

WHITEPAPER

SOFTWAREGETRIEBENE PRODUKTE

**– ORGANISATIONAL READINESS UND
METHODISCHE GRUNDLAGEN**

Management Summary

- Eigenschaft softwaregetriebener Produkte ist eine insgesamt erhöhte Komplexität, wobei diese sich jedoch von der Mechanik in die Software verschiebt
- Bei gleichzeitig steigenden Innovationsdruck müssen sich Unternehmen der Beherrschung dieser Komplexität stellen
- Schlüssel ist ein methodischer Blick auf das Gesamtsystem, zum Beispiel mit Methoden des Systems Engineering
- Erfolgskritisch bei der Verschmelzung von Mechanik, Elektronik und Software ist aber genauso der Faktor Mensch, und die Herausforderung, die Organisation für interdisziplinäres Denken zu öffnen

Inhaltsverzeichnis

- 1** Die Welt wird immer digitaler und von Software getrieben
– ihre Produkte auch **Seite 04**

- 2** Warum Sie für Software-getriebene Produkte den Blick
auf Ihr Lifecycle Management neu schärfen müssen **Seite 06**

- 3** PLM und ALM neu gedacht – die zunehmende Komplexität
beherrschbar machen **Seite 10**

- 4** Fazit **Seite 18**

1 Die Welt wird immer digitaler und von Software getrieben – ihre Produkte auch!

Die Digitalisierung an sich ist nichts Neues, kein Neuland, das es erst noch zu erforschen gilt. Die Digitalisierung ist schon lange da, und sie wird es auch bleiben. Die Vernetzung der Dinge schreitet immer weiter voran, und kratzt doch nur an der Oberfläche ihrer Möglichkeiten. Auch in unser aller Alltag lässt sich das beobachten, und es ist gut möglich, dass sich in fünf Jahren keiner mehr daran erinnert, dass ein Fahrrad einst ein rein mechanisches Produkt war.

Treiber und Enabler dieser Entwicklung ist Software. Die Funktion eines Produktes ergibt sich zunehmend nicht mehr nur aus seiner elektromechanischen Beschaffenheit, vielmehr ergibt sich die Funktion aus einer immer enger werdenden Symbiose aus Software und Hardware, sei es im Automobil, in der Medizintechnik, im Maschinen- und Anlagenbau – oder eben bei Fahrrädern.

Unternehmerisch betrachtet sind diese Entwicklungen Chance und Herausforderung gleichermaßen.

Chance, weil softwaregetriebene Produkte bei gesteigertem Wettbewerbs- und Innovationsdruck eine höhere Schlagzahl und verkürzte Entwicklungszeiten, aber auch ganz neue Geschäftsmodelle ermöglichen. In einem E-Bike etwa, in dem Kraftsensoren ohnehin die Unterstützungsleistung steuern, und sich das Display per Bluetooth mit ihrem Handy koppelt, ist es bis zur cloudbasierten Messung des Trainingserfolgs kein weiter Weg mehr. Oder wie wäre es mit einem kleinen Powerboost, wenn dann der Berg doch mal wieder zu steil ist – natürlich gegen einen kleinen Betrag via Handy-Payment?

Herausforderung aber deshalb, weil dieser Wandel alles andere als trivial ist. Auch wenn die Varianz auf der mechanischen Seite sinken mag, wird die Komplexität des Gesamtsystems steigen. Schon allein, weil hinter Hard- und Software oft unterschiedliche Entwicklerteams oder zumindest interdisziplinäre Personengruppen mit unterschiedlichen Denkweisen stecken, die in unterschiedlichem Takt die Innovationszyklen ihrer Gewerke vorantreiben.

Außerdem muss berücksichtigt werden, wie die Absicherung des Gesamtsystems erfolgt – insbesondere bei sicherheitskritischen Produkten. Apropos, je nach Branche und Produkt fordern Regularien unter Umständen, dass sich jeder Entwicklungsschritt, jede Einzelentscheidung bis hin zur Anforderungsdefinition zurückverfolgen lassen muss (mehr dazu im BHC Whitepaper „Softwaregetriebene Produkte – Herstellen einer durchgängigen Traceability“). Diese oder ähnliche Erfahrungen bezüglich steigender Komplexität dürften die meisten Unternehmen erleben, spätestens wenn sie sich erstmals mit der Thematik der Zusammenführung von Hard- und Software beschäftigen.

Auch wenn die Produktkomplexität zukünftig insgesamt steigen wird, verlagert sich ein großer Anteil der Komplexität von der elektromechanischen Seite in die Software. Das allein macht die Sache zwar noch nicht einfacher, aber durch den Wegfall der Beschränkungen der physischen Welt können andere Methoden der Komplexitätsbeherrschung angewandt werden, mit einer ungleich höheren Effizienz und Skalierbarkeit. Wer diese Disziplin beherrscht, wird seine Leistungsfähigkeit daher in weitaus höherem Maße steigern können, als der Aufwand zur Komplexitätsbeherrschung ansteigt. Leider gilt aber auch: ein Selbstläufer ist das nicht; und wer dies nicht schafft, dem wird die Komplexität auf kurz oder lang über den Kopf steigen.

