



Levin • Stathatou • Lehmann • Benning

Handbuch Automotive SPICE[®] 4.0

Grundlagen und Know-how für die Praxis

dpunkt.verlag

Geleitwort

Seit etwa 20 Jahren engagieren sich Assessoren und Interessierte in der Automobilindustrie, um die Qualität in der Softwareentwicklung zu verbessern. Automotive SPICE®, dessen erste stabile Version 2007 veröffentlicht wurde, liegt inzwischen in der vierten Version vor. Seither wurden mehr als 10.000 Assessoren weltweit ausgebildet und etwa ebenso viele Assessments durchgeführt.

Die Anwendung des Modells erfolgte bislang in zwei Phasen. Phase eins ab der frühen 2000er-Jahre war die, in der die Entwicklung von Fahrzeugen mit 40–60 eingebetteten Steuergeräten im Fokus stand, beginnend bei den Oberklasse-Fahrzeugen. Das Ziel war die Beherrschung der Komplexität von elektronischen Systemen im Fahrzeug, Entwicklungsleistungen sollten nun einen definierten Qualitätsstand aufweisen. Die OEMs beschränkten sich weitgehend auf die Integration und den Systemtest bzw. das »Freifahren« der Funktionen, meist im Dauerlauf. Erste »Hardware in the loop«-Systeme (HIL-Systeme) wurden angeschafft. Bezüglich der Prozessqualität fokussierten sich die OEMs im Wesentlichen darauf, ihre Lieferanten zu bewerten und qualifizieren zu lassen. Damit setzten sie den Jobsplit aus den Jahren zuvor im Wesentlichen fort. Die eigene Entwicklungsmannschaft beschäftigte sich eher selten mit dem Thema Prozessqualität, CMM oder SPICE.

Das Thema Produktqualität geriet aufgrund der Erfahrungen, u.a. mit Softwareänderungen, schnell in den Fokus und mündete in verschiedenen Anforderungen der OEMs, die bis heute gültig sind. Zu nennen sind hier beispielsweise »Konzern Grundanforderungen Software« (KGAS) der Volkswagen AG.

Im Jahr 2001 entschieden die deutschen OEMs, ein eigenes Prozessassessment-Modell auf Basis der ISO/IEC 15504 zu entwickeln, was in der Entwicklung von Automotive SPICE® mündete. Dieses Modell erlangte seine heutige Bedeutung, als der Standard bei einigen OEMs Bestandteil der Lieferanteneinstufung und damit vergaberelevant wurde. Automotive SPICE® kam damit auf Augenhöhe zu den Bewertungen, die mit VDA 6.1 und 6.3 seit Jahren etablierte Praxis waren. Die Einstufungen von Entwicklungsprojekten auf Grundlage von Softwareprozess-Fähigkeitsbewertungen erfolgten zu dem Zeitpunkt, als bei der Mehrheit der bewertenden Firmen keine Prozesse mehr auf Level 0 festgestellt wurden und damit keine »C-Einstufung« ausgesprochen werden musste. Prozessseitig war nun der Grundstein für die Entwicklung komplexerer, höherwertiger Elektroniksysteme gelegt. Zwischenzeit-

liche Versuche und Initiativen, die Softwareentwicklung bzw. deren Steuerung bei den OEMs selbst zu verbessern, waren meist nicht erfolgreich. Dass dieses künftig ein Wettbewerbsnachteil sein würde, wurde häufig nicht erkannt. Diese Phase dauerte etwa bis 2015.

Mit der geplanten Entwicklung von Elektrofahrzeugen begann die Phase zwei. Das Erfordernis der Steigerung der Rechenleistung im Fahrzeug konnte nur durch eine stärkere Integration und Konzentration von Funktionen erreicht werden. Die geplanten Fahrerassistenzsysteme, Online-Dienste, elektrisches Laden und Fahren zwang die Hersteller zur Entwicklung neuer Systemarchitekturen und zur Verbindung mit dem Internet. Die plötzliche Angreifbarkeit der Systeme von außen erforderte technisch die Entwicklung von Domänenarchitekturen und Schutzmaßnahmen im Fahrzeug, wie z.B. verschlüsselte Kommunikation. Neue Computing-Plattformen mit Domänenrechnern und leistungsfähigeren Prozessoren zogen in die Fahrzeugarchitektur ein. Damit änderten sich auch die Spielregeln der Entwicklung und die Zusammenarbeitsmodelle. Ansätze aus Phase eins funktionierten nicht mehr. Die Zusammenarbeitsmodelle funktionierten teilweise nicht mehr, weil die Diskrepanz der Prozessreife zwischen OEMs und Lieferanten zu groß geworden war. Spätestens durch die Unterbrechung der Lieferketten ab 2020 wurden die Karten neu gemischt. Prozessoren konnten nun nicht mehr »wie Kartoffeln« eingekauft werden, die Verlagerung eines Großteils der Entwicklungsleistung zu den Lieferanten wurde grundsätzlich infrage gestellt. Die Abhängigkeit der OEMs von den Chipherstellern trat erstmals eindeutig zutage. Die Planung und Entwicklung von Domänenarchitekturen wurde unter Berücksichtigung der Funktionszuordnung und der einzusetzenden Prozessorfamilien ein wesentlicher Bestandteil der Architekturarbeit mit Auswirkungen auf die Softwareentwicklung. Es wurde festgestellt, dass der Wechsel von der Prozessorfamilie eines Chipherstellers auf einen anderen eine Wiederholung wesentlicher Arbeiten aus der Design- und Architekturphase erforderlich machte – und das geschah mit teils erheblichem Aufwand.

Wie wollte man diese Situation nun in den Griff bekommen? Seit den frühen 2000er-Jahren wurden hauptsächlich Lieferanten bewertet. Die Entwicklungsorganisationen der OEMs selbst gerieten nun in den Fokus. Es wurde erkannt, dass die Beschäftigung der eigenen Entwicklungsteams mit dem Thema Software und Software-Entwicklungsprozesse unausweichlich war und benachbarte Themen und Domänen entlang der Wertschöpfungskette mit betrachtet und einbezogen werden mussten. Jetzt war die Zeit gekommen für eine ganzheitliche Betrachtung der Entwicklungsleistungen, die Fokussierung auf Systems Engineering und neue SPICE-Initiativen. Das System befand sich nun nicht mehr mit abgeschlossener eingebetteter Architektur allein im Fahrzeug, sondern zwischen den Sensoren im Fahrzeug und dem IT-Rechenzentrum, beide mit Internetanschluss. Die Frage der Systemdefinition musste neu gestellt und beantwortet werden. Notwendig wurde eine erweiterte, integrative Betrachtung neuer technischer Domänen, z.B. Cybersecurity, und auch die diesbezügliche Ergänzung des Automotive SPICE®-Basismodells.

Der VDA QMC und iNTACS e.V. begleiten und unterstützen diese Entwicklung seit Jahren inhaltlich und methodisch durch neue Versionen von Automotive SPICE® und durch die Erstellung von Modellerweiterungen, die als integriertes System Betrachtungen und Bewertungen in einem erweiterten Scope entlang der Wertschöpfungskette ermöglichen. Um die Projektrisiken für softwarebasierte Systeme weiter zu reduzieren, soll dieses integrierte Modell ausgebaut und kontinuierlich weiterentwickelt werden. Weil der Erfolg von Entwicklungsprojekten wesentlich von den dort arbeitenden Teammitgliedern abhängt, können diese ab der Veröffentlichung von Automotive SPICE® Version 4.0 in der Anwendung der SPICE-Methodik geschult werden. iNTACS führt hierzu den Ausbildungspfad eines Process Expert neu ein. Automotive SPICE® 4.0 kann dadurch noch besser in Projekten angewendet werden.

Das neue intacs®-Schulungskonzept ermöglicht zudem ab 2024 die Auswahl von verschiedenen Basismodellen (PAM/PRMs) und Erweiterungen nach dem jeweiligen Bedarf einzeln oder in Kombination.

Der Erfolg der Entwicklungsorganisationen hängt auch heute noch davon ab, den Stand der Technik und Best-Practice-Ansätze sinnvoll anzuwenden. Standards und Normen sind ausreichend vorhanden. Der Grundsatz »Wie ich mein Projekt plane, so liege ich im Rennen« hat noch immer Gültigkeit. Es besteht damit nach wie vor die Möglichkeit und Notwendigkeit, die eigene Entwicklungsleistung systematisch zu verbessern.

Dieses Buch leistet dazu einen wichtigen Beitrag. Ich wünsche den Leserinnen und Lesern viel Erfolg beim Lernen und Anwenden des Wissens. Die Autorinnen und Autoren sind ausgewiesene Fachleute mit langjähriger praktischer Erfahrung auf diesem Gebiet.

Lars Dittmann

International Assessor Certification Scheme e.V.
Vice President

Vorwort

Das Buch, das Sie in den Händen halten, ist nicht in ein oder zwei Monaten entstanden, sondern hat uns bei unserer Arbeit als Assessoren und bei der Verbesserung von Prozessen, auf unseren beruflichen Pfaden bei Methodpark und Kugler Maag CIE sowie bei privaten Veränderungen, Auszeiten etc. begleitet. Nun sind wir froh, am Ziel angekommen zu sein und Ihnen dieses Buch präsentieren zu können. Natürlich sind wir diesen Weg nicht allein gegangen, und auf diejenigen, die uns dabei unterstützt haben, kommen wir in der Danksagung zu sprechen.

Uns verbindet nicht nur unsere berufliche Zugehörigkeit, sondern auch das Bestreben, die Prozesse unserer Kunden in der Automobilindustrie zu verbessern und den Nutzen von Automotive SPICE® als Messinstrument für Prozessqualität herauszustellen. Der Standard hat sich mit seiner neuesten Version 4.0 weiterentwickelt, um den aktuellen Stand von Forschung und Technik in der Automobilindustrie abzubilden. Er nimmt, wie auch schon die vorherigen Versionen, zunächst die gelebten Prozesse in den Blick, anstatt zuerst auf Prozessbeschreibungen und -dokumentation zu fokussieren.

Als Autorenteam haben wir uns zusammengefunden, um unser Wissen und unsere Erfahrungen in diesem Buch zu teilen. Alle Beteiligten bringen besondere Fähigkeiten und Perspektiven mit, die sich beim Schreiben dieses Buches ergänzt haben:

- Alexander Levin, aktiv in der iNTACS-Arbeitsgruppe »Functional Safety«, erweitert unser Verständnis von Automotive SPICE® um die kritische Komponente der funktionalen Sicherheit und durch sein profundes Wissen im Systems Engineering. Zudem bringt er einen Erfahrungsschatz aus vielen Jahren der Arbeit bei verschiedenen OEMs und Zulieferern bezüglich der Umsetzung von Automotive SPICE® mit, sowohl als Lead Assessor als auch als Prozessberater oder Experte für funktionale Sicherheit.
- Christina Stathatou hat maßgeblich an der Entwicklung des Prozess-Assessmentmodells (PAM) für Machine Learning mitgewirkt. Ihre Einblicke und Erfahrungen sind besonders wertvoll, da Machine Learning in der modernen Automobilentwicklung zunehmend an Bedeutung gewinnt. Sie hat langjährige Erfahrung im Management großer Prozessverbesserungsprogramme. In diesem Bereich war sie bei verschiedenen OEMs und Zulieferern erfolgreich tätig.

- Volker Lehmann ist Mitglied und ehemaliger Leiter der iNTACS-Arbeitsgruppe »New Training Courses and Coordination«. Seine umfangreiche Erfahrung in der Entwicklung und Koordination von Trainingskursen ermöglicht es uns, die theoretischen Grundlagen von Automotive SPICE® praxisnah zu vermitteln. Dabei wollen wir auch seine Arbeit in China erwähnen, wo er durch zahlreiche Trainings und spezielle Gate4SPICE-Events dazu beigetragen hat, die Methodik von Automotive SPICE® zugänglich zu machen. Darüber hinaus hat er sich seit Jahren als Assessor und Ausbilder von Assessoren einen Namen gemacht.
- Josefin A. Benning, ebenfalls Mitglied und derzeitige Leiterin dieser Arbeitsgruppe, bringt eine seltene Kombination aus technischer Präzision und pädagogischem Geschick in unser Team mit ein. Sie hat große Erfahrung im Management komplexer Prozessverbesserungskampagnen bei internationalen Unternehmen und Big Playern.

