

Wie trägt integriertes Anforderungsmanagement zur vollständigen Produktabsicherung bei?

Anforderungsbasiertes Testen in der Fahrzeugentwicklung

M. Demming

ZUSAMMENFASSUNG Die Absicherung von Produkteigenschaften ist ein zentraler Erfolgsfaktor in der Entwicklung zunehmend softwaregetriebener Systeme. Eine unvollständige Anforderungserhebung ist dabei eine der Hauptursachen für Entwicklungsfehler und Rückrufe. Ein integrierter Entwicklungsprozess, der auf einem interdisziplinären Anforderungsmanagement basiert, ein durchgängiges Datenmodell sowie eine zentrale Tool-Landschaft aus ALM- und PLM-Systemen bieten strukturierte Ansätze zur frühen Fehleridentifikation.

STICHWÖRTER

Anforderungsmanagement, Systems Engineering, Requirements-Based Testing

Requirements-Based Testing in an Integrated Vehicle Development Process

ABSTRACT The validation of product characteristics is a key success factor in the development of increasingly software-driven systems. An incomplete requirements elicitation is one of the main causes of development errors and product recalls. An integrated development process based on interdisciplinary requirements management, a consistent data model, and a centralized tool landscape combining ALM and PLM systems provide structured approaches for early error identification.

1 Sicherstellung der Produkteigenschaften in der Produkttransformation

Gemäß der DIN EN ISO 9000 wird die Qualität eines Produkts durch die Übereinstimmung seiner Merkmale mit definierten Anforderungen beschrieben. [2] Um eine ganzheitliche Erfüllung der Produktspezifikationen und damit deren Qualität sicherzustellen, ist eine Überprüfung ihrer Umsetzung bereits in der Entwicklungsphase erforderlich. Dabei ist die häufigste Ursache für das Scheitern von Entwicklungsprojekten eine unvollständige Anforderungserhebung. [3] Dies verdeutlicht die Wichtigkeit einer vollständigen Anforderungsbasis für eine Produktabsicherung in Entwicklungsprojekten. Beispielhaft zeigt sich, dass der Einfluss einer unvollständigen oder falschen Anforderungsbasis zu 80 % der aufgetretenen Fehler führt. [1]

Die Erhebung und Verwaltung von Anforderungen beschränkt sich nicht auf die Phase der Produktdefinition, sondern erstreckt sich über den gesamten Produktentwicklungsprozess. [4] Definierte Anforderungen bilden während jeder Phase der Entwicklung die Messgröße zur Bewertung des Zielerreichungsgrads und sind somit die Grundlage für die Beurteilung des aktuellen Entwicklungsfortschritts. Nicht in den Anforderungen abgebildete Entwicklungsziele lassen sich nicht über die Projektphasen hinweg nachverfolgen. Durch eine frühzeitig virtuelle Absicherung mittels Simulationen können Fehler, deren Behebung in späteren Phasen der Entwicklung mit deutlich höherem Zeit- und Kostenaufwand verbunden sind, bereits in einem frühen Stadium identifiziert und beseitigt werden. Als Faustregel wird zur Bewertung die 10er-Regel verwendet, die besagt, dass die Kosten zur Fehler-

behebung mit jeder fortgeschrittenen Phase um eine 10er-Potenz steigen. Der überproportionale Anstieg der Änderungskosten ist durch die steigende Anzahl von Personen und Unternehmensfunktionen begründet, die Änderungen ihres jeweiligen Arbeits-teils prüfen und gegebenenfalls adaptieren müssen. [2, 5]

Das Anforderungsmanagement sieht sich in der Produktentwicklung mit verschiedenen Herausforderungen konfrontiert. Eine davon liegt in der zunehmenden Komplexität, die einerseits durch eine steigende Anzahl von Anforderungen unterschiedlicher Stakeholder bedingt ist. Beispiele dafür sind neue Sicherheitsfunktionen, die für die Fahrzeugzulassung gefordert werden. Die zweite Herausforderung resultiert aus der wachsenden organisatorischen Komplexität, die an der Produktentwicklung beteiligt ist, und die dritte besteht in der Fähigkeit der Organisation zur Umsetzung von Agilität. Viele bestehende Organisationsformen sind nicht in der Lage, mit einer Anpassung oder Neudefinition von Zielen während des Entwicklungsprozesses umzugehen. [6]

1.2 Neue Produkte fordern ein erneuertes Vorgehen

Der Gegenstand der Entwicklung hat sich in den vergangenen Jahrzehnten von mechanischen Systemen über mechatronische Systeme hin zu cyber-physischen Systemen (CPS) entwickelt. CPS beschreiben eine Verknüpfung der virtuellen Welt, des „Cyberspace“, mit der physischen Welt, die aus physischen Materialien besteht und auf physikalischen Gegebenheiten beruht. Sie zeichnen sich durch in das System eingebettete Software sowie ihre Vernetzung mit dem „Internet der Dinge und Dienste“ aus.