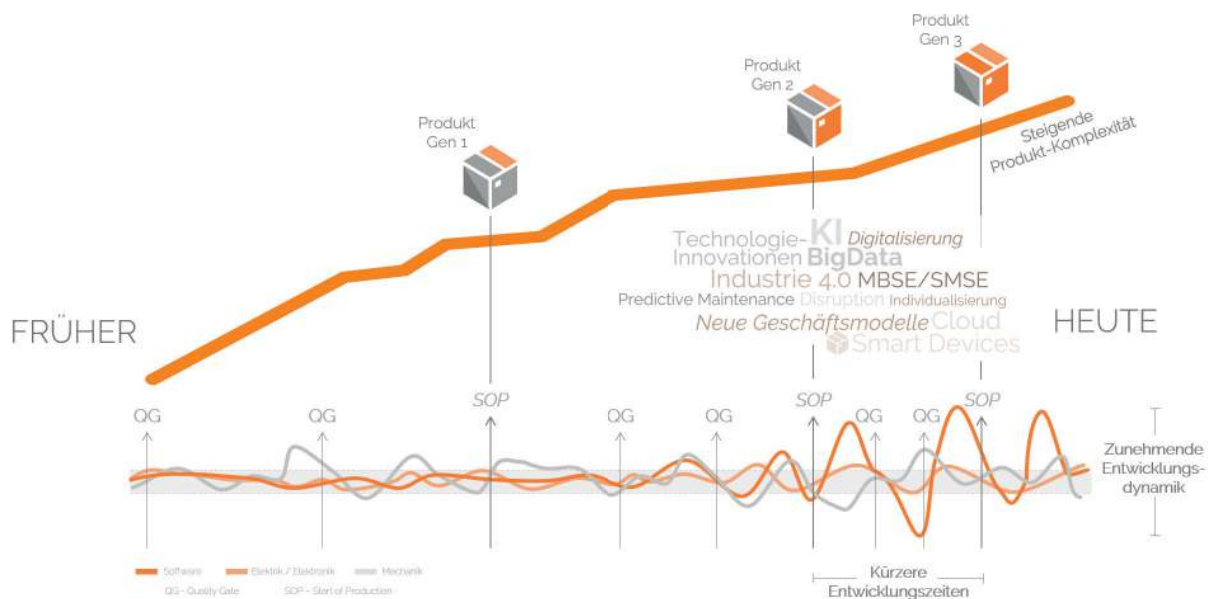


Abbildung 1: Steigende Produktkomplexität bei immer kürzeren Entwicklungszeiten

2 Warum Sie für Software-getriebene Produkte den Blick auf Ihr Lifecycle Management neu schärfen müssen

Die unterschiedlichen Historien des klassischen Engineerings und der Software-Entwicklung haben jeweils zu spezialisierten Tool- und Methoden-Welten geführt: PLM und ALM

Die Funktionssymbiose aus Mechanik, Elektronik und Software und die Verlagerung von Funktionskomplexität in die Software wird ermöglicht durch eine stetige Zunahme von Rechenleistung. Das wirkt sich allerdings auch auf den Umfang der Software aus, und wo viele Elemente zusammenwirken, steigt leider auch rein statistisch die Wahrscheinlichkeit von Fehlern. Zum Vergleich: ein Linux-Kernel von 1994 kam mit knapp 200.000 Codezeilen aus, 2018 waren es schon über 25 Millionen. Erschwerend kommt hinzu, dass bei fehlender Struktur die Effizienz in der Fehlersuche sinkt. Über die Zeit entwickelte sich aus dem Methodenbaukasten der Softwareentwicklung deshalb das, was wir heute als Application Lifecycle Management (ALM) kennen, und dessen Vorgehensmodell Einzug in das moderne Systems Engineering gefunden hat.

In der elektromechanisch geprägten Welt steht dagegen vor allem der Bedarf, den Konstruktionsprozess zu unterstützen, technische Daten von Bauteilen zu verwalten, und als Team zusammenarbeiten zu können. Der Fokus lag und liegt in vielen Unternehmen auch heute noch auf der Verwaltung der vielen einzelnen Bauteile, während die Herstellung der Durchgängigkeit vom Lastenheft bis in das fertige Produkt üblicherweise nicht durch ein im Unternehmen durchgängig verankertes Produkt Lifecycle Management-Konzept (PLM) (Methode+Tool) gestützt wird. Auch wenn moderne PLM-Lösungen heute deutlich mehr könnten, die Stärken und der Schwerpunkt der Nutzung liegen noch immer in einer Vielzahl von Disziplinen, die für ein reines Software-Produkt schlicht keine Relevanz haben: 3D-gestützte Konstruktion, mechanische Berechnungen, Montageplanung, sowie Unterstützung von Fertigungs-, Beschaffungs- und Logistikprozessen in der Supply Chain.

*Weder nebenläufige PLM- und ALM-Welten noch halbherzige Integrationen
werden softwaregetriebenen Produkten gerecht*

Ein Unternehmen, das mit rein mechanischen oder elektromechanischen Produkten aufgewachsen ist und nun zunehmend Software integriert, kämpft mit der Herausforderung, dass das vorhandene Tool- und Methodenset vorrangig aus der PDM/PLM-Welt stammt. Allerdings kommt nun nicht nur die Komplexität von Softwareprodukten additiv hinzu, sondern die Gesamtkomplexität potenziert sich. Schließlich steigt die Anzahl der Beteiligten, die Anzahl der prozessualen Schnittstellen und der Abstimmungsbedarf unter Beteiligten, deren Arbeits- und Denkweisen sehr verschieden sind. Die zu erfüllenden Anforderungen werden dabei nicht weniger sondern anspruchsvoller und, vor allem was Sicherheitsaspekte angeht (Stichwort Funktionale Sicherheit), auch strenger.