Unser gesamtes Team ist tief in der Automotive SPICE®-Community verwurzelt, auch durch die regelmäßige Organisation und Teilnahme an Gate4SPICE-Veranstaltungen. Die Zusammenführung unserer unterschiedlichen Qualifikationen und langjährigen praktischen Erfahrungen ermöglicht es uns, ein breites Spektrum an Perspektiven zu bieten, und bildet das Fundament dieses Buches.

Automotive SPICE® ist mittlerweile ein zentraler Bestandteil in der Entwicklung von OEMs und wird in Zukunft voraussichtlich sogar die Homologation beeinflussen. Seit der Version 3.1 hat sich die Norm kontinuierlich weiterentwickelt, insbesondere durch Erweiterungen in Bereichen wie Hardware, Datenmanagement oder Cybersecurity.

In diesem Buch verfolgen wir das Ziel, das Zusammenspiel aus neuen und alten Prozessen klar und logisch zu erklären, und legen Wert darauf, die Sprache der Norm zu entmystifizieren.

Wir möchten Sie in die Welt der neuen und überarbeiteten Prozesse einführen, deren Logik und Motivation erklären, erste Erfahrungen teilen und aus der Perspektive der Prozessverbesserung Hilfestellung bieten.

Aus eigenen Assessments und Prozessverbesserungsprojekten wissen wir, dass der Standard teilweise sehr unterschiedlich interpretiert wird. Unser Anliegen ist es, zu einem einheitlichen Verständnis beizutragen und durch praxisnahe Beispiele, Tipps und Tricks unsere langjährige, auch internationale Erfahrung mit Ihnen zu teilen. Ziel ist es, die oft komplexe Normensprache in verständliche Anleitungen zu übertragen, die es Prozessanwendern ermöglichen, die Anforderungen von Automotive SPICE® nicht nur zu verstehen, sondern auch aktiv zu gestalten und zu leben.

Neben den technischen Details ist es uns wichtig, auch die menschliche Seite der Prozessverbesserung zu betonen. Denn in gelebten Prozessen sind die Mitarbeitenden und ihr Verständnis der Prozesse wichtiger als das, was auf dem Papier steht.

Wir hoffen, dass Sie dieses Buch nicht nur als eine Sammlung von Richtlinien betrachten, sondern als einen lebendigen Begleiter in Ihrer beruflichen Laufbahn. Unabhängig davon, ob Sie ein erfahrener Assessor oder ein Neuling in der Welt von Automotive SPICE® sind, wir hoffen, dass dieses Buch Ihnen die Werkzeuge und das Wissen an die Hand gibt, um Ihre Prozesse zu meistern und Ihre Ziele zu erreichen. Wir laden Sie ein, gemeinsam mit uns die Möglichkeiten von Automotive SPICE® 4.0 zu erkunden und die Qualität Ihrer Projekte und Produkte zu transformieren.

Alexander Levin, Christina Stathatou, Volker Lehmann und Josefin A. Benning
Hamburg, Stuttgart, Köln, im Juni 2024

Danksagung

Ohne die Unterstützung vieler Kolleginnen und Kollegen wäre das Schreiben dieses Buches nicht möglich gewesen. Sicherlich werden wir hier einige vergessen zu erwähnen, wofür wir um Entschuldigung bitten.

An erster Stelle möchten wir Vesna Djordjevic, Fiona Zhang und Rocio Rojas nennen, die in der Anfangsphase das Buchkonzept mitentwickelt und auch erste Entwürfe zu einzelnen Kapiteln geschrieben haben. Diese Zusammenarbeit hat den Start des Projekts geprägt. Vielen Dank für diese große Unterstützung in der Anfangsphase des Projekts.

Besonders danken wollen wir weiter Bhaskar Vanamali, der uns sowohl als Mentor mit vielen Anregungen als auch als fachlicher Reviewer unterstützt hat. Insbesondere im Bereich des Machine Learning war er ein wichtiger Ansprechpartner.

Nicht zu vergessen ist auch Dominik Strube, der neben der Koordination mit dem Verlag mit uns geplant, strukturiert, diskutiert hat und viele wichtige Tipps zum Aufbau des Buches gegeben hat. Außerdem war er einer unserer eifrigsten Reviewer.

Unser Dank gilt auch Sandra Schnetzer, die sich unserer Abbildungsentwürfe angenommen hat.

Speziell in Bezug auf die Hardwareprozesse möchten wir uns bei Thomas Gabler und Kosmas Kopmeier für die fachliche Beratung und das Review bedanken.

Die Reviews von Christian Hübscher und Michael Vrbán haben ebenfalls zur Qualität dieses Buches beigetragen.

Auch möchten wir dem dpunkt.verlag, insbesondere unserer Lektorin Christa Preisendanz, für die Unterstützung danken sowie unseren externen Reviewern Lars Dittmann und Jörg Zimmer, deren Hinweise uns geholfen haben, dem Buch den letzten Schliff zu geben.

Lars Dittmann sei an dieser Stelle auch herzlich für sein Geleitwort zu diesem Buch gedankt.

In Zeiten großer Umstrukturierungen und der Zusammenführung ehemals eigenständiger Unternehmen wie Methodpark, Kugler Maag und kVA unter dem Dach von UL Solutions möchten wir uns auch bei unserem Management bedanken, das uns den notwendigen Freiraum für unser Schaffen gegeben hat. Stellvertretend geht unser Dank insbesondere an Steffen Herrmann, Peter Seidenschwang und Bonifaz Maag.

Wir danken auch unseren Partnerinnen und Partnern, Familien und Freunden, denn in ein solches Buchprojekt fließt nicht nur Arbeitszeit, sondern auch eine ganze Menge an Freizeit.

1 Einleitung

1.1 Wir stehen auf den Schultern von Riesen

Automotive SPICE® ist nicht erst gestern entstanden, sondern ein fester Bestandteil der Branche. Es stimmt, dass mit jeder aktualisierten Version von Automotive SPICE® Neuerungen einhergehen. Aber tief in seinem Kern behält das Prozessmodell die unerschütterlichen Grundprinzipien bei, die in früheren Publikationen herausgearbeitet wurden, insbesondere in »Automotive SPICE® in der Praxis« von Markus Müller, Klaus Hörmann, Lars Dittmann und Jörg Zimmer [Müller et al. 2016] sowie in »Automotive SPICE® Capability Level 2 und 3 in der Praxis« von Pierre Metz [Metz 2016].

Diese Werke bilden das Fundament, auf dem wir aufbauen. Weitere Bücher, von denen wir profitiert haben, sind die »AUTOMOTIVE SPICE Essentials« [Abowd et al. 2021] und »The guide for Automotive SPICE® Interpretation« [Hoermann et al. 2022]. Anstatt das Rad neu zu erfinden, erkennen wir den Wert des bereits Erreichten und streben danach, diesen Weg fortzusetzen.

Die vergangene Dekade hat einen Wandel in der Auffassung von Mobilität gebracht, geprägt von aufkommenden Themen wie dem autonomen Fahren und der Vernetzung von Fahrzeugen. Diese evolutionären Veränderungen fordern uns heraus, unsere Methoden ständig zu überdenken und zu erneuern. Es ist eine bemerkenswerte Zeit, in der die Werkzeuge der Entwicklung nicht nur moderner werden, sondern auch effizienter in ihrer Fähigkeit, Prozesse zu unterstützen.

Aber mit jeder technologischen Verbesserung steigen auch die Erwartungen und die Risiken. In einer Ära, in der Autos vernetzt sind, wird die Konformität mit Standards wie Automotive SPICE® unverzichtbar, um Risiken zu minimieren, insbesondere bei sicherheitskritischen Systemen, die das automatisierte Fahren unterstützen. Cybersecurity und Machine Learning sind zwei Themen, die unsere Arbeit und unsere Prozesse dominieren werden. Prozesse für das Machine Learning Engineering (MLE) sind Bestandteil von Automotive SPICE® 4.0 (s. Abschnitt 4.7 und Kap. 12) und für das Thema Cybersecurity gibt es eine Erweiterung des Standards (s. Kap. 19).

Die Relevanz von Automotive SPICE® ist so tief verwurzelt, dass inzwischen auch Fahrzeughersteller ihr Vorgehen nach seiner Konformität ausrichten.

Automotive SPICE® hat eine beeindruckende Historie, beginnend mit der Entwicklung durch die AUTOSIG im Jahr 2001 und der anschließenden Nutzung durch deutsche Automobilhersteller für die Lieferantenqualifizierung im Rahmen der Herstellerinitiative Software (HIS).

Die Entstehung dieses Buches war eine tiefgreifende Expedition – nicht nur durch die Facetten des Standards, sondern auch durch die unterschiedlichen Perspektiven, mit denen wir auf Automotive SPICE® blicken. Diese Reise hat uns geholfen, die Essenz von Automotive SPICE® 4.0 besser zu verstehen. Wir haben uns mit vielen visionären Kollegen und Experten ausgetauscht, um ein ganzheitliches Bild von Automotive SPICE® zu zeichnen. Unser Wunsch ist es, dass dieses Buch vielen als Leuchtturm dient, der den Weg zu exzellenten Prozessen in der Automobilindustrie weist.

1.2 Zielsetzung und Verwendung dieses Buches

In einer Ära, in der sich Technologie und Fortschritt in der Automobilindustrie mit atemberaubender Geschwindigkeit verändern, bilden zuverlässige Prozesse das Rückgrat der Entwicklung. Von Steuergeräten über Komponenten bis hin zum integrierten Fahrzeug – Qualität ergibt sich auch aus dem Prozess. Tatsächlich könnte man sagen, dass ohne die Integrität des Prozesses keine außergewöhnliche Ingenieurleistung möglich ist. Dieses Buch richtet sich an jene, die an vorderster Front in dieser Industrie arbeiten: Systemingenieure, Projektmanager und viele andere. Es bietet eine Landkarte, die den Weg zur Prozessgüte weist. Ein Schwerpunkt liegt dabei auf der praktischen Umsetzung von Automotive SPICE® in Forschungs- und Entwicklungsprojekten.

Doch was genau ist dieses viel zitierte Automotive SPICE®? Es ist nicht nur ein detaillierter Bauplan, sondern vielmehr ein Instrument, das die Reife unserer Projekte misst. Das ist damit gemeint, wenn wir von einem Assessmentmodell sprechen.

Die Flexibilität von Automotive SPICE® ist bewundernswert. Es stellt Erwartungen auf, definiert jedoch nicht starr, wie diese zu erfüllen sind. Es vertraut auf die Innovation und Kreativität der Ingenieure und legt den Fokus auf die gelebten Prozesse. Und während es den Unternehmen die Freiheit lässt, ihre eigenen Pfade zu zeichnen, dient es gleichzeitig als Barometer, um den Erfolg dieser Wege zu messen.

Automotive SPICE® ist also keine Blaupause, die man einfach einem Projekt überstülpen kann, stattdessen gibt es Anstöße, die eigenen Prozesse zu hinterfragen und zu verbessern.

Man könnte meinen, dass ein solch flexibles Modell zu Chaos führt. Doch paradoxerweise ist es gerade diese Flexibilität, die Struktur schafft. Wenn man die Basispraktiken, die Automotive SPICE® vorgibt, versteht, wird einem bewusst, dass es nicht nur um das Einhalten von Regeln geht, sondern um ein echtes, tiefes Verständnis von dem, was es bedeutet, Systeme zu entwickeln.

Mit diesem Buch wollen wir Brücken bauen: zwischen den konkreten Anforderungen von Automotive SPICE® und der täglichen Arbeit von Ingenieuren, Entwicklern und Projektmanagern. Es dient nicht nur als Nachschlagewerk, sondern auch als Kompass, der Ihnen hilft, durch die vielfältigen Aspekte des Systems Engineering zu navigieren. Es beleuchtet nicht nur, wie Prozesse funktionieren, sondern auch, warum sie existieren und wie sie miteinander verwoben sind, um ein aufeinander abgestimmtes Ganzes zu bilden.

Für die Assessoren unter Ihnen, die als Hüter der Automotive SPICE®-Konformität fungieren, enthält dieses Buch ebenfalls wertvolle Ressourcen. Es bietet einen Leitfaden, der nicht nur das »Was«, sondern auch das »Warum« erläutert, und gibt Tipps und bewährte Verfahren für Ihre Assessments.