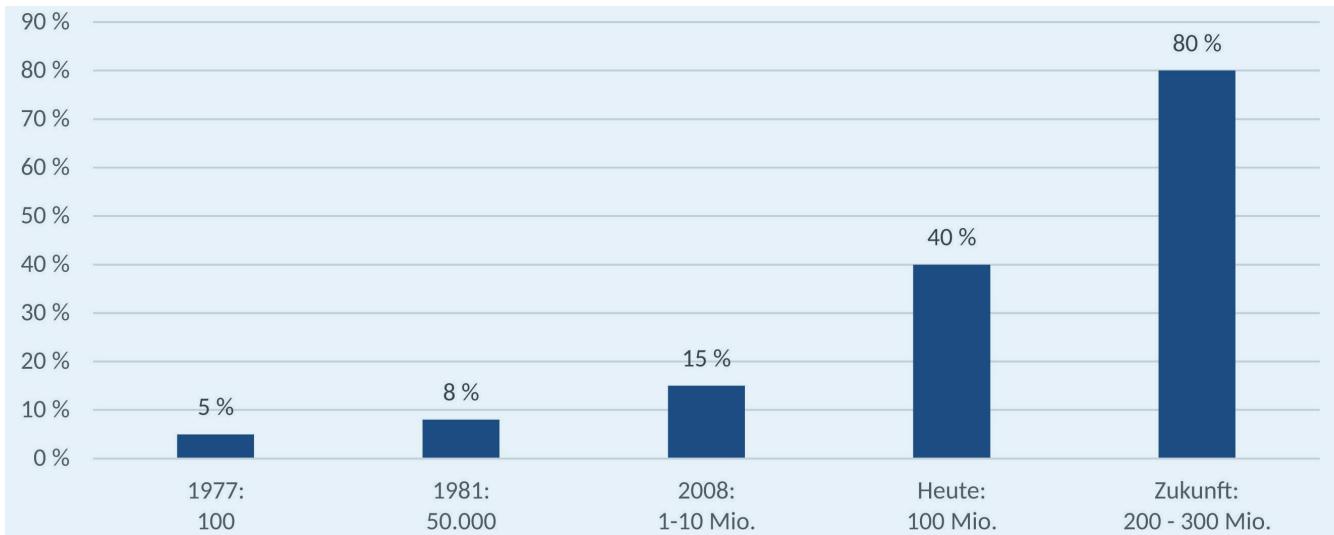


Bild 1. Anteil von Software an der gesamten Wertschöpfungskette sowie Anzahl an Quellcodezeilen. Grafik: [8] Ackermann 2021

Dadurch liegt bei der Entwicklung von CPS ein starker Fokus auf der Vernetzung hochgradig integrierter Systeme sowie auf der Systemanpassung während des laufenden Betriebs. [3]

Ein Beispiel für CPS sind moderne Fahrzeuge, die sich längst nicht mehr durch rein mechanische Eigenschaften differenzieren, sondern zunehmend durch Software-Funktionen und Vernetzung des Fahrzeugs, zum Beispiel mit dem Hersteller, im Flottenbetrieb oder mit anderen mobilen Systemen wie Smartphones. Dieser Software-Fokus eröffnet Fahrzeugherstellern und Zulieferern ein hohes Potenzial zur Abhebung vom Wettbewerb. [7] Der Anteil von Software an der Wertschöpfung moderner Fahrzeuge liegt bereits bei 40 % und dürfte in den kommenden Jahren auf rund 80 % steigen (siehe **Bild 1**). Dies hat zur Folge, dass der Umfang des Software-Codes exponentiell steigt. Während das Hubble-Weltraumteleskop mit zwei Millionen Codezeilen auskam, besteht die Software moderner Autos aus etwa 100 Millionen Zeilen. Mit dem zunehmenden Software-Anteil in Fahrzeugen steigt auch die Zahl der Rückrufe in erheblichem Ausmaß. Dies macht deutlich, dass Unternehmen ihre Strukturen und Prozesse überdenken und an den neuen Entwicklungsfokus anpassen müssen. [8] Daraus ergibt sich ein klarer Bedarf in Richtung einer weiterentwickelten, anforderungsbasierten Eigenschaftsabsicherung, die den veränderten Rahmenbedingungen und Herausforderungen gerecht wird und eine ganzheitliche Absicherung des Produkts erlaubt.

Der Handlungsbedarf liegt in neuen Prozessen und Methoden, die eine interdisziplinäre Entwicklung ermöglichen. Diese Prozesse müssen das Entwicklungsvorgehen über die verschiedenen Disziplinen der Entwicklung sowie zusätzlich über Unternehmensgrenzen hinaus organisatorisch und systemtechnisch unterstützen. Gegenwärtig erfolgt die Entwicklung meist disziplinspezifisch, so dass die entsprechenden Aktivitäten hauptsächlich in den Disziplinen Mechanik, Elektrik/Elektronik, Software und Dienstleistungen getrennt voneinander ablaufen. [9] Dies hat unter anderem zur Folge, dass Anforderungen nicht übergeben werden, Änderungen nicht nachvollzogen werden können und Informationen redundant erarbeitet werden, was wiederum zu Qualitätseinbußen, erhöhten Kosten sowie verlängerten Entwicklungszeiten führt. Das Anforderungsmanagement kann dabei eine zentrale Schnittstellenfunktion zwischen den Disziplinen übernehmen.

