einzelfall dar und sollen diese auch nicht ersetzen. Der Herausgeber übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der bereitgestellten Informationen.

Experten-Kontakt



Philipp.Hasenaecker@b-h-c.de
... findet stets die gesunde Mitte zwischen Pragmatismus und Perfektion.



Julian.Weyer@b-h-c.de
... liebt Ende-zu-Ende gedachte PLM-Prozesse.



Melanie.Kluge@b-h-c.de
... behält mit Systems Engineering den Blick aufs Ganze.



Robert.Neher@b-h-c.de
... kennt die Welt der ALM-Tools aus dem FF.