Schließlich ist dieses Buch mehr als nur eine Ansammlung von Worten und Konzepten. Es ist eine Einladung – eine Einladung, tiefer zu graben, mehr zu lernen und vor allem den wahren Wert und die Bedeutung von Automotive SPICE® in Ihrer täglichen Arbeit zu erkennen. Es ist ein ständiger Begleiter auf Ihrer Reise durch die Welt des Systems Engineering, der Sie dazu ermutigt, über den Tellerrand hinauszuschauen und das volle Potenzial Ihrer Prozesse auszuschöpfen.

1.3 Kapitelüberblick

In Kapitel 2 stellen wir Automotive SPICE® 4.0 vor, bevor wir in Kapitel 3 einen Überblick über Automotive SPICE® und seine Elemente geben.

Kapitel 4 befasst sich dann damit, wie sich Automotive SPICE® 4.0 gegenüber seinem Vorgänger weiterentwickelt hat und was die wesentlichen Änderungen sind.

Kapitel 5 erläutert die wesentlichen Konzepte, auf die wir bei Automotive SPICE® treffen, wie Rückverfolgbarkeit (Traceability) und Konsistenz, Kommunikation und Transparenz, aber auch die Nachvollziehbarkeit bei der Ausführung der Prozesse. Außerdem wird erläutert, was Automotive SPICE® unter einem System versteht, und eine erste Einführung in das Systems Engineering gegeben.

In Kapitel 6 werden verschiedene prozessübergreifende Themen erläutert, die uns später in den Beschreibungen der Prozesse wieder begegnen werden. Hier geht es beispielsweise um Verifikation und Validierung.

In den Kapiteln 7 bis 17 stellen wir dann die einzelnen Prozessgruppen im Detail mit den in ihnen strukturierten Prozessen vor:

- Kapitel 7 beschäftigt sich mit dem Zulieferermanagement und beleuchtet diese Prozessschnittstelle genauer.
- In Kapitel 8 geht es um das Produktrelease und was dabei zu beachten ist.
- Mit Kapitel 9 steigen wir direkt in die Entwicklungsprozesse ein und beginnen auf der Systemebene. Hier werden dem Anforderungsmanagement und dem Architekturentwurf die zugeordneten Verifikationsprozesse gegenübergestellt.

Diese Prozessgruppe stellt (ohne den SYS.1) das erste Plug-in für einen Assessment-Scope dar.

- In Kapitel 10 betrachten wir die Entwicklungsprozesse auf der Softwareebene, von den Anforderungen über die Architektur, den Entwurf, die Implementierung und die entsprechenden Verifikationsprozesse.
- In Kapitel 11 befasst sich mit der Validierung, die als Prozessgruppe mit nur einem Prozess einen starken Bezug zum SYS.1 aufweist.
- In Kapitel 12 gehen wir dann auf die Prozesse des Machine Learning ein, die in Automotive SPICE® neu hinzugekommen sind.
- Ebenfalls neu in Automotive SPICE® 4.0 sind die Prozesse der Hardwareentwicklung in Kapitel 13.
- Kapitel 14 beschreibt dann die unterstützenden Prozesse, wie die Qualitätssicherung, das Konfigurationsmanagement, Problemlösungs-Management, Änderungsmanagement und das Datenmanagement für das Machine Learning.
- Das Automotive SPICE®-konforme Projektmanagement wird dann in Kapitel 15 mit seinen drei Prozessen dargestellt.
- Danach folgen in Kapitel 16 das Thema Prozessverbesserung und in Kapitel 17 das Thema Wiederverwendung.

In Kapitel 18 gehen wir auf die Frage ein, was es bedeutet, einen bestimmten Automotive SPICE® 4.0-Level zu haben, und betrachten die Fähigkeitsdimension (Capability-Dimension) des Modells näher.

Auf Modellerweiterungen wie Plug-ins und Add-ons kommen wir dann in Kapitel 19 zu sprechen. Dabei geht es u.a. um die Erweiterung Automotive SPICE® for Cybersecurity, die zunehmend an Bedeutung gewinnt und immer häufiger assessiert wird. Darauf, dass Automotive SPICE® nicht für sich allein in Fahrzeugprojekten umgesetzt wird, sondern auch andere Standards, wie z.B. die ISO 26262 [ISO 26262], in der Automobilindustrie eine Rolle spielen, wird in Kapitel 20 genauer eingegangen. Wir skizzieren in diesem Zusammenhang auch das Zusammenspiel von Automotive SPICE® mit den betrachteten Standards.

In Kapitel 21 gehen wir dann näher auf das Thema Assessment ein und beschreiben, was dabei zu beachten ist. Darüber hinaus geben wir Tipps aus der Praxis, die aber auch bei den Beschreibungen der einzelnen Prozesse in den Kapiteln 7 bis 17 zu finden sind.

4 Neuerungen in Automotive SPICE® 4.0

Im Automobilsektor gibt es ein ständiges Bestreben, die Prozesse und Arbeitsmethoden zu verbessern und zu verfeinern. Mit der Veröffentlichung von Automotive SPICE® 4.0 erleben wir einen weiteren Schritt in der Gestaltung und Bewertung von Prozessen, der sicherstellen soll, dass die Branche weiterhin auf dem neuesten Stand bleibt.

In diesem Kapitel möchten wir ein wenig Licht in die Neuerungen bringen und erläutern, was sie aus unserer Sicht bedeuten. Ausführliche Erläuterungen finden Sie im weiteren Verlauf dieses Buches. Wir versuchen allerdings, Automotive SPICE® 4.0 so darzustellen, wie es ist, und nicht ein Buch im Stil von »was früher besser war« und »was heute besser ist« zu schreiben.

Bei einem Überblick über die Veränderungen fällt auf, dass nicht weniger als zehn Prozesse aus dem Prozessassessment-Modell (PAM) entfernt wurden. Dies ist ein deutliches Zeichen dafür, dass der Ansatz gestrafft und effizienter gestaltet wurde. Die in Automotive SPICE® 4.0 nicht mehr enthaltenen Prozesse können bei Bedarf noch aus Automotive SPICE® 3.1 verwendet werden.

Im Gegenzug wurden aber auch zehn relevante Prozesse und drei Prozessgruppen hinzugefügt, die den Fokus auf Hardwareentwicklung und Machine Learning erweitern. Diese Änderungen sollen sicherstellen, dass das Modell sowohl den aktuellen Anforderungen als auch den zukünftigen Herausforderungen gerecht wird.

Alle PAM-Prozesse wurden überarbeitet, um sie an den neuesten Industriestandards und Best Practices auszurichten. Dies beinhaltet auch ein aktualisiertes Bewertungsmodell, das auf der ISO/IEC 33020:2015 basiert. Diese Norm gibt klare Richtlinien vor, wie die Qualität und Effizienz von Prozessen gemessen und bewertet werden sollten, und das neue Bewertungsmodell spiegelt dies direkt wider.

Ein weiteres bemerkenswertes Merkmal der Änderungen ist, dass alle planungsbezogenen Aspekte vollständig in den Capability Level 2 verschoben wurden. Darauf wird in Abschnitt 4.4 näher eingegangen. Die Umstrukturierung der generischen Praktiken im Capability Level 3 (die Semantik der Ebene bleibt unverändert) soll bewirken, dass die Prozessattribute PA 3.1 und 3.2 besser miteinander harmonisieren und ihr Zusammenspiel deutlicher wird.

Diese Änderungen zielen darauf ab, das PAM effektiver und effizienter zu gestalten und eine genaue und umfassende Bewertung der in der Automobilindustrie verwendeten Prozesse zu ermöglichen.

4.1 Klarheit in der Terminologie

Wie bei jeder Sprache oder jedem System gibt es auch bei Automotive SPICE® eine Evolution der Terminologie. Die Worte, die wir wählen, und wie wir sie verwenden, können die Art und Weise, wie Prozesse durchgeführt und interpretiert werden, maßgeblich beeinflussen.

Automotive SPICE® 4.0 legt großen Wert auf klare und eindeutige Begrifflichkeiten, um Missverständnisse zu vermeiden und eine bessere Übereinstimmung mit anderen Standards, wie ISO 26262, zu gewährleisten. Beispiele für neue Begriffsbestimmungen:

- Ein konsistenter und präziser Gebrauch der Terminologie ist eingeführt worden, wobei Unterschiede zwischen »Maßnahme«, »Metrik« und »Handlung« klargestellt werden.
- »Integrierte Software« in SWE.6 bezieht sich jetzt auf das technische Softwareprodukt zur Überprüfung und nicht auf die Einbindung oder Dokumentation.
- Der Begriff »Item« wurde in den Test-/Verifikationsprozessen durch »System-/Softwareelement« ersetzt, um Konflikte mit der ISO 26262 zu vermeiden.
- Es wurde ein Unterschied zwischen »Betroffene Partei« und »Beteiligte Partei« gemacht, wobei letztere eine erweiterte Gruppe von Stakeholdern beinhaltet.
- Bei den erzeugten Informationsobjekten wurde das Wort »record« durch »evidences« ersetzt, und es gibt keine explizite Erwähnung von »aktualisierten Arbeitsprodukten« in den Kommunikations-BPs mehr.
- Der Begriff »Element« ist hier ein allgemeiner Begriff für »Unit« oder »Komponente«. Eine bemerkenswerte Ergänzung ist SWE.5.BP4, die sich auf die Integration und Verifizierung von Elementen bezieht. Diese beinhalten die Integration und Verifizierung von Einheiten in Komponenten.
- Das Konzept der »Units« wurde neu definiert. Interessanterweise kann eine Unit sowohl eine einzelne Unterfunktion als auch eine Gruppe von Unterfunktionen sein, je nach Anwendungsdomäne. Das bedeutet, dass sie eine »unteilbare kohärente Verhaltensweise« darstellt. Die Grenze einer Software-Unit ist unabhängig von ihrer Darstellung im Quellcode, der Struktur des Code-Files oder der modellbasierten Implementierung. Sie wird vielmehr durch die Semantik der Anwendungsdomäne bestimmt.
- Eine weitere Änderung ist die Verwendung des Wortes »Verifikation« anstelle von »Test«. Während ein Test oft als eine physische Handlung oder die Durchführung eines Tests interpretiert wird, vermittelt der Begriff »Verifikation« das

Bild einer umfassenderen Überprüfung, die sowohl die physische als auch die konzeptionelle Überprüfung einschließt. Ebenso ersetzen »Verifikationsmaßnahmen« die »Testspezifikation« und »Anforderungen« ersetzen die »Spezifikation«.

- Der Begriff »Output Work Products« wird durch »Output Information Items« und »Work Product Characteristics« durch »Information Item Characteristics« ersetzt. Diese Änderungen, obwohl subtil, signalisieren eine Verschiebung hin zu einer präziseren und kontextbezogenen Art der Kommunikation. Wir übersetzen den Begriff »Output Information Items« mit »erzeugte Informationsobjekte« und gehen in Abschnitt 4.2 näher darauf ein.
- Auf den Systembegriff in Automotive SPICE® 4.0 wird genauer in Abschnitt 5.1 eingegangen. Ein System kann z.B. ein mechatronisches System oder ein Antrieb sein, wie ein Motor plus ECU (Electronic Control Unit). Es kann sich jedoch auch um eine Steuereinheit (ECU), einen Mikrocontroller oder ein System-on-Chip handeln. Noch komplexer wird es, wenn man an ein »Softwaresystem« denkt, das aus verschiedenen Softwareteilen besteht.

4.2 Erzeugte Informationsobjekte

In Automotive SPICE® 4.0 gibt es den neuen Begriff »Information Item« (II), der sich auf Merkmale oder Attribute, die während der Prozessimplementierung als potenzielle Ausgaben betrachtet werden können, bezieht. Das ISO 33060-Konzept des »Information Item Characteristic« (IIC) ersetzt die bisherigen Arbeitsprodukt-Charakteristiken, um die Rückverfolgbarkeit der zu bewertenden Prozesse zu verbessern. Hinzu kommt, dass viele Assessoren explizit nach den Arbeitsprodukten gesucht haben und nicht nach den Informationen, die sie enthalten sollen. Der Übergang zu der neuen Begrifflichkeit soll daher auch hier für mehr Klarheit sorgen.