Die Absicherung spielt eine zentrale Rolle in der Entwicklung. Im Verlauf des iterativen Entwicklungsprozesses ist durch kontinuierliche Absicherung der Produkteigenschaften sicherzustellen, dass die angestrebten Entwicklungsziele weiterhin erreicht werden können. [5] Während in der Vergangenheit Geometrien und konstruktive Eigenschaften im Vordergrund der Produktentwicklung standen, liegt der Fokus inzwischen auf der gezielten Auslegung von Produkteigenschaften und Funktionen. Zentrale Merkmale wie Crash-Sicherheit, Kraftstoffverbrauch oder Fahrkomfort werden bereits in der Konzeptphase festgelegt und weiter konkretisiert. Da zu diesem Zeitpunkt noch keine detaillierten Konstruktionsdaten verfügbar sind, kommt der Simulation mithilfe von Konzeptwerkzeugen und vereinfachten virtuellen Modellen eine zentrale Rolle zu. Ein Werkzeug dafür ist der digitale Prototyp (DPT), der den physikalischen Modellstand eines Fahrzeugs bezeichnet, der schrittweise weiterentwickelt und vervollständigt wird. Er bündelt alle CAE-Berechnungsdisziplinen für eine frühe virtuelle Absicherung. Ziel ist eine funktionale Bewertung aller relevanten Fahrzeugfunktionen auf Basis definierter Randbedingungen und Kennwerte zu einem festgelegten Modellstand. Dank der schnellen Fortschritte in der Rechentechnologie lassen sich heute selbst Modelle mit mehreren Millionen Finite-Elementen (FE) effizient simulieren. Somit können Simulationen schon in kürzester Zeit Ergebnisse der Analyse liefern. In den späteren Phasen der Fahrzeugentwicklung wird der DPT für reale Erprobungen schrittweise durch physische Prototypen ersetzt. Trotz der zunehmenden Bedeutung virtueller Methoden bleibt die physische Erprobung zentraler Bestandteil der Fahrzeugentwicklung. [10]

2 Systemisches Denken in der Entwicklung

Das „Systems Engineering“ (SE) stellt einen Ansatz zur ganzheitlichen Systementwicklung dar. Dabei werden Systeme als Ganzes sowie deren Interaktion mit der Umgebung betrachtet. Ziel ist es, die Aktivitäten aller am Problemlösungsprozess Beteiligten zu koordinieren und dadurch eine Verbindung zu sowie einen Austausch zwischen den verschiedenen Fachdisziplinen herzustellen. [11] Die Nutzung des „Systems Engineering“-Ansatzes birgt unter anderem die Vorteile einer Kostensenkung in der

Absicherung durch frühzeitiges modellbasiertes Validieren und Verifizieren sowie eine frühere und übergreifende Qualitätssicherung durch die Verknüpfung und Integration von Teilmustern aus verschiedenen Ingenieurdisziplinen zu einem ganzheitlichen Gesamtsystem. [12]

Im Gegensatz zu anderen Modellierungsansätzen verfolgt das „Systems Engineering“ einen Top-Down-Ansatz. Er beginnt mit der Definition der Systemgrenzen und wird von einer schrittweisen Detaillierung der Systembeschreibung vom Groben ins Detail begleitet. Dabei werden sogenannte Blackboxes verwendet, die einzelne Subsysteme repräsentieren und ausschließlich über die Schnittstellen zu anderen Blackboxes definiert werden. Die inneren Strukturen dieser Subsysteme bleiben zunächst unbeachtet, um die Modellkomplexität zu reduzieren und den Fokus auf wesentliche Zusammenhänge im System zu richten. [9] Dabei bildet die modellbasierte Darstellung der Systeme und ihrer komplexen Zusammenhänge ein zentrales Prinzip des systemischen Denkens im „Systems Engineering“. [13]

2.1 Das V-Modell gemäß VDI/VDE 2206

VDI/VDE 2206 beschreibt einen standardisierten SE-Ansatz zur Entwicklung mechatronischer und cyber-physischer Systeme, in dessen Zentrum das V-Modell steht. Die ganzheitliche Systembetrachtung, das „Systemdenken“, und damit verbunden eine ganzheitliche Betrachtung aller Anforderungen der vor- und nachgelagerten Lebenszyklusphasen bilden eines der Kernprinzipien des V-Modells gemäß VDI/VDE 2206. Der linke Schenkel des „V“ stellt die Zerlegung von Systemen in ihre Elemente dar, und der rechte Schenkel repräsentiert die schrittweise Integration von Elementen zu einem Gesamtsystem. Die Grundlage für die Systementwicklung bilden das Geschäftsmodell, der konkrete Entwicklungsauftrag und die Restriktionen, die sich aus den nachgelagerten Phasen des Produktlebenszyklus ergeben. Im Verlauf der Systementwicklung werden die zentralen Aufgabenbereiche „Anforderungserhebung“, „Entwicklung der Systemarchitektur“, „Implementierung der Systemelemente“, „Systemintegration und Verifikation“ sowie „Validierung und Übergabe“ iterativ und zum Teil überlappend bearbeitet.

Das V-Modell gliedert sich in drei parallel verlaufende Stränge. Der äußere Strang steht für die Modellbildung und Analyse des Gesamtsystems sowie der Subsysteme, die begleitend zu den zentralen Entwicklungsaufgaben erfolgen. Der mittlere Strang bildet die wesentlichen Schritte der Systementwicklung ab, während der innere die Anforderungsentwicklung darstellt und den durchgängigen Umgang mit Anforderungen im gesamten Entwicklungsprozess hervorhebt. Zur Verdeutlichung des Zusammenspiels der beteiligten Disziplinen der Entwicklung werden die drei Stränge als miteinander vernetzte Teilstränge in Form von gepunkteten, gestrichelten oder durchgezogenen Linien visualisiert. Diese Darstellung hebt die notwendige tiefgreifende Vernetzung der beteiligten Disziplinen während der Implementierung der Systemelemente hervor. Ebenso soll die Darstellung der Disziplinen vor einheitlichen Hintergrundfarben der jeweiligen Stränge einen übergreifenden Austausch betonen. [3]