Wir beobachten in unseren Kundenprojekten, dass die Unternehmen dem Aufkommen von Software in Produkten oft mit einem der folgenden (vereinfacht dargestellten) Lösungsansätze begegnen:

- Entweder: **Software wird als Hardware-Anhängsel betrachtet**
(Motto: „Das bisschen Software ist doch auch nur ne Teilenummer“)

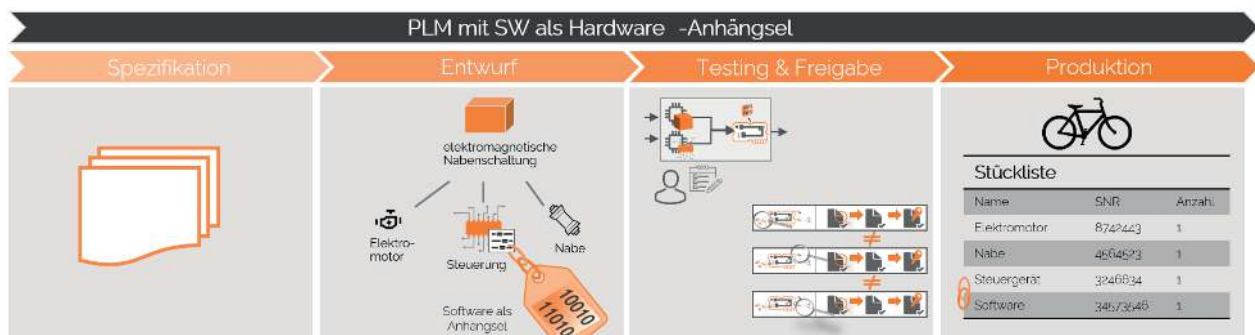


Abbildung 2: Software als Hardware-Anhängsel

Softwarekomponenten werden dabei mit elektromechanischen Komponenten gleichgesetzt und erhalten eine Teilenummer. Während Sie in einer Produktstruktur vielleicht noch wenigstens dem Steuergerät zuzuordnen sind, ist die Gesamtheit der gegenseitigen Abhängigkeiten spätestens in einer flachen Stückliste nicht mehr zuordenbar.

■ Oder: Neben der PLM-Welt wird eine eigenständige parallele ALM-Welt hochgezogen

(Motto: „Keine Ahnung, was die da drüben machen, interessiert mich aber auch nicht“)

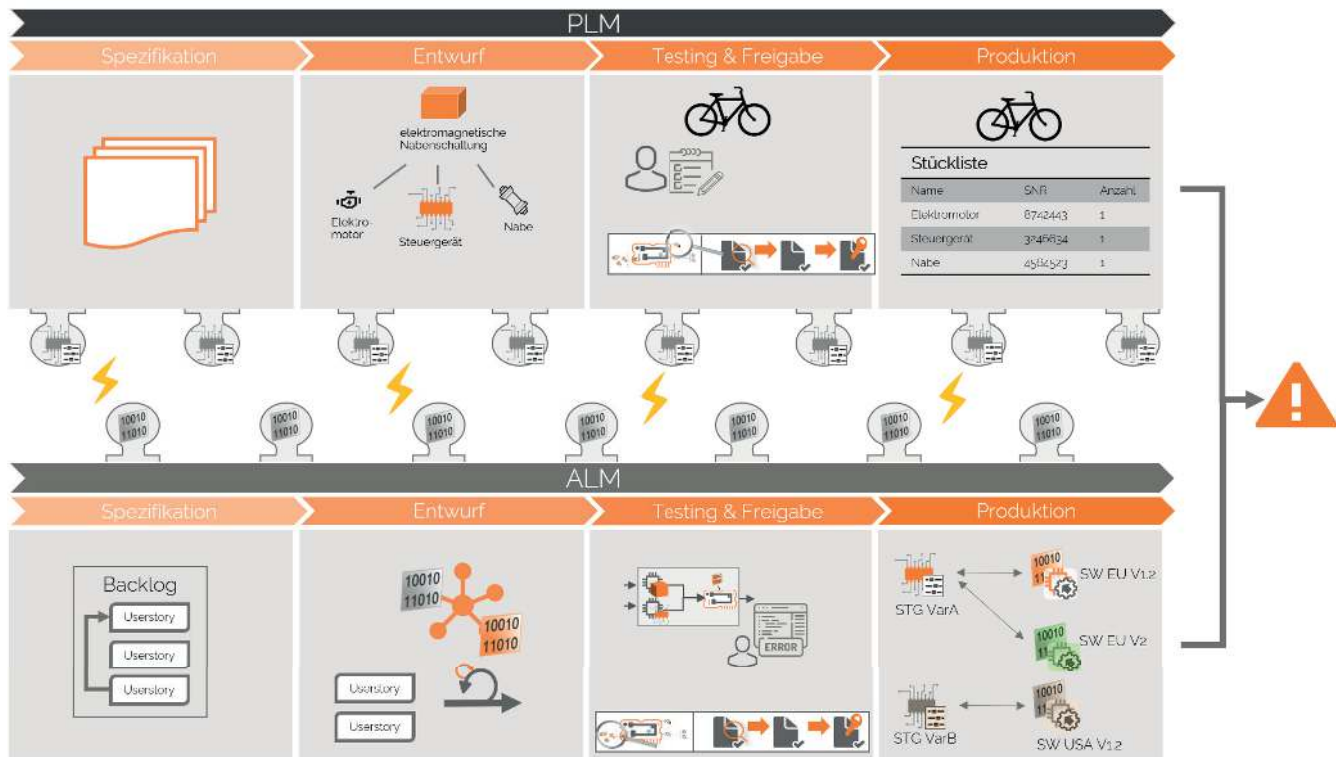


Abbildung 3: Nebenläufiges PLM- und ALM-Prozesse

Losgelöst von den starren Fesseln der elektromechanischen Entwicklung können die Softwareentwickler in ihrer ALM Welt ihre ganze Dynamik entfalten. Die Software wird in kurzen Iterationen und sehr agil für die aktuellen Kundenanforderungen optimiert. Was bei dieser vollständigen Trennung meist zu kurz kommt, ist die gegenseitige Synchronisation, die sich dann als Inkonsistenz bis in die Produktion und das fertige Produkt erstreckt.