Diese Merkmale dienen als Anhaltspunkt dafür, welche Attribute in einem bestimmten Beispielarbeitsergebnis zu suchen sind, um objektive Beweise zur Unterstützung der Bewertung eines bestimmten Prozesses bereitzustellen. Wir benutzen im Rahmen dieses Buches den Begriff »Informationsobjekt«, der die Arbeitsprodukte (Work Products) aus früheren Versionen des Standards ersetzt.

Automotive SPICE® 4.0 kennt eine Vielzahl von Informationsobjekttypen, die mit den in Tabelle 4–1 dargestellten Eigenschaften definiert werden.

Attribut	Bedeutung
Information Item Identifier	Kennnummer, mit der auf das Informationselement verwiesen wird
Information Item Name	Typischer Name, der dem Informationselement zugeordnet ist. Organisationen können diese Informationselemente unterschiedlich benennen. Das eigentliche Format oder der tatsächliche Name ist jedoch nicht von Bedeutung. Es kann auch vorkommen, dass Organisationen mehrere äquivalente Informationselemente haben, die die in einem Informationselementtyp definierten Merkmale enthalten. <i>Hinweis:</i> Das Mapping liegt weiterhin in der Verantwortung der Assessoren.
Information Item Characteristics	Genauere Erläuterungen des Informationsobjekts. Diese bieten Beispiele für mögliche Eigenschaften, die mit den Informationselementtypen verbunden sind. Assessoren können nach diesen in den von der Organisationseinheit bereitgestellten Beispielen suchen.

Tab. 4-1 Beschreibung von Informationsobjekttypen

Außerdem wurden zusätzliche Tabellen für die Zuordnung von BP und IIC zu Prozessergebnissen, die Integration der BPs zur Rückverfolgbarkeit und Konsistenz sowie ein neuer grundlegender Anwendungsbereich mit weniger Beispielen aufgenommen, um den Eindruck von Checklisten zu vermeiden und die Assessoren zu motivieren, sich mehr auf den tatsächlichen Assessmentkontext zu konzentrieren und eindeutige Begriffe zu verwenden.

Informationsobjekte bieten lediglich eine Hilfestellung bei der Bewertung oder Implementierung von Prozessen. In einem Assessment sollten sie nicht als verbindliche Checkliste verwendet werden. Wir erläutern die entsprechenden Informationsobjekte in diesem Buch im Zusammenhang mit den Prozessen, in denen sie Anwendung finden. Es ist zu beachten, dass spezifische Informationselementtypen normalerweise von Prozesseigentümern in Form von Vorlagen etc. erstellt und von Prozessumsetzern angewendet werden, um ein Ergebnis eines bestimmten Prozesszwecks zu erreichen.

Der Prozesskontext, wie die Anwendungsdomäne, der Geschäftszweck, das Entwicklungsverfahren, die Unternehmensgröße usw., spielt eine wichtige Rolle bei der Entscheidung, was für einen Prozess angemessen ist.

4.3 Änderungen bei den Basispraktiken

Automotive SPICE® 4.0 bringt Änderungen in den Basispraktiken mit sich, die tiefgreifende Auswirkungen auf die Art und Weise haben, wie Unternehmen ihre Software- und Systementwicklungs-Prozesse betrachten und implementieren.

- Eine wichtige Änderung, die das Fundament dieser Revision prägt, ist die Neuordnung des Inhalts der vorherigen Strategie-BPs. Diese wurden in andere BPs und in GP 2.1.1 verschoben, was Auswirkungen auf die Interpretation der Unterstützungs- und Verifikationsprozesse hat. Darauf wird in Abschnitt 4.4 näher eingegangen.
- Die Basispraktiken für Rückverfolgbarkeit (Traceability) und Konsistenz wurden wieder zu einer BP zusammengeführt, da Konsistenz ohne Rückverfolgbarkeit nicht gewährleistet werden kann und eine Kernaufgabe der Rückverfolgbarkeit die Überprüfung der Konsistenz ist.
- Auch für Verifizierungskriterien gibt es keine Basispraktik mehr. Diese sind jetzt Teil der System- und Softwareanforderungen.
- Die Notwendigkeit, alternative Architekturen zu bewerten, entfällt ebenfalls; stattdessen muss die gewählte Architektur begründet werden, d.h., es muss nachvollziehbar sein, anhand welcher Kriterien die Entscheidung für diese Architektur getroffen wurde.
- Bei den Architekturprozessen (z.B. SYS.3 und SWE.2) wurden die bisherigen BPs 1, 3 und 4 zu einer BP zusammengeführt, um ein klareres Verständnis von Architektur und Design zu ermöglichen. Zudem wurde eine neue BP eingeführt, die sich mit der Analyse der Architektur befasst, wobei auch Aspekte wie Cybersecurity, funktionale Sicherheit und Robustheitsanforderungen berücksichtigt werden.

4.4 Automotive SPICE® 4.0 und Strategien

Ein kontrovers diskutiertes Thema ist die Rolle von Strategien in Automotive SPICE® 4.0. Vergleicht man die Versionen 3.1 und 4.0, so wird eines deutlich: Während es in Automotive SPICE® 3.1 Prozesse ohne und mit Strategie gab, werden in 4.0 keine Strategien mehr benötigt, um Capability Level 1 zu erreichen. Jedoch sind sie für alle Prozesse ab Capability Level 2 erforderlich. Auch wenn keine Strategie mehr notwendig ist, bleibt ein systematischer Ansatz essenziell.

Für spezifische Prozesse wie SYS.4/5 und SWE.5/6 wird zwar keine Strategie mehr gefordert, aber einige Punkte, die bisher Teil der Strategie waren, sind für BP1 erforderlich. Hierzu gehören Techniken, Bestanden-/Nicht-bestanden-Kriterien, Eingangs- und Ausgangskriterien für die Verifikationsmaßnahmen, die erforderliche Reihenfolge oder Abfolge der Verifikationsmaßnahmen und das benötigte Verifikationsinfrastruktur- und Umgebungssetup. In diesem Kontext kann festgehalten werden, dass die Kernthemen einer Verifikationsstrategie weiterhin definiert werden müssen.

Auch wenn für den Capability Level 1 keine Strategien mehr erforderlich sind, sollten sie vor allem für die Unterstützungsprozesse in Betracht gezogen werden, um die notwendigen Informationen sinnvoll zu bündeln.

4.5 Unabhängigkeit der Prozesse

Diese Neuerung ist weniger ein Bestandteil des Standards selbst, sondern findet sich in den Guidelines [VDA 2023]. Sie wird aber in Zukunft einen deutlichen Einfluss auf die Assessmentpraxis haben.

Jeder Prozess soll in zukünftig unabhängig in seinen Prozessattributen bewertet werden. Dies impliziert, dass nur Schwächen des zu bewertenden Prozesses als Grundlage für eine potenzielle Abwertung herangezogen werden dürfen. Konkret bedeutet dies, dass eine Bewertung von PA 1.1 (Prozessattribut 1.1) mit P (partiell erfüllt) oder N (nicht erfüllt) für einen Prozess X nicht verwendet werden darf, um PA 1.1 des Prozesses Y abzuwerten, was bisher durchaus nicht unüblich war und einen eher »einfachen« Weg der Bewertung dargestellt hat.

Jetzt können nur die Basispraktiken (BP), die explizit auf einen anderen Prozess verweisen, wie z.B. die Konsistenz- und Rückverfolgbarkeits-BPs, aufgrund einer Schwäche in diesem anderen Prozess abgewertet werden, da diese die einzigen »Verbindungspunkte« zwischen den Prozessen sind.

Daraus folgt zunächst, dass eine simple Abwertung, weil der vorausgehende Prozess fehlerhaft ist und abgewertet wurde, nicht länger möglich ist. Die Abwertung muss also immer aus dem Prozess selbst heraus begründet werden.

Trotzdem bedeutet das aus unserer Sicht nicht, dass es z.B. bei unzureichenden Anforderungen im SYS.2 völlig ausreicht, den Architekturentwurf im SYS.3 korrekt aus diesen unzureichenden Anforderungen herzuleiten. Es muss zumindest ein systematischer Umgang mit diesem Problem gefunden werden, damit der Prozess seinen Zweck erfüllt. Insofern kann diese Schwäche auch immer noch dem Prozess SYS.3 zugeordnet werden.

Der logische Sinnzusammenhang der Prozesse bleibt aus unserer Sicht bestehen. Das heißt, wenn wesentliche Voraussetzungen für einen Prozess nicht erfüllt sind, kann er auch weiterhin nicht seinen Zweck erfüllen und die geforderten Arbeitsergebnisse liefern. Diese Einschätzung ist jedoch klar zu begründen und stellt aus unserer Sicht keine Verletzung der Bewertungsregel dar.

4.6 Hardwareprozesse

In der facettenreichen Landschaft des Hardware Engineering gibt es das »Hardware Engineering SPICE« als separates Prozessassessment-Modell (PAM) schon seit einiger Zeit. Bei der Integration in Automotive SPICE® 4.0 wurden bereits einige Verbesserungen berücksichtigt.

Der Grundgedanke hinter der Integration in Automotive SPICE® 4.0 war, eine konsistente Methode für die Handhabung von Prozessverbesserungen über mechanische Disziplinen hinweg zu ermöglichen – sei es System (SYS), Software (SWE) oder Hardware (HWE). Die folgenden Aspekte werden ausgeklammert: System-Level-Engineering oder Einkauf, mechanische Herstellung oder Produktionsprozesse.

4.7 Machine Learning

Wir möchten diesen Abschnitt mit einer persönlichen Anekdote einer Kollegin beginnen. Vor einigen Jahren wurde sie als Beraterin eingestellt, um das erste ADAS-Projekt eines Kunden zu unterstützen und Automotive SPICE® zu implementieren. Als sie in das Unternehmen kam, fand sie ein großes Team von brillanten Ingenieuren vor, die an der Entwicklung eines Produkts arbeiteten.

Der Beraterin wurde mitgegeben, dass es sich beim ADAS-Team um eine Gruppe von »Rebellen« handelte, die sich an keine Qualitätsregeln halten wollten. Also begann sie zu versuchen, das Projektteam dazu zu bringen, die Vorgaben von Automotive SPICE® zu erfüllen. Als Erstes forderte sie das Team auf, bei der Anforderungsdefinition die Regeln von SWE.1 einzuhalten. Aber das Team fing an, über Datenanforderungen, Business Cases und Testlogik als Anforderungen zu sprechen und wie diese sich von den klassischen Anforderungen unterscheiden würden. Dann wollte sie, dass das Team eine Architektur definiert. Die Antwort war, dass klassische UML-Diagramme und Schnittstellendefinitionen nicht ausreichend waren, um die Probleme zu beschreiben, mit denen sie bei der Architektur zu kämpfen hatten. Als sie das Team aufforderte, einen Feinentwurf zu erstellen, musste sie sich anhören: »Wir können gerne ein detailliertes Design für unsere 20 Codezeilen erstellen, aber das wird unser Modell nicht verbessern« – was ein Modell sein sollte, wagte sie nicht zu fragen!

Und dann wollte sie, dass das Team Unit Tests und Qualifikationstests durchführt. Aber das Team zeigte ihr Fahrsimulatoren und Autos, die mit aufgeschnallten Kameras um das Firmengebäude fuhren. Verzweifelt darüber, dass sie weder in der Lage war, dem Team bei der Verbesserung seiner Qualität zu helfen, noch die Implementierung von Automotive SPICE® einzufordern, fragte sie: »O.k., was brauchen Sie also, um die Qualität dieses Produkts zu verbessern?« Die klare Antwort lautete: »Gute Daten, ein gutes Machine-Learning-Modell und gute Bewertungsmetriken, um zu überprüfen und sicherzustellen, dass dem ML-Modell und im weiteren Verlauf dem ADAS-System auf den Straßen vertraut werden kann, eigenständig Entscheidungen zu treffen.« Leider gab es im Automotive SPICE®-Koffer nichts, was geholfen hätte, die Qualität dieser wesentlichen Aspekte des Produkts zu verbessern. Dies war eine sehr lehrreiche Erfahrung für unsere Kollegin, aber sie war mit einer solchen Erfahrung nicht allein: In den vergangenen Jahren haben mehrere Automotive SPICE®-Experten weltweit ähnliche Erfahrungen gemacht und kamen zu einer gemeinsamen Erkenntnis: Es ist ein Wandel in Automotive SPICE® erforderlich, wenn es der Maßstab für die Entwicklungsqualität in der Automobilindustrie bleiben soll. Aus dieser Notwendigkeit heraus wurde die Erweiterung »Machine Learning for Automotive SPICE®« entwickelt.