Die Anforderungen nehmen eine zentrale Rolle im V-Modell gemäß VDI/VDE 2206 ein. Auf die Anforderungserhebung, bei der auf Basis des internen oder externen Entwicklungsauftrags die Anforderungen aller Stakeholder zusammengetragen, strukturiert und analysiert werden, folgt das parallel zur Entwicklung

verlaufende Anforderungsmanagement. Dabei wird die Anforderungsbasis durch Erkenntnisgewinn, Rückkopplungen und Iterationen sowie durch die Aktualisierung von Kunden- oder Marktwünschen kontinuierlich ergänzt oder geändert. Anforderungsmanagement und Anforderungserhebung bilden gemeinsam die Anforderungsentwicklung und werden im V-Modell als innerer Strang dargestellt. Das Anforderungsmanagement spielt sowohl auf Gesamtsystemebene als auch bei der Gestaltung und Detaillierung einzelner Systemelemente innerhalb verschiedener Fachdisziplinen oder zwischen ihnen eine zentrale Rolle. Da sich die verschiedenen Systemebenen gegenseitig beeinflussen, müssen Änderungen von Anforderungen aus einer Disziplin stets im Gesamtkontext des Projekts berücksichtigt werden. [3]

Das V-Modell gemäß VDI/VDE 2206 stellt die sachlogischen Abhängigkeiten bei den Entwicklungsaufgaben dar, ohne diese zeitlich einzuordnen. Ebenso ist explizit keine sequenzielle Abarbeitung der Aufgaben gefordert, sondern ein partiell paralleles und iteratives Durchlaufen der Aufgaben empfohlen. Somit soll das V-Modell unabhängig von der gewählten Projektorganisation sowohl bei Großprojekten als auch bei agilen Entwicklungsprojekten Anwendung finden. [3] Im Gegensatz zu anderen Vorgehensweisen, die auf dem Stage-Gate-Modell basieren, sind keine „Gates“ oder Meilensteine vorgesehen, die mit Projektzeitpunkten korrelieren oder eine Freigabefunktion besitzen. Die definierten Kontrollpunkte sollen demgegenüber der Entwicklung zur Reflexion des Entwicklungsstands dienen. Im V-Modell werden die Planung und Durchführung der Eigenschaftsabsicherung durch Blockpfeile dargestellt und sollen parallel zu den zentralen Entwicklungsaufgaben ablaufen. Die Absicherung kann kontinuierlich und beliebig oft vorgenommen werden. Die Prüfkriterien zur Absicherung werden bereits auf der linken Seite des V-Modells vorgegeben. [3]

Die Prognose und Absicherung von Systemeigenschaften verlaufen parallel zu den zentralen Entwicklungsschritten. Somit lassen sich die erwarteten Eigenschaften des Systems bereits in frühen Phasen modellbasiert abschätzen. Ziel dieser begleitenden Eigenschaftsabsicherung ist die kontinuierliche Überprüfung der angestrebten Lösung mit Hilfe virtueller Prototypen – etwa Computersimulationen oder „Software in the Loop“ (SiL) – sowie physischer Prototypen, beispielsweise Hardwaretests. Eine Kombination beider Testverfahren bildet das „Hardware in the Loop“ (HiL)-Testen. Die Verifikation erfolgt stets auf derselben Systemebene, auf der zuvor die Anforderungen definiert wurden. Dabei muss für jede Anforderung eine passende Verifikationsmethode ausgewählt werden, jedoch können einzelne Verifikationsmaßnahmen mehrere Anforderungen gleichzeitig abdecken. Idealerweise findet die Verifikation in einem möglichst frühen Entwicklungsstadium und auf einer möglichst niedrigen Ebene der Systemhierarchie statt. Dies sollte bereits früh in der Planung der Eigenschaftsabsicherung berücksichtigt werden. [3]

Das V-Modell nach VDI/VDE 2206 fördert die interdisziplinäre Zusammenarbeit zwischen den Disziplinen Mechanik, Elektrik/Elektronik, Software und Dienstleistungen, indem es die sachlogischen Zusammenhänge auf Makroebene strukturiert darstellt. Konkrete Handlungsempfehlungen zur praktischen Umsetzung oder eine detaillierte Beschreibung eines Entwicklungsprozesses enthält das Modell nicht. Die Ausgestaltung und Überführung der dargestellten Zusammenhänge in einen konkreten Entwicklungsprozess auf Mikroebene obliegen dem Anwender innerhalb des Projekts.