Beide Ansätze sind in den seltensten Fällen optimal. Software als Anhängsel zur Hardware zu behandeln kann vielleicht noch funktionieren, wenn die Funktionsintegration nicht sonderlich hoch oder die Änderungsdynamik gering ist. Wenn die Software also gezielte und lokal abgrenzbare Spezialaufgaben löst und ihr Einfluss auf das Gesamtsystem unwesentlich ist. Außerdem darf man dabei keine hohe Erwartung an die Agilität der Software-Entwicklung stellen.

Der zweite Ansatz mag aus einer reinen Software-Perspektive richtig sein – aus einer ganzheitlichen Sicht auf das Produkt und seiner Wertschöpfungsprozesse aber nicht. Das Zusammenspiel und die Absicherung über Tool- und Abteilungsgrenzen hinweg sicherzustellen, wird dann möglicherweise zu einer nicht zu lösenden Herausforderung.

Je nach Komplexität der Produktvarianz und Skalierung der Unternehmensgröße können beide Strategien wenigstens eine Zeit lang funktionieren. Methodische und prozessuale Mängel werden glücklicherweise oft genug durch motivierte und kompetente Mitarbeiter ausgeglichen. Außerdem entwickeln Unternehmen schnell eine bemerkenswerte Toleranz und Leidensfähigkeit gegenüber systematischen Problemen. Sie kann sich bis zu einer erhöhten Risikobereitschaft steigern, die zur Folge hat, dass nicht ausreichend abgesicherte Produkte auf den Markt gebracht werden. Auch Kunden nehmen gewisse Imperfektionen des Produkts eine Zeitlang hin, bis sie keinen Mehrwert gegenüber Wettbewerbsprodukten mehr erkennen und das Weite suchen.



Abbildung 4: Gegenüberstellung von PLM und ALM

3 PLM und ALM neu gedacht – die zunehmende Komplexität beherrschbar machen

Die hohe Kunst ist also, Prozesse, Methoden und Tools bereitzustellen, die allen an der Produktentwicklung beteiligten Domänen Raum zur optimalen Entfaltung geben. Diese müssen aber gleichzeitig so orchestriert und aufeinander ausgerichtet werden, dass das Endprodukt alle Anforderungen erfüllt und als eine Einheit funktioniert. Das ist in erster Linie nicht nur eine Frage der Tools, der Prozesse und Methoden, sondern hängt vor allem davon ab, ob ein Unternehmen bereit ist für einen umfassenden Wandel.

Bereitschaft der Organisation

Leider wird der Faktor Mensch viel zu oft unterschätzt, wenn grundlegende Veränderungen anstehen. Und Unternehmen bestehen nun mal aus Menschen. Für ein junges Startup mag die Herausforderung, ein softwaregetriebenes Produkt zu entwickeln, eine eher kleine Herausforderung sein. Für ein gewachsenes Unternehmen mit bestehenden Strukturen und Gepflogenheiten kann es zur schier unüberwindlichen Hürde werden. Der Engpass ist dabei häufig nicht einmal die fachliche Kompetenz der Mitarbeiter, sondern die Steuerung und Orchestrierung des großen Ganzen bzw. die mangelnde Bereitschaft der Organisation, sich orchestrieren zu lassen.

In der Praxis erleben wir das als eine der größten Herausforderungen – oft mit der Folge, dass Projekte und Initiativen zwar mit viel Enthusiasmus gestartet werden, aber dann nicht zum gewünschten Erfolg führen. Und das um so häufiger, je ambitionierter und tiefgreifender das Vorhaben ist.

Hard- und Software kraftschlüssig, agil und sicher in einem Produkt zusammenzubringen, erfordert ein Umdenken aller Beteiligten. Das ist, als würden zwei Freundeskreise mit unterschiedlichen Hobbies mehr gemeinsam unternehmen wollen – wobei man im einen Freundeskreis viel wandert, im anderen dagegen lieber Fahrradtouren macht. Vielleicht mag sich das der eine oder andere Wanderer ja sogar so vorstellen, dass man dann zu den gemeinsamen Ausflügen die Radler

einfach mit auf die nächste Wanderung nimmt. Denn laufen könne ja schließlich jeder, aber radfahren liege eben nicht jedem. Das kann dauerhaft natürlich nicht die Lösung sein, die Fahrradfahrer würden zum „Anhängsel“ degradiert (ähnlich: SW als Anhängsel). Dabei ist eine Lösung, die beiden Seiten gerecht wird, gar nicht so weit entfernt. Wer sagt denn, dass auf einem gemeinsamen Ausflug alle wandern müssen. Man kann die Gemeinsamkeit auch durch ein gemeinsames Picknick auf halber Strecke und eine gesellige Einkehr am Abend erreichen. Um diese Lösung zu finden, braucht es aber von allen Beteiligten einen gewissen Blick über den Tellerrand und die Bereitschaft sich neuen Sichtweisen zu öffnen.

In Unternehmen hat man es aber üblicherweise nicht mit der Beschaulichkeit einer Ausflugsgruppe, sondern mit Abteilungen mit dutzenden bis mehreren hundert Menschen zu tun. Um so heterogener das Umfeld ist, desto wichtiger ist es, diese Vielfalt zusammenzuhalten und auf das gemeinsame Ziel hin zu dirigieren. Im Orchester macht das der Dirigent, auf dem Bau der Architekt in Funktion des Bauleiters. In Unternehmen erleben wir dagegen häufig, dass diese so wichtige Koordinationsfunktion eher nebenherläuft oder eben nicht mit ausreichend interdisziplinär ausgebildeten Personen besetzt ist.