Bevor wir allerdings in die weiteren Details des Plug-ins einsteigen, werden wir zwei Fragen vorab beantworten:

- Was ist Machine Learning?
- Was sind die wesentlichen Unterschiede zwischen der typischen Entwicklung von automobilen, softwaregesteuerten Systemen und Machine Learning?

Fangen wir mit der ersten Frage an. »Machine Learning« oder »maschinelles Lernen« besteht aus einem dreistufigen Prozess: Problemdefinition, Training und Implementierung. Am Anfang steht das zu lösende Problem. Dieses sollte so strukturiert sein, dass es mittels Mustererkennung bewältigt werden kann. Die Problemdefinition bestimmt dabei wesentlich die Qualität des späteren Resultats. So müssen beispielsweise die richtigen Daten identifiziert und ein geeignetes untrainiertes Modell gewählt werden. Die Modellentwicklung beginnt mit einem untrainierten Modell, das mithilfe von Trainingsdaten trainiert wird. Zum Training werden dedizierte Datensätze aus einer großen Sammlung von Dateninstanzen definiert. Als Starthilfe müssen die Dateninstanzen kategorisiert oder mit einem Hinweis auf das gewünschte Ergebnis versehen sein. Das untrainierte Modell wird dann den Trainingsdaten ausgesetzt. Übertragen werden anschließend die unstrukturierten Informationen in ein standardisiertes Format, beispielsweise als Zuordnung zu den wahrscheinlichsten Verkehrszeichennamen. Die ML-Ingenieure erhalten mehrere trainierte Modelle. Diese entsprechen potenziellen Lösungswegen. Bei der Erkennung des Stoppzeichens orientiert sich Modellkandidat #1 an der prozentualen Verteilung von Rot, während #2 die achteckige Form zur Identifizierung nutzt. Die MLE-Ingenieurteams bewerten dann das Ergebnis, ohne zu wissen, welche Kriterien der Kandidat angewendet hat. Sie wählen den vielversprechendsten Modellkandidaten aus und verfeinern diesen auf der Basis von Versuch und Irrtum immer weiter. Dieser kuratierte Trainingsprozess wird als »überwachtes maschinelles Lernen« bezeichnet. Dabei bedarf es erhöhter Aufmerksamkeit, da im Datenkörper möglicherweise Informationen enthalten sind, die es in der Wirklichkeit nicht gibt. Wurden die Verbotszeichen im Trainingsdatensatz mit auffälligen Schrauben montiert, nutzt Kandidat #3 vielleicht besagte Schrauben als Marker. So etwas wird als »Data Bias« (verzerrte Daten) bezeichnet – in diesem Fall basiert die Identifikation auf einer vermeintlichen Objekteigenschaft. Der iterative Trainingsprozess wird so lange durchgeführt, bis eines der generierten Modelle imstande ist, die Dateninstanzen dem Verkehrszeichenkatalog zuzuordnen. Dieses Artefakt wird anschließend in die Anwendungssoftware implementiert. Es enthält z.B. einen Entscheidungsbaum, der zeigt, wie die Bildfarben mit den verschiedenen Verkehrszeichen verknüpft sind. Letztlich wird an einem unabhängigen Datensatz getestet, ob das Modell richtig funktioniert und keine falschen Daten – Stichwort Muttern und Schrauben – fokussiert.

Nun kommen wir zur zweiten Frage. Wie in der Anekdote schon angedeutet, sind einige Unterschiede fundamental: Die Datenanalyse und -verwaltung sowie das Training der Algorithmen sind die wichtigsten und kritischsten Aktivitäten. Die Datenqualität ist viel kritischer und komplexer als die Qualität des Quellcodes, und das maschinelle Lernen erfordert einen Forschungs- und Entwicklungsansatz mit Versuch-und-Irrtum-Zyklen.

Die Validierung und Verifikation von Modellen des maschinellen Lernens basieren auf dem Labeling von Daten, dem Training, dem Aufspüren von Data Bias und der Sicherstellung, dass die wichtigsten Leistungsindikatoren (Key Performance

Indicators, KPIs) erfüllt werden. Das Projektmanagement muss den großen Aufwand berücksichtigen, der für die Sammlung von Datensätzen und deren Kennzeichnung erforderlich ist. Der Forschungs- und Entwicklungscharakter des maschinellen Lernens muss berücksichtigt werden, da diese Art der Entwicklung wenig vorhersehbar ist. Dies ist ein Projektrisiko.

Da die Entwicklung von Modellen des maschinellen Lernens nicht deterministisch ist, können die typischen Ansätze zur Qualitätssicherung und Verifikation nicht angewendet werden. Stattdessen müssen Prinzipien wie Vertrauenswürdigkeit und Erklärbarkeit (ISO TR 24048) befolgt werden, um ein ML-System zu verstehen und zu bewerten. Es sollte klar sein, welche Datensätze für das Training, die Validierung und das Testen verwendet wurden.

Machine Learning wird sehr häufig bei der Entwicklung von ADAS (Advanced Driver Assistance Systems) verwendet. Aber auch ohne autonomes Fahren nimmt der Einsatz von Ansätzen des maschinellen Lernens in der Automobilindustrie rapide zu, etwa in den Bereichen Diagnose, Wartung und Vorhersage des Verschleißes von Getriebesystemen.

Die ML-Prozesse in Automotive SPICE® 4.0 wurden als Standard für die Bewertung der Prozessfähigkeit und der Fähigkeiten des maschinellen Lernens in der Automobilindustrie entwickelt. Sie decken einen ganzheitlichen Ansatz für die Entwicklung des maschinellen Lernens ab, der Datenmanagement, Algorithmus-Training, Validierung und Testen für die Entwicklung des maschinellen Lernens umfasst. Das Plug-in »Machine Learning for Automotive SPICE®« besteht aus fünf Prozessen: der Prozessgruppe Machine Learning Engineering (MLE) mit vier Entwicklungsprozessen und einem zusätzlichen unterstützenden Prozess für das Datenmanagement (SUP.11).

Das Plug-in ist eine Ergänzung der SWE-Prozesse. Im Rahmen der Erstellung der Softwarearchitektur wird entschieden, welche Softwareanforderungen durch ML-Komponenten umgesetzt werden sollen.

Im Prozess MLE.1 werden die Anforderungen an diese Komponenten sowie an die Datensätze, die zum Training und Testen der entsprechenden ML-Modelle verwendet werden, erstellt, analysiert und strukturiert.

Im Prozess MLE.2 werden die Architektur des ML-Modells sowie die initialen Hyperparameter-Werte definiert. Hyperparameter sind die Steuerelemente des ML-Modells (s. auch das Glossar ab S. 561).

Im Prozess MLE.3 werden Datensätze aus den in SUP.11 gesammelten und verarbeiteten Daten genutzt, um das ML-Modell zu trainieren und zu validieren.

Im Prozess MLE.4 werden weitere unabhängige Datensätze verwendet, um zu testen, ob erst das trainierte ML-Modell und anschließend das einsatzbereite ML-Modell die Anforderungen und die ML-Architektur erfüllen.

Im Prozess SUP.11 werden die Daten entsprechend den Datenanforderungen aus MLE.1 für das Training des ML-Modells sowie dessen Validierung und Test gesammelt und verarbeitet. Um die Datenqualität zu sichern, erfolgen des Weiteren Qualitätsprüfungen, die auf definierten Qualitätsmetriken basieren.

Auch wenn die Anwendung des MLE-Plug-ins erst nach der Definition der Softwarearchitektur startet, ersetzt das MLE-Plug-in die Prozesse SWE.3 und SWE.4 nicht. Der ML-Algorithmus besteht weiterhin aus Quellcode. Auch dieser Quellcode muss auf Grundlage eines Feinentwurfs entwickelt werden und die Unit-Verifikation durchlaufen.

Wenn das ML-Modell die Tests bestanden hat, wird es in die Gesamtsoftware integriert und im Rahmen von SWE.5 getestet.

4.8 Validierung

Am Ende dieser Reise durch die verschiedenen neuen Prozessgruppen steht die Validierungs-Prozessgruppe. Ihr Hauptziel ist es, den Beweis zu erbringen, dass das Endprodukt, das eine direkte Benutzerinteraktion ermöglicht, die Erwartungen an den vorgesehenen Gebrauch in seiner betrieblichen Zielumgebung erfüllt. Das Augenmerk liegt auf dem sogenannten vorgesehenen Gebrauch und richtet sich an die Endbenutzer des Produkts. Es wird jedoch ein großer Satz von Produkten im Fahrzeug ausgeschlossen, die nicht direkt an dieser Interaktion mit dem Endbenutzer beteiligt sind. Homologations- und gesetzliche Typgenehmigungs-Anforderungen sind Beispiele für Stakeholder-Anforderungen, die einer Validierung unterliegen. Fehlen gesetzliche Vorschriften, können die Erwartungen hinter der Validierung explorativer oder sogar subjektiver Natur sein – denken Sie an die Benutzererfahrung, das »Gefühl« oder die Einstellung während der Fahrtests.

Der einzige Prozess dieser Gruppe, VAL 1, kann in großen Teilen nur vom Auftraggeber und nicht vom Lieferanten ausgeführt werden, was sicherlich einer der Gründe ist, warum er nicht als SYS.6 definiert wurde. Auf diesen Punkt wird in Abschnitt 11.1 näher eingegangen.

12 Prozessgruppe Machine Learning Engineering

Diese Prozessgruppe wurde neu in Automotive SPICE® 4.0 aufgenommen und umfasst vier Prozesse. Hinzu kommt der Unterstützungsprozess SUP.11 Machine-Learning-Datenmanagement.

12.1 MLE.1 Anforderungsanalyse für Machine Learning

Der Zweck besteht in der Verfeinerung der mit dem Machine Learning zusammenhängenden Softwareanforderungen zu einem Satz von ML-Anforderungen.	Plug-in
<p>Erwartete Prozessergebnisse:</p> <ul style="list-style-type: none"> ■ Die ML-Anforderungen einschließlich der ML-Datenanforderungen werden auf Grundlage der Softwareanforderungen und der Komponenten der Softwarearchitektur ermittelt und spezifiziert. ■ Die ML-Anforderungen werden strukturiert und priorisiert. ■ Die ML-Anforderungen werden auf Korrektheit und Verifizierbarkeit analysiert. ■ Die Auswirkungen der ML-Anforderungen auf die ML-Betriebsumgebung werden analysiert. ■ Konsistenz und bidirektionale Rückverfolgbarkeit zwischen ML-Anforderungen und Softwareanforderungen sowie zwischen ML-Anforderungen und Softwarearchitektur werden hergestellt. ■ Die ML-Anforderungen sind vereinbart und allen betroffenen Parteien mitgeteilt worden. 	

12.1.1 Prozessbeschreibung

Beim maschinellen Lernen oder Machine Learning (ML) baut die Anforderungsanalyse auf den Softwareanforderungen auf, die im vorherigen Prozess der Softwareanforderungs-Analyse verarbeitet wurden, und verwendet die Elemente der Softwarearchitektur als Eingabe.

Die Ergebnisse dieser Analyse umfassen detaillierte, gut lesbare Spezifikationen, sowohl für funktionale als auch nicht funktionale Anforderungen des Machine Learning (ML-Anforderungen), einschließlich der Anforderungen an die ML-Daten.

Die Definition von ML-Anforderungen ist entscheidend für eine qualitativ hochwertige ML-Datenerfassung und -verarbeitung sowie für die Entwicklung eines zuverlässigen ML-Modells. Die ML-Anforderungen legen den strukturierten Rahmen fest, um sicherzustellen, dass die erhobenen Daten den spezifischen Bedürfnissen des ML-Modells entsprechen. Ein zentraler Aspekt sind die ML-Datenanforderungen, die Eigenschaften wie Datencharakteristika, Verteilungen, Genauigkeiten und Auflösungen definieren. Diese präzisen Vorgaben ermöglichen eine zielgerichtete Sammlung von Daten, die für das Training, die Validierung und das Testen des ML-Modells erforderlich sind.

Neben den technischen Aspekten umfassen die ML-Anforderungen auch Qualitätskriterien für die Datensammlung, wie z.B. Anforderungen an die Labeling-Qualität und die Integrität der Daten. Die Struktur und das Format der ML-Daten sollten ebenfalls detailliert beschrieben werden, um sicherzustellen, dass die Daten effektiv verarbeitet werden können.