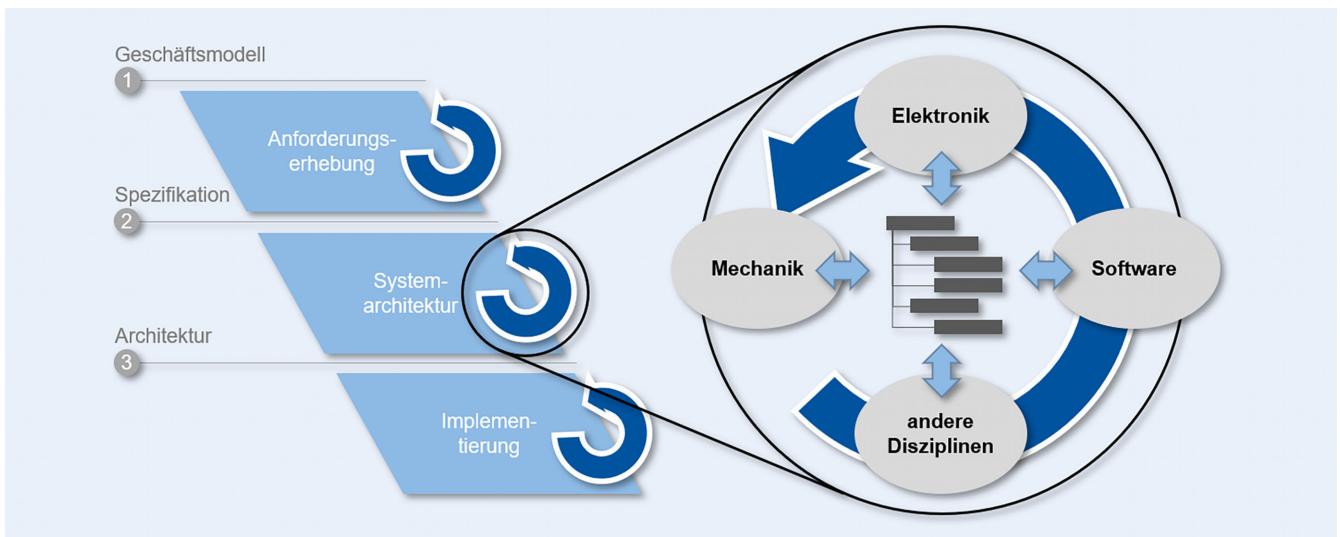


Bild 2. Iterationen auf Basis eines übergreifenden Strukturmodells im V-Modell. *Grafik: eigene Darstellung*

3 Integriertes Anforderungsmanagement für eine ganzheitliche Absicherung

Um eine vollständige Produktabsicherung zu gewährleisten, muss eine Methodik geschaffen werden, die disziplinübergreifend die System-, Subsystem- und Komponenteneigenschaften validiert und verifiziert. Dafür ist ein integrativer Ansatz gemäß dem „Systems Engineering“ zu wählen, der ein mechatronisches oder cyber-physisches System ganzheitlich betrachtet. Die Grundlage dafür bildet das integrierte Anforderungsmanagement, das über den gesamten Entwicklungsprozess als „Backbone“ der Entwicklung dient. Um die vollständige Produktabsicherung transparent sicherzustellen, muss das integrierte Anforderungsmanagement vollständig, konsistent und nachvollziehbar sein.

Die genannten Anforderungen an ein zeitgerechtes Vorgehen machen eine strukturierte Unterstützung durch geeignete Entwicklungsmethoden, Werkzeuge und Hilfsmittel erforderlich, die derzeit jedoch nur eingeschränkt oder gar nicht zur Verfügung stehen, da es an etablierten Methoden, Prozessen und Vorgehensmodellen mangelt, die eine disziplinübergreifende Produkt- und Systementwicklung unterstützen. Die in den einzelnen Fachbereichen angewendeten Entwicklungsansätze unterscheiden sich erheblich voneinander, was zu einer Denk- und Arbeitsweise in fachspezifischen Silos führt. Zudem besteht ein deutlicher Mangel an IT-Werkzeugen, die eine effektive Abstimmung und Synchronisierung zwischen den verschiedenen Fachdisziplinen unterstützen. [9, 14]

Zur Umsetzung eines integrierten und gesamtheitlichen Anforderungsmanagements in der Entwicklung cyber-physischer Systeme bedarf es einer umfassenden Anpassung des gegenwärtigen Entwicklungsvorgehens. Diese Anpassungen fokussieren sich auf die Themenfelder

- Prozess, der das Vorgehen definiert,
- Modell, das die Informationsgrundlage aller Daten bildet, und
- Tool-Umgebung, die die Umsetzung von Modell und Prozess ermöglicht.

3.1 Anforderungen an den Prozess

Da der Entwicklungsprozess in der Produktentwicklung nicht linear verläuft, sondern viele Arbeitsschritte mehrfach durchlaufen werden und sie eine kontinuierliche Anpassung der Planung erfordern, sind Iterationen wichtige Elemente des Prozesses. Diese Iterationen müssen sowohl auf der Makro- als auch auf der Mikroebene ermöglicht werden. [5]

Die größte Herausforderung des Entwicklungsprozesses besteht in der Abstimmung der unterschiedlichen Fachdisziplinen. Besonders deutlich wird dies in der Phase des disziplinspezifischen Entwurfs, da in diesem Abschnitt das übergreifende Systemkonzept innerhalb der einzelnen Disziplinen konkretisiert und ausgestaltet wird. [14]

Auf der Makroebene ist zur Strukturierung des Entwicklungsprozesses die Umsetzung des V-Modells gemäß VDI/VDE 2206 erforderlich, das explizit iterative Abläufe vorsieht. **Bild 2** veranschaulicht das parallele Arbeiten der verschiedenen Fachdisziplinen im linken Zweig des V-Modells. Dabei greifen alle Disziplinen auf ein gemeinsames Modell zu und nutzen eine einheitliche, zentrale Datenbasis. Dies ermöglicht eine reibungslose Abstimmung und erleichtert den Datenaustausch erheblich. Kontrollpunkte dienen dazu, den Datenstand abzulegen und den Fortschritt der Entwicklung zu reflektieren.