Unsere Empfehlung lautet daher: Begleiten Sie die Veränderung aktiv und verankern Sie die Koordination mit klarem und starkem Mandat in Ihrer Organisation. Lassen Sie das nicht einfach jemanden als „Hobby“ nebenher betreuen (der womöglich beim Freiwilligen-Aufruf als letzter einen Schritt zurück gemacht hat). Wählen Sie Menschen aus, die ein gutes Standing haben und nicht zu fixiert auf ein Fachgebiet sind, sondern eine gesunde Mischung aus Pragmatismus und Methodenkompetenz mitbringen oder bereit sind, sich diese anzueignen.

Methodischer Unterbau

Ist die Bereitschaft, sich auf neue Denk- und Arbeitsweisen einzulassen, vorhanden, ist der nächste Schritt ein gemeinsames Vorgehensmodell zur Lösungsentwicklung zu etablieren.

Denken wir wieder an die beiden Freundeskreise. Nehmen wir an, diese haben den Schritt der „mentalen Öffnung“ hinter sich, und wollen nun für den nächsten Ausflug eine Lösung finden, bei der weder die Radler noch die Wanderer zu kurz kommen.

Bei einem kurzfristig einberufenen Webmeeting beschließen beide Seiten, dass es in den Schwarzwald gehen soll, und beide Parteien suchen auch das passende Wirtshaus zusammen aus. Etwas kniffliger wird es, eine passende Stelle für das Picknick zu finden – schließlich muss sie sowohl für Wanderer als auch für die Radfahrer gut erreichbar sein und den jeweiligen Bedürfnissen gerecht werden. Auch diesen Synchronisationspunkt müssen die Freunde gemeinsam festlegen. Ist der aber erstmal definiert, kann jede Teilgruppe das Optimum für sich rausholen und eine ideale Wander- bzw. Radstrecke planen. Julia, aus der Gruppe der Tourenradler und hauptberuflich Risikomanagerin, wirft noch ein, dass in der Gegend oft einige Wege gesperrt sind, und es daher vielleicht schwierig werden könnte, dass alle gleichzeitig an der Picknickstelle ankommen. Als risikomindernde Maßnahme wird deshalb vereinbart, dass man sich ungefähr zur Halbzeit kurz anruft, um die voraussichtliche Ankunftszeit abzugleichen und die Routen ggf. noch anpassen zu können. Für den Fall, dass alle Stricke reißen, werden noch Kartenspiele eingepackt, damit sich keiner langweilen muss, falls er zu früh ankommt.

Eine solche Planung, die im kleineren privaten Kreis noch intuitiv funktionieren mag, muss in der komplexen Unternehmensrealität aktiv gesteuert werden. Als Grundlage eignen sich Methoden des Systems Engineerings. Diese umfassen bereits sehr brauchbare Baukästen, mit denen sich alle Bestandteile eines Produktes bzw. eines Systems, auf die gemeinsamen Anforderungen ausrichten lassen.

Ob Sie dabei eine der Systems-Engineering-Normen oder -Standards exakt umsetzen oder sich nur daran anlehnen, ist fast schon mehr Geschmacksache bzw. nur dann wirklich wichtig, wenn Sie (wie in manchen Industrien erforderlich) Ihren Kunden oder anderen Stakeholdern gegenüber die Einhaltung vorgegebener Normen nachweisen müssen. Ausschlaggebend ist, dass Ihr Unternehmen ein geeignetes Vorgehensmodell wie zum Beispiel das V-Modell als eine Art Kompass verinnerlicht. Die tatsächliche Vorgehensweise ist der privaten Ausflugsplanung aber nicht unähnlich:

Sie beschreiben ganzheitlich Anforderungen an das integrierte Produkt (Ausflug mit Picknick und gemütlichem Ausklang), überlegen sich die funktionale Architektur (Anreise, Bewegung an der frischen Luft, Einkehr), bestimmen wer welchen funktionalen Beitrag zusteuert (die Anfahrt übernehmen die Personen, die Dachfahrradträger auf ihren Autos haben, die anderen organisieren dafür das Picknick), identifizieren Risiken (gesperrte Wege) und legen die integrativen Absicherungsstufen fest (gegenseitiger Anruf zu einer bestimmten Uhrzeit). Davon abgesehen hat jede Teildisziplin alle Freiheiten, ihre Stärken optimal auszuspielen (Wanderer: direkter, teils felsiger Weg, Radler: großräumige, weite Strecke auf ebenen Wegen).