Eine gründliche Analyse der ML-Anforderungen ist unerlässlich, um mögliche Fallstricke zu vermeiden. Unklare oder vage Formulierungen können zu Missverständnissen führen, die wiederum die Qualität der Datenerhebung und -verarbeitung beeinträchtigen können. Durch eine klare Kommunikation und Festlegung des Status der Anforderungen sowie die Einbeziehung von verschiedenen Stakeholdern, wie dem Software-Projektleiter, dem ML-Modell-Tester, aber auch einem Vertreter des Teams, das für die Daten verantwortlich ist, wird sichergestellt, dass die Datenbasis für das ML-Modell von guter Qualität ist.

12.1.2 Basispraktiken

MLE.1.BP1: Spezifikation der ML-Anforderungen. Nutze die Softwareanforderungen und die Softwarearchitektur, um funktionale und nicht funktionale ML-Anforderungen sowie ML-Datenanforderungen zu identifizieren und zu spezifizieren, die Datenmerkmale (z.B. Geschlecht, Wetterbedingungen, Straßenzustand innerhalb der Betriebsumgebung (Operational Design Domain, ODD)) und deren erwartete Verteilungen angeben.

Anmerkung 1: Zu den nicht funktionalen Anforderungen können relevante Merkmale der ODD und der KPIs wie Robustheit, Leistung und Grad der Vertrauenswürdigkeit gehören.

Anmerkung 2: Die ML-Datenanforderungen bilden den Input für SUP.11 Datenmanagement für das Machine Learning, aber auch für andere MLE-Prozesse.

Anmerkung 3: Nur im Falle der ML-Entwicklung stellen die Anforderungen der Stakeholder die Softwareanforderungen dar.

Wie in der Basispraktik schon erwähnt, entstehen hier nicht nur die uns bereits bekannten funktionalen und nicht funktionale Anforderungen, sondern auch sogenannte »ML-Datenanforderungen«.

Die Anforderungen an die ML-Daten sind von entscheidender Bedeutung, da Daten einen der wichtigsten Faktoren im ML darstellen. Während die Logik und die Mathematik typischer ML-Modelle oft relativ einfach sind, hängt die Leistung eines Modells in hohem Maße von den während des Trainings und Tests verwendeten Daten ab. Die ML-Datenanforderungen umfassen u.a.:

■ Datencharakteristika

Die Anforderungen an die Daten beinhalten die charakteristischen Merkmale und ihre erwarteten Eigenschaften, wie z.B. Verteilungen, Genauigkeiten, Auflösungen. So müssten beispielsweise Bilder von Verkehrszeichen Bedingungen wie Tages- und Nachtzeiten sowie verschiedene Wetterbedingungen wie Regen- und Schnee abdecken.

■ Verteilung der Datencharakteristika

Es wird auch die Verteilung der Datencharakteristika festgelegt, beispielsweise »10 % Tageszeit, 30 % Schnee etc.«. Die Verteilung muss so nahe wie möglich an der Verteilung in der Betriebsumgebung (ODD) liegen. Jede Abweichung kann zu einer fehlerhaften Mustererkennung im Feld führen, da ein ML-Modell die Mustererkennung in den meisten Fällen auf Basis von statistischen Daten berechnet.

■ Formatanforderungen

Die Struktur und das Format der ML-Daten müssen ebenfalls definiert werden, einschließlich der Einbettung von Metadaten. In den beiden obigen Beispielen könnte man sich vorstellen, dass es eine Anforderung gibt, die Daten mit Tages- und Nachtzeiten sowie Wetterbedingungen zu kennzeichnen.

■ Qualitätsanforderungen

Die Anforderungen an die Datenqualität, insbesondere die Qualität der Kennzeichnung und der Struktur, sind von großer Bedeutung.

Die ML-Datenanforderungen dienen als hauptsächlicher Input für den Prozess SUP.11.

Auch wenn die ML-Datenanforderungen der Protagonist dieses Prozesses sind, sollten weitere Arten von ML-Anforderungen nicht außer Acht gelassen werden:

■ Funktionale Bedingungen für Training und Test

Es werden die funktionalen Teile festgelegt, die für das Training und den Test des ML-Modells implementiert werden müssen. Ein Beispiel sind Einschränkungen der Trainingsumgebung, Programmiersprache(n), Einschränkungen, die sich aus den System- oder Softwarebedingungen (SYS.2, SYS.3, SWE.1, SWE.2) ergeben.

■ Hardwarebezogene ML-Funktionen

Anforderungen, die sich auf die Hardware beziehen, auf der das ML-Element ausgeführt werden soll, werden definiert.

■ Empfang von Signalen von elektronischen Sensoren

Anforderungen an den Empfang von Signalen von elektronischen Sensoren werden ebenfalls berücksichtigt.

■ Nicht funktionale Anforderungen

Nicht funktionale Anforderungen, wie Leistungs- und Qualitätsanforderungen, werden ebenfalls spezifiziert. Beispiele hierfür sind Anforderungen aus anderen Standards (z.B. ISO 26262 – Funktionale Sicherheit oder ISO/SAE 21434 – Cybersecurity Engineering) sowie Anforderungen an die Qualität des Outputs des ML-Modells.

Diese Anforderungen bilden den Input für die Prozesse MLE.2 bis MLE.4.

MLE.1.BP2: Strukturierung der ML-Anforderungen. Strukturiere und priorisiere die ML-Anforderungen.

Anmerkung 4: Beispiele für Strukturierungskriterien können Gruppierung (z.B. nach Funktionalität) oder Variantenidentifikation sein.

Anmerkung 5: Die Priorisierung kann entsprechend den Anforderungen des Projekts oder der Stakeholder erfolgen, z.B. durch die Definition von Release-Umfängen (s. SPL.2.BP1).

Dabei ist darauf zu achten, dass die Anforderungen so strukturiert sind, dass sie für die interne Organisation aussagekräftig sind und die Verteilung der Anforderungen auf die verschiedenen Interessenbereiche unterstützen, z.B. Datenmanagement, ML-Architektur, ML-Modell-Training, ML-Modell-Test usw., um sicherzustellen, dass jeder Bereich weiß, welche Anforderungen für ihn relevant sind und in seiner Verantwortung liegen.

Ein weiteres strukturierendes Element können Features sein, die das Modell abdecken muss. Angenommen, es handelt sich z.B. um ein Modell zur Verkehrszeichenerkennung, dann könnte man sich vorstellen, dass die Anforderungen zwischen generellen Anforderungen und spezifischen Anforderungen pro Verkehrszeichen unterscheiden.

Eine Möglichkeit, ML-Anforderungen zu priorisieren, besteht darin, sichtbar zu machen, für welches Release die Umsetzung der ML-Anforderung geplant ist. Innerhalb eines Release sollten Prioritäten gesetzt werden, um im Falle eines plötzlichen Ressourcenmangels schnell und angemessen reagieren zu können; so sollten beispielsweise komplexe Funktionen oder Sicherheitsfunktionen eine hohe Priorität erhalten.

Die Strukturierung der Anforderungen unterstützt die Übersicht und Priorisierung und ermöglicht ein schnelles Auffinden der für eine bestimmte Aufgabe relevanten Anforderungen.

MLE.1.BP3: Analyse der ML-Anforderungen. Analysiere die spezifizierten ML-Anforderungen einschließlich ihrer Abhängigkeiten zur Sicherstellung der Korrektheit, der technischen Machbarkeit und der Fähigkeit zum Testen von Machine-Learning-Modellen sowie zur Unterstützung des Projektmanagements bei der Projektschätzung.

Anmerkung 6: Siehe MAN.3.BP3 für Projektdurchführbarkeit und MAN.3.BP5 für Projektschätzungen.

Die ML-Anforderungen werden konsolidiert und zusammengefügt. Sie müssen umsetzbar sein und Qualitätskriterien erfüllen. Hier greifen also die Analyse der ML-Anforderungen (BP3) und deren Erstellung bzw. Verfeinerung (BP1) ineinander. Die Basispraktiken sind keinesfalls, wie in anderen Prozessen, als sequenzielle Handlungsanweisungen zu verstehen, sondern müssen sehr oft iterativ in kleinen Zyklen durchgeführt werden. Dieses Zusammenspiel sollte nicht losgelöst von der Software- und ML-Architektur erfolgen, sodass erst am Ende die Machbarkeit und andere Kriterien, wie z.B. Überprüfbarkeit und Wartbarkeit, bewertet werden können. Aus der Betrachtung der Architektur ergeben sich wertvolle Hinweise für die Spezifikation von ML-Anforderungen.

Eine gute ML-Anforderungsanalyse ist in diesem Prozess ein Kernaspekt der Ingenieurarbeit. Der Umfang und die Angemessenheit der Analyse und ihrer Dokumentation hängen vom Kontext ab. Sie müssen durch Nachweise gestützt werden können. Letztlich muss zumindest die technische Machbarkeit und Überprüfbarkeit bewertet werden. Bei einem ML-Modell ist es durchaus denkbar, dass im Rahmen der Machbarkeitsanalyse auch ein Proof of Concept zum Tragen kommt. So könnte in Falle der Verkehrsschildererkennung zunächst ein Modell erstellt werden, das einen Typ von Verkehrszeichen bei Tageslicht und optimalen Lichtbedingungen erkennt.

Die ermittelten ML-Anforderungen werden auf bestehende Abhängigkeiten geprüft. Bei komplexen, verteilten ML-Funktionen bilden einzelne ML-Modelle die Grundlage für andere. Die Gesamtfunktion kann also von mehreren Teilen abhängen. Diese müssen identifiziert und später beschrieben werden. Es ist eine möglichst umfassende Beschreibung einer ML-Funktion anzustreben. Damit soll verhindert werden, dass einzelne Komponenten vergessen und später nachgetragen werden müssen. Im Mittelpunkt der funktionalen Machbarkeitsanalyse stehen die Anforderungen an die Daten und das ML-Modell.

Es ist zu prüfen, ob eine Anforderung überhaupt bzw. mit welchem Aufwand verifiziert werden kann. Die Betrachtung der Verifizierbarkeit muss zusammen mit der Spezifikation der Anforderung erfolgen und kann nicht a posteriori durch das Vorhandensein eines Testfalls begründet werden, es sei denn, der Testfall wurde gleichzeitig mit oder vor der Anforderung erstellt (testgetriebene Entwicklung). Testgetriebene Entwicklung wird im ML-Kontext sehr häufig genutzt. Datenmanagement- und ML-Experten müssen die Umsetzung von ML-Funktionen oft explorativ angehen, weil keine Lösung nach dem Stand der Technik zur Verfügung

steht. Dies führt auch dazu, dass ML-Anforderungen häufig aus den Erkenntnissen der Explorationsphase bzw. des explorativen Testens entstehen.

Häufig werden die Tester auch mit der Überprüfung der ML-Anforderungen beauftragt.

Die an der Entwicklung beteiligten Fachabteilungen bewerten die Machbarkeit des Projekts, d.h., es wird geprüft, inwieweit die ML-Anforderungen aus technischer Sicht, aber auch im Hinblick auf die Einhaltung von Kosten und (zugesagten) Terminen, realisiert werden können. Die Machbarkeitsüberlegungen umfassen folgende Fragen:

- Sind die ML-Anforderungen und andere Annahmen realistisch?
- Welche Abhängigkeiten und Wechselwirkungen bestehen?
- Sind der geschätzte Aufwand und die Ressourcen ausreichend?
- Können die Kosten und Termine eingehalten werden?
- Welches sind die geschäftlichen und technischen Risiken?

Die Antworten auf diese Fragen dienen als Input für das Projektmanagement.

Die Analyse von ML-Anforderungen ist entscheidend, um deren Korrektheit und technische Machbarkeit sicherzustellen. Selbst wenn Anforderungen einfach erscheinen, muss eine fundierte Analyse durchgeführt werden. Der Umfang und die Angemessenheit der Analyse hängen vom Produktkontext ab, z.B. von der Plattform. Das Ergebnis der Analyse kann von einer einfachen Attribuierung bis hin zu komplexen Simulationen oder dem Bau eines Demonstrators variieren, um die Machbarkeit der ML-Anforderungen zu bewerten.