Auf der Mikroebene ist eine detaillierte Prozessbeschreibung notwendig, die konkrete Aktivitäten innerhalb der einzelnen Entwicklungsschritte definiert. Diese Detaillierung ist in die übergeordnete Makrostruktur eingebettet. Im Unterschied zur Makroebene werden auf der Mikroebene nicht nur sachlogische Abhängigkeiten dargestellt, sondern auch konkrete Abläufe festgelegt. Dabei dominieren häufig sequenzielle Prozessschritte, die jedoch in ein übergeordnetes, iteratives Gesamtmodell eingebunden sind.

3.2 Entwicklung eines zentralen Datenmodells

Der Wandel in der Art der zu entwickelnden Produkte erfordert ein grundlegendes Umdenken im gesamten Entwicklungsprozess. Dabei wird die klassische, dokumentenbasierte Entwicklung zunehmend von einer durchgängigen, modellbasierten Vorgehensweise abgelöst. Dies soll Medienbrüche und ineffiziente

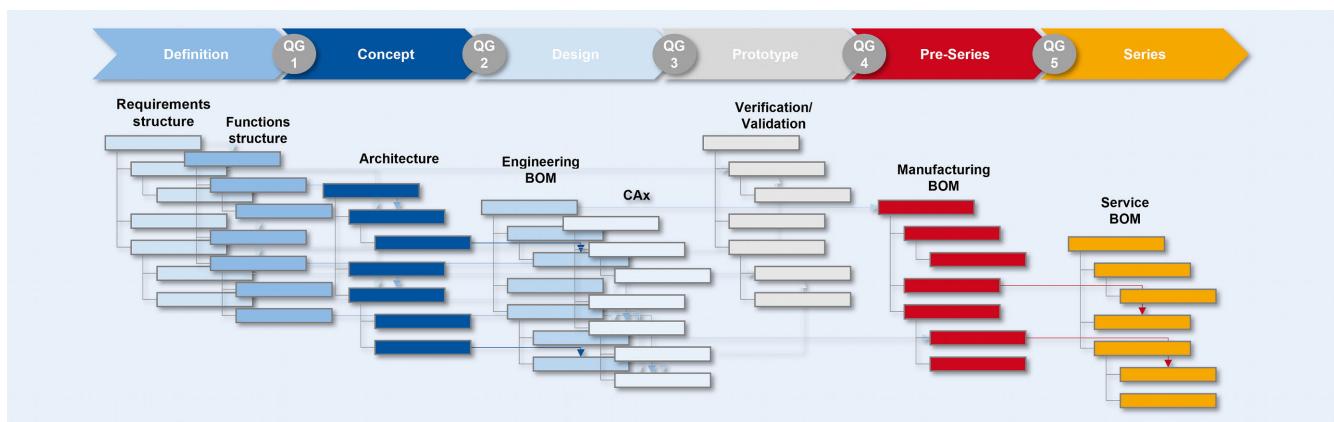


Bild 3. Modellbildung während der Entwicklungsphasen. *Grafik: [9] in Anlehnung an Eigner 2021*

Informationsflüsse zwischen den unterschiedlichen Fachdisziplinen vermeiden. Ein Beispiel für ein im gesamten Entwicklungsprozess wachsendes Modell ist in **Bild 3** dargestellt. Das Modell bildet durchgängig die Basis für die Verifizierung und ist damit unerlässlich für eine frühzeitige Produktabsicherung. Besonders relevant ist dieses Vorgehen bei integrierten Entwicklungen mit parallelisierten Abläufen. Eine große Lücke zeigt sich aktuell in der Integration der CAE-Werkzeuge in den Entwicklungsprozess: Erst mit Beginn der Design-Phase wird der Entwicklungsprozess wieder durch digitale CAD- oder Simulationsmodelle und entsprechende Software-Lösungen unterstützt. Diese Prozess- und Informationslücke wirkt sich negativ auf das Entwicklungsergebnis aus – etwa in Form von höheren Entwicklungskosten und längeren Entwicklungszeiten. [15]

Ein wichtiger Ansatz zur Lösung der Systembrüche ist das „Model-Based Systems Engineering“ (MBSE). Es schließt die Modellierung der Anforderungen und der Systemarchitektur in ein ganzheitliches und digitales Vorgehensmodell ein. Gleichzeitig koordiniert es die Anforderungen und Spezifikationen während des gesamten Lebenszyklus. [12] Als geeignete grafische Modellierungssprache, die das MBSE unterstützt, hat sich SysML (Systems Modeling Language) etabliert. Damit lassen sich Systemanforderungen und deren Beziehungen zueinander in sogenannten Anforderungsdiagrammen visualisieren, aus denen sich wiederum spezifische Anforderungen für Subsysteme und Komponenten ableiten lassen. Über die grafische Darstellung hinaus ermöglichen Systemmodelle eine automatisierte Verknüpfung und Nachverfolgung der verschiedenen Systemartefakte. [9] Das Systemmodell dient als Basis für eine frühzeitige Verifikation und Validierung des Systems, indem es formale Simulationsmodelle integriert. Darüber hinaus bildet es die Ausgangsplattform für das anschließende disziplinspezifische Design, etwa in den Bereichen der mechanischen Konstruktion (M-CAD), der elektrischen Planung (E-CAD) oder der Software-Entwicklung (CASE). [9]