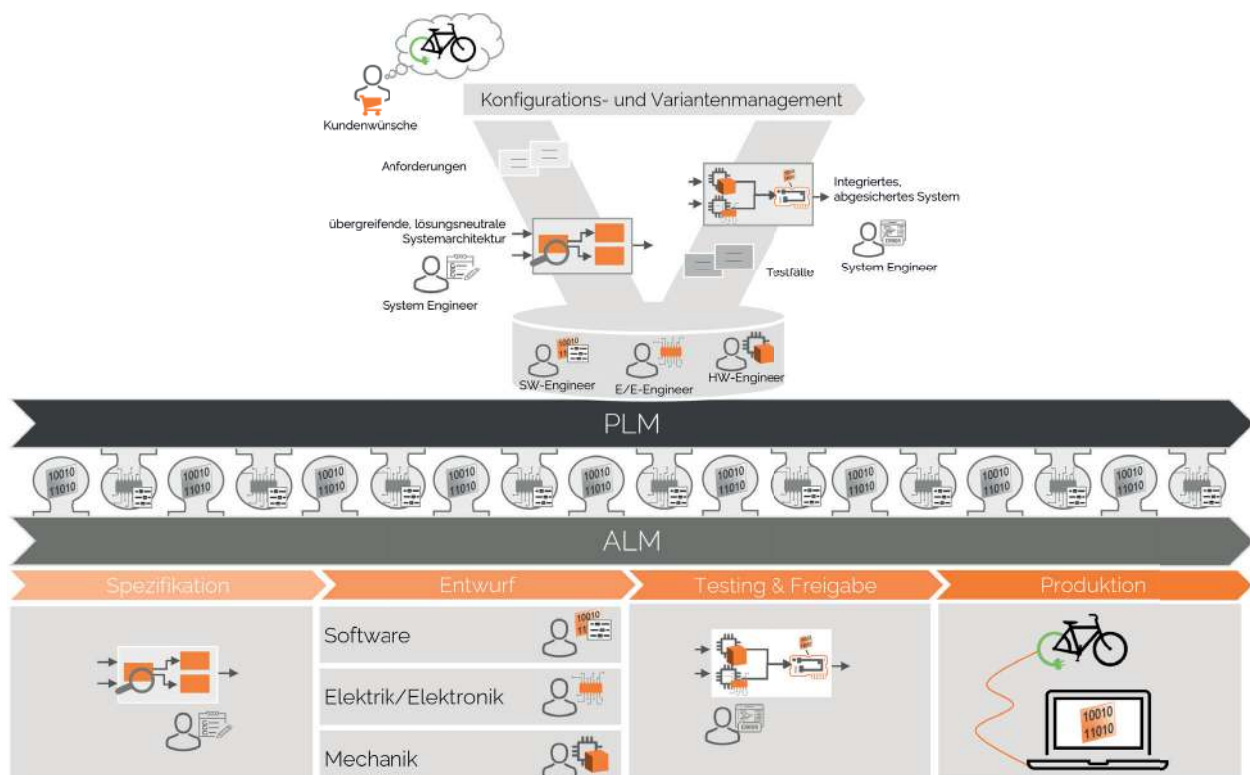


Abbildung 5: V-Modell des Systems-Engineering zur Synchronisation der Domänen

Was heißt das nun konkret für ein softwaregetriebenes Produkt? Wesentlich ist, dass Sie sich ganz am Anfang Gedanken darüber machen, was es können soll und welchen weiteren Anforderungen (z.B. normative Vorschriften) das Produkt genügen muss. Möglichst unvoreingenommen und ohne auf einen bestimmten Lösungsansatz zu zielen.

Beschreiben Sie, was ihr Produkt können soll |

Wenn die Anforderungen an Ihr Produkt klar sind, geht es im nächsten Schritt darum, den Funktionsbeitrag der einzelnen Teildisziplinen (Mechanik, Elektronik, Software) zu bestimmen. Sie befinden sich jetzt noch in der Phase, in der alle Beteiligten eng zusammenarbeiten müssen. Erliegen Sie aber nicht der Versuchung, schon jetzt alle Fragen bis ins letzte Detail klären zu wollen. Beispiel E-Bike: Für eine als Anforderung vorgegebene Funktion „Elektronisches Fahrradschloss“ müssen Sie zu diesem Zeitpunkt eigentlich nur festlegen, dass es auf mechanischer Seite eine irgendwie geartete Verriegelung geben und diese softwareseitig auf dem Fahrraddisplay bedienbar sein muss. Die konkrete Implementierung ist aber noch nicht relevant.

Schneiden Sie Ihre Systeme sinnvoll |

In der Architekturphase sind also zwei Dinge wichtig: Einen sinnvollen Schnitt des Systems zu finden und die Abhängigkeiten der Systemelemente untereinander zunächst abstrakt zu beschreiben. Sinnvoll heißt, dass die Abhängigkeiten zwischen den Teilsystemen möglichst gering sind, denn eben diese Abhängigkeiten werden fortan der Dreh- und Angelpunkt in allen Abstimmungen zwischen den einzelnen Entwicklungs-Teams in Ihrem Unternehmen sein. In der weiteren Projektarbeit gilt es dann, die Schnittstellen immer feiner zu beschreiben, bis schließlich das Produkt vollständig (bzw. als MVP) definiert ist.

Wichtig ist der Systemschnitt und die Beschreibung der Abhängigkeiten aber auch für die laufende Produktpflege, über die initiale Entwicklung hinaus. Diese Vorgehensweise ermöglicht es den einzelnen Disziplinen, sich frei zu entfalten und im eigenen Takt agil Innovationen voranzutreiben, solange dabei keine durch die Abhängigkeiten der Teilsysteme gesetzten Grenzen überschritten werden. Für ein softwaregetriebenes Produkt bedeutet das: wenn Hardware und Mechanik keine Grenzen setzen, ist auf Software-Seite alles möglich.

Schaffen Sie einen Mechanismus zur Synchronisation

Bei umfassenderen Weiterentwicklungen sind fast immer mehrere Teilsysteme und Disziplinen zu berücksichtigen, weil sich nicht mehr jede neue Anforderung allein softwareseitig realisieren lässt, ohne die Hardware zu verändern. Deshalb sollte zur Grundausstattung Ihrer Aktivitäten eine Entwicklungs-Roadmap gehören, die aufzeigt, welche größeren funktionalen Innovationen und welche Erweiterungen der Schnittstellen der Teilsysteme geplant sind.