Die Anforderungsanalyse unterstützt das Projektmanagement (MAN.3) bei der Projektbewertung. Eine Überarbeitung der ML-Anforderungen durch die Analyse kann den Arbeitsumfang neu definieren. Sobald diese Lösung festgelegt ist, wählt MLE.4 Verifizierungsmaßnahmen aus, die genau diesen Problembereich ansprechen.

Eine Besonderheit bei ML-Anforderungen ist, dass diese oft mit Wahrscheinlichkeiten belegt sind. Dies ist bereits aus der »konventionellen« ADAS-Entwicklung bekannt, gilt aber für ML-Modelle in fast allen Bereichen. So findet man z.B. Anforderungen wie: »Ein Fußgänger muss bei Schneefall noch mit 95 % Sicherheit erkannt werden, selbst wenn er zu 60 % verdeckt ist.« Die Analyse einer solchen Anforderung ist aufwendig, da man einmal die Anforderung selbst hat, die an das Modell gestellt wird. Daraus ergeben sich aber weitere Anforderungen an die Daten, die es erst ermöglichen, ein Modell zu trainieren, das diese Anforderung erfüllen kann. Vereinfacht zusammengefasst ergeben sich aus dieser Anforderung die folgende High-Level-Anforderungen an die Daten:

- Die Daten müssen Schneefall beinhalten.
- Es müssen Daten mit verschiedenen Abdeckungsgraden von Fußgängern vorliegen.

Es ist ersichtlich, dass eine Anforderung, die aus dem Softwarebereich übertragen wird, große Auswirkungen auf die ML-Anforderungen haben kann und entsprechend analysiert werden muss.

MLE.1.BP4: Analyse der Auswirkungen auf die ML-Betriebsumgebung. Analysiere die Auswirkungen, die die ML-Anforderungen auf die Schnittstellen der Softwarekomponenten und die ML-Betriebsumgebung haben.

Anmerkung 7: *Die ML-Betriebsumgebung ist definiert als die Infrastruktur und die Informationen, die sowohl das trainierte ML-Modell als auch das eingesetzte ML-Modell zur Ausführung benötigen.*

Ein wesentlicher Aspekt der ML-Anforderungsanalyse ist es, den Einfluss der Betriebsumgebung sowohl auf das trainierte als auch auf das eingesetzte ML-Modell zu verstehen und zu analysieren. Dies schließt die Auswirkungen des ML-Modells auf Nutzer, das Zielsystem, auf dem das ML-Modell läuft, usw. ein. Diese Auswirkungen müssen dokumentiert werden, um Änderungen zu ermöglichen und die Anforderungen entsprechend anzupassen.

Mögliche Schlüsselfragen sind:

- Hat diese Anforderung Auswirkungen auf andere Komponenten oder Systeme?
- Betrifft diese Anforderung z.B. das Betriebssystem oder die Hardware, sodass Informationen innerhalb einer bestimmten Zykluszeit verfügbar sein müssen?
- Betrifft diese Anforderung auch andere Anforderungen außerhalb der direkten Betriebsumgebung?

Eine Besonderheit bei der ML-Entwicklung, die hier zu beachten ist, besteht darin, dass in den meisten Fällen für das Training und die Nutzung des Modells unterschiedliche Betriebsumgebungen genutzt werden, meist auch unterschiedliche Programmiersprachen. So wird das Training oft auf einem PC in Python durchgeführt, wohingegen die Nutzung des Modells später in der eigentlichen Betriebsumgebung, z.B. in einem ADAS-System, in C++ umgesetzt wird.

Letztendlich muss in den ML-Anforderungen identifiziert und dokumentiert werden, was von der Infrastruktur und dem Datenmanagement benötigt wird, um das (eingesetzte und trainierte) ML-Modell auszuführen.

Die Basispraktiken MLE.1.BP3 und MLE.1.BP4 sind zusammen mit der Rückverfolgbarkeit und der Konsistenz wesentlich, um einen systematischen Ansatz für das Anforderungsmanagement aufzuzeigen.

MLE.1.BP5: Sicherstellung der Konsistenz und Herstellung der bidirektionalen Rückverfolgbarkeit. Stelle die Konsistenz sicher und richte eine bidirektionale Rückverfolgbarkeit zwischen ML-Anforderungen und Softwareanforderungen sowie zwischen ML-Anforderungen und der Softwarearchitektur ein.

Anmerkung 8: *Bidirektionale Rückverfolgbarkeit unterstützt die Konsistenz, erleichtert die Auswirkungsanalyse von Änderungsanträgen und den Nachweis der Verifikationsabdeckung. Rückverfolgbarkeit allein, z.B. das Vorhandensein von Verknüpfungen, bedeutet nicht notwendigerweise, dass die Informationen zueinander konsistent sind.*

Anmerkung 9: *Redundante Rückverfolgbarkeit ist nicht beabsichtigt, aber mindestens einer der angegebenen Rückverfolgbarkeitspfade ist redundant.*

Die Rückverfolgbarkeit von den ML-Anforderungen kann zu den Softwareanforderungen oder zur Softwarearchitektur nachgewiesen werden oder eine Kombination aus beiden Möglichkeiten darstellen. Es sollte der Ansatz verfolgt werden, der die Modellentwicklung am besten unterstützt.

Bei der Modellentwicklung ohne Softwareanforderungen beziehen sich die ML-Anforderungen möglicherweise direkt auf die Systemanforderungen. In diesem Fall ist es wichtig, Konsistenz und bidirektionale Nachverfolgbarkeit zwischen den Systemanforderungen und den ML-Anforderungen zu gewährleisten.

Voraussetzung für die Nachvollziehbarkeit und den Nachweis der Konsistenz ist, dass die Softwareanforderungen oder vorgelagerte Systemanforderungen in strukturierter Form vorliegen. Fehlen diese Informationen, müssen Annahmen getroffen und dokumentiert werden.

Zusammenfassend kann gesagt werden, dass die Rückverfolgbarkeit Folgendes unterstützt:

- Die Konsistenzprüfungen, z.B. die Überprüfung der Vollständigkeit und Korrektheit von ML-Anforderungen
- Die Folgenabschätzung bei Änderungswünschen oder Fehlern
- Die Berichterstattung über den Stand der Umsetzung von ML-Anforderungen

Die Themen Rückverfolgbarkeit und Konsistenz werden auch in Abschnitt 5.7 behandelt.

MLE.1.BP6: Kommunikation der vereinbarten ML-Anforderungen und Auswirkungen auf die Betriebsumgebung. Kommuniziere die vereinbarten ML-Anforderungen und die Ergebnisse der Auswirkungsanalyse auf die ML-Betriebsumgebung an alle betroffenen Parteien.

Der Projektplan sollte die Projektschnittstellen zu den ML-Anforderungen definieren. Diese Schnittstellen sollten mindestens den Software-Projektleiter, den ML-Modell-Tester, die Qualitätssicherung und Vertreter der betroffenen Komponenten der Betriebsumgebung umfassen.

Die Kommunikation sollte nach dem Push-Prinzip und nicht nach dem Pull-Prinzip erfolgen. Das bedeutet, dass die betroffenen Stakeholder aktiv informiert werden müssen.

Es muss ein definiertes Statusmodell der Anforderungen geben, z.B. »in Prüfung«, »freigegeben« usw., das die Kommunikation unterstützt. So müsste sichergestellt sein, dass nur Anforderungen mit einem Endstatus, z.B. »freigegeben«, »abgelehnt«, »doppelt«, für die weitere Verarbeitung verwendet werden.

Es ist zu dokumentieren, wer bei Änderungen des Status der Anforderungen (z.B. neue Anforderungen, analysiert, geändert usw.) oder der betroffenen ML-Modelle zu informieren ist.

Die Kommunikation und Weitergabe von Informationen (Handover) bei der Verfeinerung des Systems, was gewissermaßen der linken Seite des V-Modells entspricht, wird in Abschnitt 5.9.1 ausführlicher behandelt.

12.1.3 Erzeugte Informationsobjekte

17-00 Anforderung und 17-54 Anforderungsattribut	<p>Anforderungen und deren Attribute werden in der Regel in der ML-Spezifikation in einem Anforderungsmanagement-Tool angegeben. Anforderungen müssen mögliche Qualitätskriterien erfüllen. Sie sollten beispielsweise eindeutig und ausreichend atomar sein.</p> <p>Attribute können diverse zusätzliche Informationen beinhalten wie Einstufungen, Priorisierungen, die Zuordnung zu Releases, Reviewstatus etc.</p> <p>Anforderungen müssen strukturiert sein, wozu u.a. die Anforderungsattribute dienen.</p>
15-51 Analyseergebnisse	<p>Die Ergebnisse der Analysen und die Argumentationen, die zu ihnen geführt haben, müssen nachvollziehbar dokumentiert werden. Dies kann werkzeuggestützt erfolgen oder in Review-Checklisten. Es gibt diesbezüglich keine besonderen Vorgaben zur Form.</p>
13-51 Konsistenznachweise	<p>Die Konsistenz der ML-Anforderungen zu Softwareanforderungen oder direkt zu Systemanforderungen oder zu Annahmen in Ermangelung vorhandener übergeordneter Anforderungen muss für jede Anforderung einzeln überprüft werden, was nachvollziehbar sein muss. Ein Reviewprotokoll, das generell die Konsistenz belegt, genügt also nicht. Das notwendige feingranulare Review sollte nach Möglichkeit direkt im Anforderungsmanagement-Tool dokumentiert werden. Ein spezielles Arbeitsprodukt als Konsistenznachweis ist nicht notwendig.</p>



13-52**Kommunikationsnachweise**

Jedes greifbare Artefakt, z.B. E-Mails, Sitzungsprotokolle, Offene-Punkte-Listen, kann als Nachweis für Capability Level 1 herangezogen werden, solange die relevanten Interessengruppen angesprochen werden. Es sollte jedoch immer bedacht werden, dass der entsprechende Nachweis auch zu einem sehr viel späteren Zeitpunkt noch auffindbar sein muss.

Unter Berücksichtigung von PA 2.1 und insbesondere GP 2.1.6 ist eine stärker formalisierte und geplante Vorgehensweise mittels definierter Kommunikationsmedien erforderlich. Dies geschieht oft werkzeuggestützt. Im Hinblick auf Capability Level 3 sollte dies im Standardprozess beschrieben und auf das Projekt zugeschnitten werden.

12.1.4 Zusätzliche Überlegungen

Bezug zu anderen Automotive SPICE®-Prozessen

- Die ML-Datenanforderungen dienen als Hauptinput für das ML-Datenmanagement SUP.11.
- Aus den ML-Anforderungen wird die ML-Architektur MLE.2 abgeleitet.
- Aus den ML-Anforderungen werden die Testfälle für das Testen des ML-Modells MLE.4 erstellt.
- Die Priorisierung der Umsetzung von ML-Anforderungen kann entsprechend der Definition der Release-Umfänge erfolgen (s. SPL.2).
- Die ML-Anforderungsanalyse ist mit dem Projektmanagement MAN.3 verbunden, da sie eine Risikobewertung und Machbarkeitsanalyse durchführt, um das Produkt innerhalb des Umfangs, der Kosten, des Zeitplans und der Qualität zu liefern.
- Die Releaseplanung von ML-Anforderungen sollte im Hinblick auf die Termin- und Ressourcenplanung (Schätzung) von MAN.3 erfolgen.
- Ein weiterer Bezug besteht zum Konfigurationsmanagement SUP.8 wegen der Werkzeugunterstützung für die Entwicklung, Versionierung und das Ziehen von Baselines von ML-Anforderungen.
- Die ML-Anforderungsanalyse steht im Zusammenhang mit dem Problemlösungs-Management SUP.9 und dem Änderungsmanagement SUP.10, da auch Mängel und Änderungswünsche behandelt werden müssen.

Typische Fallstricke

- Vage, mehrdeutige oder unklare Anforderungen können zu Missverständnissen und Fehlinterpretationen führen, die wiederum zu Fehlern oder Inkonsistenzen in der Datensammlung und der ML-Architektur führen können.
- Anforderungen, die nicht ausreichend detailliert oder vollständig sind, können zu unvollständigen oder fehlerhaften Datensammlungen und ML-Modellen führen, die nicht ihre beabsichtigte Funktion erfüllen können.
- Fehlerhafte oder unvollständige Anforderungen an die Datensammlung können zu einer Datenverzerrung (Data Bias) führen (s. SUP.11).