3.3 Umsetzung in einer Tool-Umgebung

Bei der Entwicklung komplexer Systeme stößt die Arbeit mit Anforderungen mittels herkömmlicher Office-Tools sehr schnell an ihre Grenzen. Ein konsistentes, durchgängiges Management voneinander abhängiger Einzelanforderungen lässt sich bei steigender Systemkomplexität und -dynamik auf diese Weise kaum noch effizient umsetzen. [5] Die Tool-Umgebung stellt den Werkzeugkoffer dar, mit dem sowohl der definierte Entwicklungs-

prozess als auch das zugrunde liegende Datenmodell umgesetzt werden müssen. Zwar ist es sinnvoll, die Auswahl der Software auf Basis von Anforderungen vorzunehmen, die sich aus dem gewählten Entwicklungsvorgehen ableiten – gleichzeitig müssen die Tools jedoch in der Lage sein, diese Anforderungen auch technisch abzubilden. In der Praxis entsteht daraus häufig ein Henne-Ei-Problem: Die gewählten Werkzeuge beeinflussen wiederum das Vorgehen, indem sie eigene Restriktionen mit sich bringen. Daher empfiehlt es sich, die Tool-Auswahl zunächst auf eine überschaubare Anzahl grundlegender Anforderungen zu stützen. Anschließend sollte die Expertise der Software-Administration möglichst frühzeitig in die Ausarbeitung und Feinabstimmung des Prozesses eingebunden werden.

Eine Möglichkeit besteht in der Koexistenz von „Product Lifecycle Management“ (PLM) und „Application Lifecycle Management“ (ALM), wie sie beispielhaft im Forschungsprojekt „SeLv“ umgesetzt wurde (**Bild 4**). Dabei verteilen sich die Entwicklungsprozesse auf die verschiedenen Systeme. Bei dieser Auswahl benötigen das Anforderungsmanagement, die Modellentwicklung und das „Computer-Aided Software Engineering“ (CASE) einen definierten „Backbone“, der als „Single Source of Truth“ dient. Standards wie OSLC (Open Services für Lifecycle Collaboration) können ALM- und PLM-Inhalte miteinander koppeln. Andere Möglichkeiten können im Verzicht auf das ALM- oder das PLM-System liegen. Dies erfüllt jedoch nur bedingt den geforderten Umfang. Im PLM-System sind ALM-Funktionalitäten nur rudimentär vorhanden und erschweren insbesondere die Nachvollziehbarkeit von Artefakten. Im Gegensatz dazu fehlen bei einem Verzicht auf einen PLM-„Backbone“ die Möglichkeiten der vollständigen Strukturbeschreibung mechatronischer Systeme. [16]

Bei der Koexistenz von ALM und PLM in einem integrierten Produktenwicklungsprozess übernimmt das ALM-Tool die Aufgaben des Anforderungs- und Testmanagements für alle beteiligten Disziplinen sowie der Problemverfolgung. Die Anwendung des PLM-Tools fokussiert sich auf das Stücklisten-Management und die Verwaltung von Hardware-Daten. [16, 17]

4 Fazit

Es lässt sich eindeutig ableiten, dass das Anforderungsmanagement den Grundstein für die erfolgreiche Absicherung von Produkten bildet. Angesichts der zunehmenden Komplexität cyber-physischer Systeme und der wachsenden Bedeutung von Software ist eine Weiterentwicklung etablierter Methoden und Prozesse