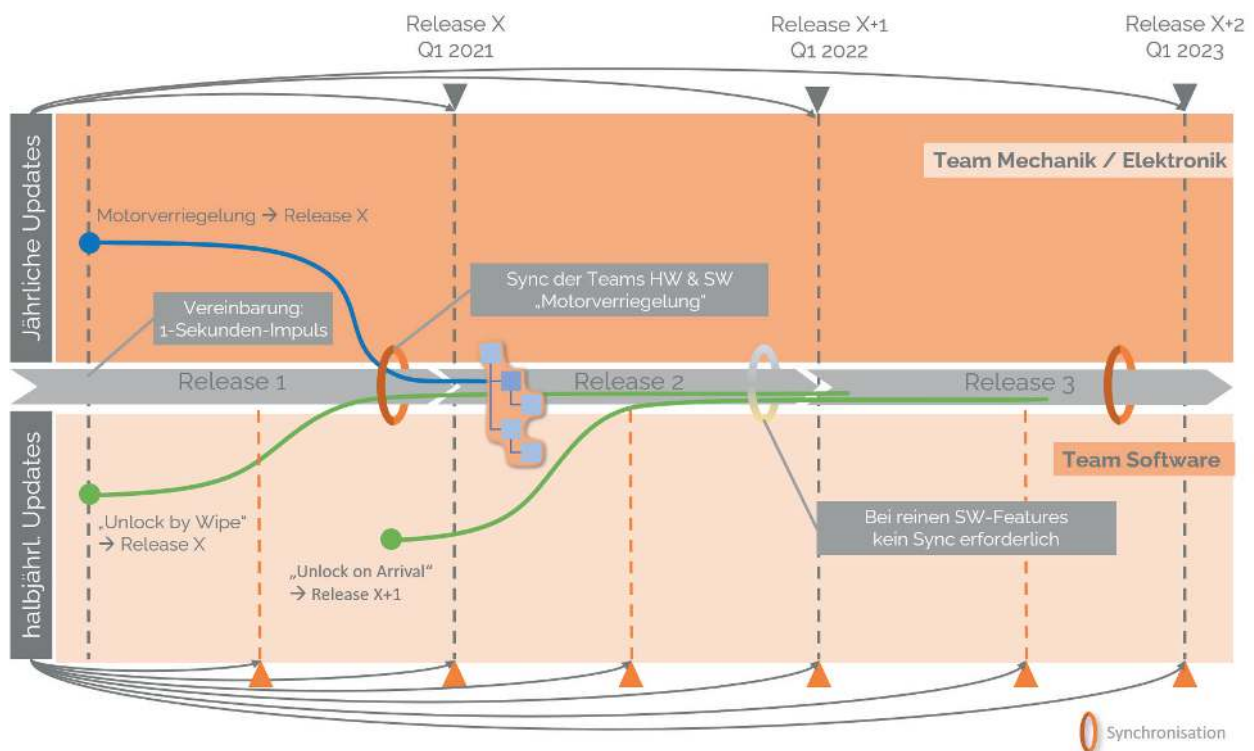
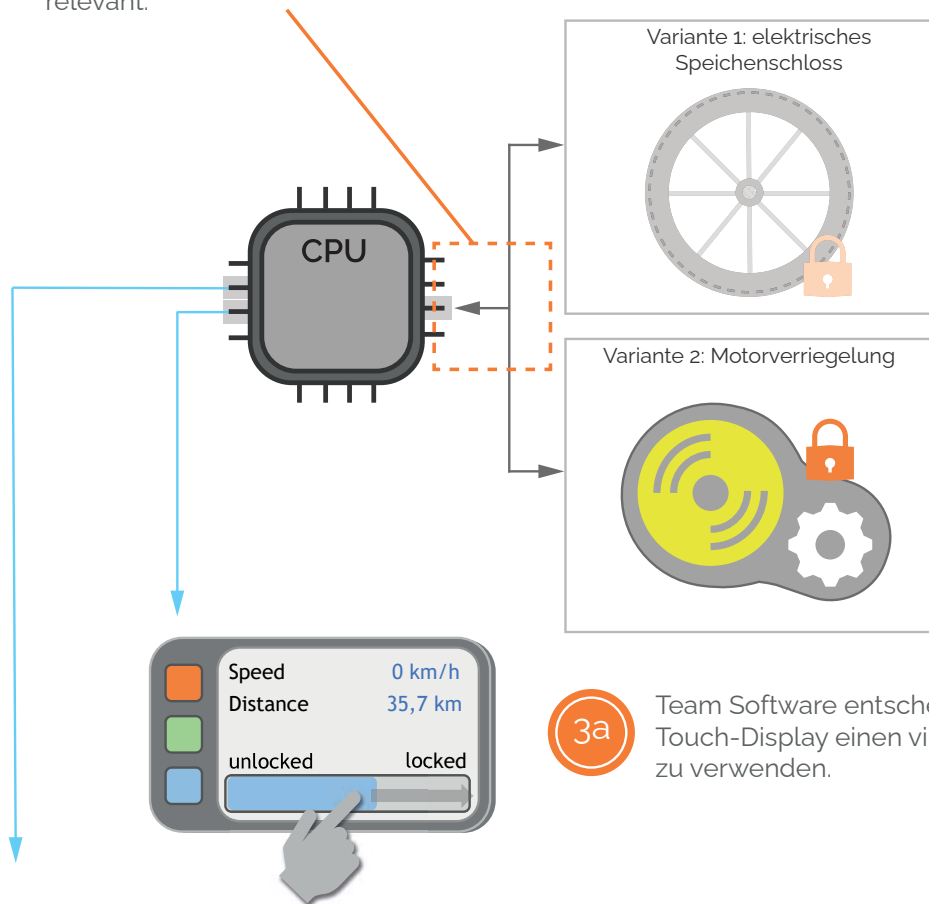


Abbildung 6: Die verschiedenen Domänen „Software“ und „Mechanik/Elektronik“ synchronisieren sich z.B. über Synchronisationspunkte einer gemeinsamen Roadmap.

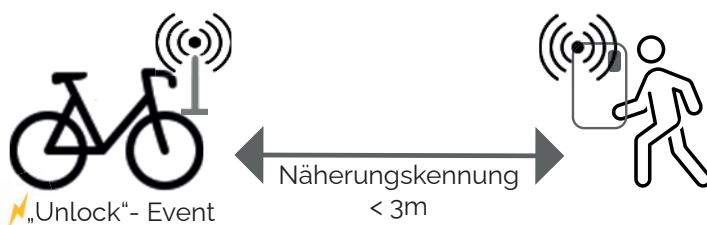
Das Gesamtsystem gibt den Takt vor, dem alle beteiligten Teilsysteme folgen müssen. Beispielsweise könnte ein E-Bike-Hersteller jährlich ein aktualisiertes Modell auf den Markt bringen. Die für diese jährlichen Updates geplanten Haupt-Features müssen durch die Entwicklungsteams rechtzeitig umgesetzt sein, wofür natürlich im Sinne der Architektur auch die Schnittstellen zwischen den Teilsystemen definiert werden müssen. Für die Funktion des elektronischen Fahrradschlosses könnte das zum Beispiel so aussehen:

- 1 Team Software und Team Mechanik/Elektronik vereinbaren, eine elektrische Schnittstelle auf einem Steuergerät zu entwickeln, über die der Schließzustand umgeschaltet und geprüft werden kann.
Wie die Verriegelung darüber hinaus technisch realisiert wird, stimmen sie nicht ab, weil nicht relevant.



- 2 Team Mechanik/Elektronik prüft im Rahmen der Entwicklung als Optionen die Verwendung eines elektrischen Speichenschlosses oder einer Motorverriegelung und entscheidet sich für letztere. Dies ist für das Team Software allerdings irrelevant.

- 3a Team Software entscheidet sich, auf dem Touch-Display einen virtuellen Schieberegler zu verwenden.



- 3b In einer späteren SW-Version gibt es außerdem eine Verriegelungsfunktion in Abhängigkeit davon, ob sich das Handy des Besitzers in der Nähe befindet.

Auch diese Weiterentwicklung ist für das Mechanik-Team nicht relevant.

Abbildung 7: Beispiel für die Abstimmung der Teams Mechanik/Elektronik und Software

Ansonsten entwickeln sich die beiden Teilsysteme unabhängig voneinander in ihrem eigenen Takt. So könnte für die Mechanik/Elektronik je Modelljahr eine Grundvariante und ein Facelift vorgesehen sein, während die Software agil vielleicht monatlich oder gar in noch kürzeren Intervallen, neue Releases veröffentlicht.

Die Einteilung in die Teilsysteme Mechanik/Elektronik und Software ist natürlich nur exemplarisch zu verstehen. In komplexeren Produkten kann es noch viele weitere geben, unter Umständen sogar mehrere Teilsysteme gleichen Typs parallel (z.B. mehrere Software-Teilsysteme). Wichtig zu berücksichtigen ist, dass trotz aller Bemühungen zur Vereinfachung die Abhängigkeiten dieser Teilsysteme untereinander sehr schnell sehr vielfältig und komplex werden können. Es ist daher essentiell wichtig, dass Ihre IT-Landschaft Sie dabei optimal unterstützt, den Überblick über all diese Abhängigkeiten zu behalten (zu diesem Thema mehr in unserem BHC Whitepaper „Softwaregetriebene Produkte – Zentrale Fähigkeiten für Ihre PLM/ALM-Tool-Landschaft“).

Mit der Etablierung eines solchen Methodenmodells legen Sie in jedem Fall die richtungsweisenden Grundsteine, um die Entwicklung Ihrer softwaregetriebenen Produkte prozesssicher in den Griff zu bekommen.

4 Fazit

Viele Produkte werden in Zukunft in einem weitaus höheren Maße von Software getrieben sein als heute. Dadurch steigt zwar die Produktkomplexität, aber es existieren Methoden, um diese Komplexität sicher zu beherrschen. Insbesondere ein durchgängiges Systems Engineering ist eine entscheidende Hilfe. Es setzt allerdings voraus, dass die Unternehmen eine Kultur des Wandels etablieren, sich interdisziplinärem Denken gegenüber öffnen, und auch bereit sind alte Gewohnheiten über Bord zu werfen. Angst vor allzu Unbekanntem muss dabei aber kaum jemand haben, denn vieles davon wird schon heute in den meisten Unternehmen auf die ein oder andere Weise gelebt. Oft fehlt es einfach nur an einer optimierten und zielgerichteteren Orchestrierung.

Die BHC GmbH hilft dabei, den Blick auf das Wesentliche zu schärfen und bringt methodische Expertise sowie Erfahrung aus einem breiten Portfolio ähnlicher Projekte mit.

Sie haben Anmerkungen oder Fragen?

Wir freuen uns auf ihr Feedback an:
info@b-h-c.de

BHC GmbH

Konrad-Zuse-Str. 5
71034 Böblingen
Deutschland

Telefon: +49 (0) 7031 20 5000 2
E-Mail: info@b-h-c.de

© 2021 BHC GmbH. Alle Rechte vorbehalten

IMPRESSUM

Herausgeber
BHC GmbH

Ansprechpartner
Philipp Hasenäcker
philipp.hasenaecker@b-h-c.de

Edition 1, 2021