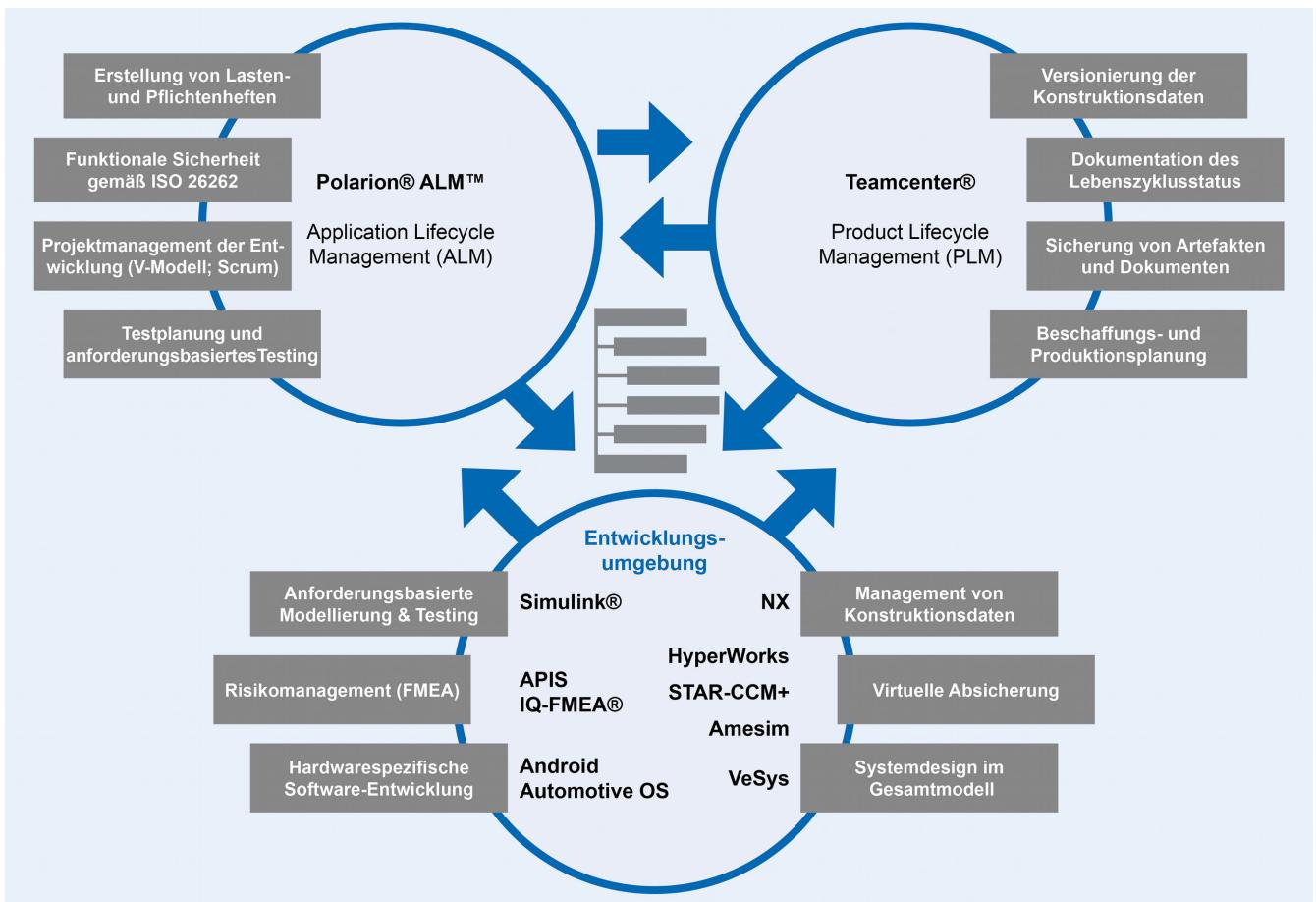


Bild 4. Tool-Landschaft im Forschungsprojekt „SeLv“. Grafik: eigene Darstellung

unerlässlich, um die Vollständigkeit, Konsistenz und Nachvollziehbarkeit von Anforderungen über alle Disziplinen hinweg sicherzustellen. Diese Weiterentwicklung muss ganzheitlich in den Bereichen der Prozessgestaltung, der Modellierung und der Tool-Landschaft erfolgen und kann nur in ihrer Gesamtheit den Herausforderungen wirksam begegnen.

L I T E R A T U R

- [1] Schuh, G.; Zeller, V.; Stich, V.: Digitalisierungs- und Informationsmanagement. Berlin: Springer 2022
- [2] Brüggemann, H.; Bremer, P.: Grundlagen Qualitätsmanagement. Von den Werkzeugen über Methoden zum TQM. Wiesbaden: Springer Vieweg 2020
- [3] Verein Deutscher Ingenieure e.V.: VDI/VDE 2206 - Entwicklung mechanischer und cyber-physischer Systeme. Berlin: Beuth Verlag GmbH 2021
- [4] Mayer-Bachmann, R.: Integratives Anforderungsmanagement. Konzept und Anforderungsmodell am Beispiel der Fahrzeugentwicklung. Universitätsverlag Karlsruhe 2007
- [5] Bender, B.; Gericke, K.: Pahl/Beitz Konstruktionslehre. Berlin: Springer 2021
- [6] Mistler, M.: Entwicklung eines Vorgehenskonzeptes zum modellbasierten agilen Anforderungsmanagement (Requirements Engineering und Requirements Management) für Organisationen – REMOt, 2021
- [7] Schäuffele, J.; Zurawka, T.: Automotive Software Engineering. Wiesbaden: Springer Fachmedien 2016
- [8] Ackermann, M.: Mobility-as-a-Service. The Convergence of Automotive and Mobility Industries. Cham: Springer International Publishing; Imprint Springer 2021
- [9] Eigner, M.: System Lifecycle Management. Berlin: Springer 2021
- [10] Pischinger, S.; Seiffert, U.: Vieweg Handbuch Kraftfahrzeugtechnik. Wiesbaden: Springer Fachmedien 2021
- [11] Schlüter, N.: Generic Systems Engineering. Ein methodischer Ansatz zur Komplexitätsbewältigung. Berlin: Springer Vieweg 2023
- [12] Löckel, K.: Systems Engineering – Ganzheitlicher Ansatz mit großem Potenzial. In: ATZproduktion 2018, S. 18–23
- [13] Haberfellner, R.; Weck, O. L. de; Fricke, E. et al.: Systems Engineering. Fundamentals and Applications. Cham: Birkhäuser 2019
- [14] Hellnerbrand, D.: Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse. München: Dr. Hut 2013
- [15] Cadet, M.; Schulze, T.; Dickopf, T. et al.: Modellbasierte Entwicklung cybertronischer Systeme in der frühen Phase des Entwicklungsprozesses – Ein Vergleich mit der klassischen Produktentwicklung. In: Stuttgarter Symposium für Produktentwicklung 2017, S. 1–10
- [16] Gerhardt, F.: Framework zur Einführung des Systems Engineering auf Basis von PLM/ALM Lösungen in Enterprise Architekturen. In: Produkt-Daten Journal 2016, S. 48–52
- [17] Deuter, A.; Otte, A.; Ebert, M. et al.: Developing the Requirements of a PLM/ALM Integration: An Industrial Case Study. In: 4th International Conference on System-Integrated Intelligence 2018, S. 107–113

M i c h a e l D e m m i n g

m.demming@pem.rwth-aachen.de

Chair of Production Engineering
of E-Mobility Components (PEM)
RWTH Aachen University
Bohr 12, 52072 Aachen
www.pem.rwth-aachen.de

LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons
Namensnennung 4.0 International (CC BY 4.0)