Zu berücksichtigen in Bezug auf PA 2.1

- Unter Berücksichtigung des Kommunikationsnachweises und insbesondere GP 2.1.7 ist ein stärker formalisiertes und geplantes Vorgehen mittels definierter Kommunikationsmedien erforderlich. Dies geschieht häufig werkzeuggestützt.

Zu berücksichtigen in Bezug auf PA 2.2

- Hier geht es vor allem darum, wie das ML-Anforderungsmanagement-Tool eingerichtet ist, wie der Review- und Baseline-Prozess implementiert ist und durch das Tool unterstützt wird und wie die Nachvollziehbarkeit sichergestellt wird.
- Es werden aber auch Qualitätsvorgaben gemacht, wie Anforderungen zu erstellen sind, welcher Syntax sie folgen sollen etc.

Hinweise für Assessoren

- Seien Sie offen für die große Bandbreite von ML-Anforderungen.
- Ein sehr wichtiger Bestandteil der ML-Anforderungen sind die Datenanforderungen. Stellen Sie sicher, dass diese präzise und ausführlich definiert sind und den ML-Datenmanagement-Prozess unterstützen.

12.2 MLE.2 Machine-Learning-Architektur

Der Zweck ist die Entwicklung einer ML-Architektur zur Unterstützung von Training und Einsatz, die mit den ML-Anforderungen übereinstimmt, und die Bewertung der ML-Architektur anhand definierter Kriterien.	Plug-in
<p>Erwartete Prozessergebnisse:</p> <ul style="list-style-type: none">■ Eine ML-Architektur wird entwickelt.■ Hyperparameterbereiche und Anfangswerte werden als Grundlage für das Training festgelegt.■ Eine Evaluierung der ML-Architekturelemente wird durchgeführt.■ Die Schnittstellen der ML-Architekturelemente sind definiert.■ Die Ziele für den Ressourcenverbrauch der ML-Architekturelemente sind definiert.■ Konsistenz und bidirektionale Rückverfolgbarkeit zwischen den ML-Architekturelementen und den ML-Anforderungen sind hergestellt.■ Die ML-Architektur ist vereinbart und allen betroffenen Parteien mitgeteilt worden.	

12.2.1 Prozessbeschreibung

Die ML-Architektur spielt eine entscheidende Rolle bei der Erstellung qualitativ hochwertiger ML-Modelle, indem sie einen strukturierten und systematischen Rahmen für den gesamten ML-Entwicklungsprozess bereitstellt. Durch die Auswahl geeigneter Modelle, die Festlegung initialer Hyperparameter und die Definition von Vor- und Nachverarbeitungs-Komponenten, aber auch durch Vorgaben der Trainings- und der eigentlichen Betriebsumgebung schafft sie eine klare Richtlinie für die Entwicklung. Dies gewährleistet nicht nur eine effektive Steuerung des Trainingsprozesses, sondern ermöglicht auch die Anpassung an verschiedene Szenarien und Anforderungen. Die Schnittstellen zwischen den verschiedenen Architekturelementen müssen berücksichtigt und dokumentiert werden, um eine nahtlose Integration und Interaktion zwischen den Modellkomponenten sicherzustellen. Darüber hinaus hilft die ML-Architektur bei der genauen Definition der Ressourcenanforderungen, einschließlich der CPU/GPU-Auslastung, des Speicherbedarfs und anderer kritischer Ressourcen. Diese genaue Spezifikation ist entscheidend, um die Machbarkeit des Modells sicherzustellen und unerwartete Herausforderungen während der Implementierung zu minimieren. Insgesamt bietet die ML-Architektur eine strukturierte Grundlage, die nicht nur die Effizienz und Qualität der ML-Modelle steigert, sondern auch die Anpassungsfähigkeit und Skalierbarkeit in unterschiedlichen Umgebungen und Einsatzszenarien unterstützt. Der Prozess der ML-Architektur geht über die einfache Auswahl eines geeigneten Modells und die Definition seiner Hyperparameter hinaus. Die Schritte umfassen u.a. die Festlegung von Vor- und Nachverarbeitungs-Komponenten, die Definition von Softwarekomponenten, die für das Training benötigt werden, sowie die Festlegung von Zielen für

19 Modellerweiterungen

19.1 Automotive SPICE® for Cybersecurity

Vor dem Hintergrund der stetigen Weiterentwicklung der Automobilindustrie und der zunehmenden Bedeutung vernetzter Fahrzeuge gewinnt das Thema Cybersecurity immer mehr an Bedeutung. Die UNECE-Verordnung R155 [UN-Regelung 155] verpflichtet Fahrzeughersteller, Cybersecurity-Risiken in der Lieferkette zu identifizieren und zu managen.

Dabei geht es um umfassende Cybersecurity-Maßnahmen auf organisatorischer Ebene, die zusätzlich zu den technischen Maßnahmen erforderlich sind. Die organisatorischen Maßnahmen sollen die kontinuierliche Implementierung und Wartung dieser technischen Mittel systematisch und strukturiert sicherstellen und steuern. Cybersecurity auf Organisationsebene ist ein komplexes und multidimensionales Gebiet, das eine sorgfältige Betrachtung und Implementierung in allen Phasen der System- und Softwareentwicklung erfordert.

Um Cybersecurity-relevante Prozesse in das etablierte Rahmenwerk von Automotive SPICE® zu integrieren, wurden zusätzliche Prozesse im Process Reference and Assessment Model for Cybersecurity Engineering (Cybersecurity PAM) definiert.

Der internationale Standard ISO/SAE 21434, auf den im folgenden Kapitel eingegangen wird, deckt Cybersecurity-Aktivitäten über den gesamten Produktlebens- und -Nutzungszyklus ab, von der Konzeption einer Baureihe oder Plattform über die Entwicklung und Nutzung bis hin zur Verschrottung der einzelnen Fahrzeuge. Automotive SPICE® hingegen betrachtet lediglich die Produktentwicklung. Eine Reihe von Cybersecurity-Maßnahmen auf Organisationsebene werden nicht im Kontext eines Entwicklungsprojekts durchgeführt; diese werden durch das Automotive Cybersecurity Management System (ACSMS) abgedeckt und sind daher nicht Teil der Erweiterung.

Bei der Erweiterung Automotive SPICE® for Cybersecurity werden die Aktivitäten zur Cybersecurity im Projekt betrachtet. Es ist zu erwarten, dass diese Erweiterung in Zukunft häufiger in den von den OEMs erwarteten Umfängen berücksichtigt wird. Dies unterstreicht die Notwendigkeit für Organisationen in der Automobilindustrie, ihre Prozesse und Systeme kontinuierlich zu bewerten und anzupassen, um nicht nur die funktionale Sicherheit, sondern auch die Cybersecurity ihrer Produkte zu gewährleisten.

Um Cybersecurity in einer Organisation zu etablieren, empfiehlt es sich jedoch, zunächst die ISO/SAE 21434 umzusetzen, die klarere Vorgaben für die unmittelbare Umsetzung macht, Automotive SPICE® for Cybersecurity dient dabei als Messinstrument für die Prozessfähigkeit.

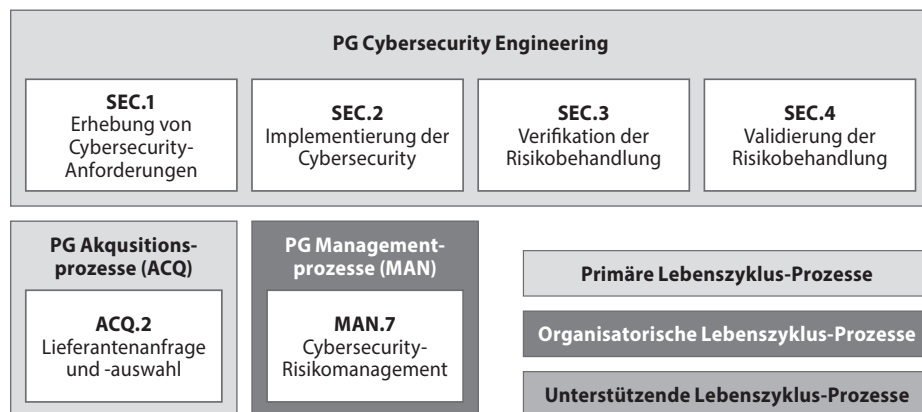


Abb. 19-1 Prozesse der Cybersecurity-Erweiterung

Die Erweiterung fügt eine Prozessgruppe mit vier Prozessen sowie zwei zusätzliche Prozesse zu bereits bestehenden Prozessgruppen hinzu, um die spezifischen Anforderungen der Cybersecurity zu adressieren. In diesem Abschnitt werden diese Prozesse und die zugehörige Prozessgruppe im Detail vorgestellt, um ein besseres Verständnis ihrer Rolle und Bedeutung im Kontext von Cybersecurity zu ermöglichen. Abbildung 19-1 zeigt die Prozesse der Cybersecurity-Erweiterung.

Die hinzugefügte Prozessgruppe ist die Security Engineering Process Group (SEC), die die folgenden Prozesse umfasst:

■ SEC.1 Erhebung von Cybersecurity-Anforderungen

Der Zweck dieses Prozesses besteht darin, Cybersecurity-Ziele und -Anforderungen aus den Ergebnissen des Risikomanagements abzuleiten und Konsistenz zwischen der Risikobewertung, den Cybersecurity-Zielen und den -Anforderungen sicherzustellen. Bei erfolgreicher Implementierung dieses Prozesses werden die Cybersecurity-Ziele definiert und die Cybersecurity-Anforderungen aus diesen Zielen abgeleitet. Zudem wird eine bidirektionale Nachverfolgbarkeit zwischen den Cybersecurity-Anforderungen und -Zielen sowie zwischen den Cybersecurity-Zielen und den Bedrohungsszenarien hergestellt. Schließlich werden die Cybersecurity-Anforderungen abgestimmt und an alle betroffenen Parteien kommuniziert.

■ SEC.2 Implementierung der Cybersecurity

Dieser Prozess zielt darauf ab, die Cybersecurity-Anforderungen den Elementen des Systems und der Software zuzuweisen und sicherzustellen, dass diese implementiert werden. Der Architekturentwurf wird verfeinert, die Cybersecurity-Anforderungen werden den Elementen zugeordnet, und geeignete Sicherheitskontrollen werden ausgewählt. Zudem wird eine Analyse der Schwachstellen durchgeführt, der Feinentwurf wird weiter detailliert, Software-Units werden entwickelt und eine Konsistenz sowie bidirektionale Nachverfolgbarkeit zwischen Architekturentwurf und Feinentwurf wird hergestellt. Abschließend wird die Implementierung der Cybersecurity-Risikobehandlung abgestimmt und an alle betroffenen Parteien kommuniziert.

■ SEC.3 Verifikation der Risikobehandlung

Der Hauptzweck dieses Prozesses besteht darin, zu bestätigen, dass die Implementierung des Architekturentwurfs und die Integration der Komponenten den Cybersecurity-Anforderungen, dem Feinentwurf und dessen Detaillierung entsprechen. Eine Strategie zur Verifizierung und Integration der Risikobehandlung wird entwickelt, implementiert und aufrechterhalten. Zudem wird eine Spezifikation für die Verifizierung der Risikobehandlung entwickelt, die geeignet ist, den Nachweis der Einhaltung der Implementierung der Cybersecurity-Anforderungen sowie des Architektur- und Feinentwurfs zu erbringen. Die identifizierten Arbeitsprodukte werden gemäß der Strategie für die Verifizierung der Risikobehandlung überprüft, und die Ergebnisse werden zusammengefasst und an alle betroffenen Parteien kommuniziert.

■ SEC.4 Validierung der Risikobehandlung

Der Zweck dieses Prozesses ist es, zu bestätigen, dass das integrierte System die zugehörigen Cybersecurity-Ziele erreicht. Eine Strategie zur Validierung der Risikobehandlung wird entwickelt, implementiert und mit den relevanten Stakeholdern abgestimmt. Das implementierte Design und die integrierten Komponenten werden gemäß der definierten Strategie zur Validierung der Risikobehandlung validiert, die Aktivitäten werden dokumentiert und die Ergebnisse aufgezeichnet. Die Ergebnisse der Validierung werden zusammengefasst und an alle betroffenen Parteien kommuniziert.

Zur Prozessgruppe der Akquisitionsprozesse (ACQ) wird der folgende Prozess hinzugefügt:

■ ACQ.2 Lieferantenanfrage und -auswahl

Der Zweck dieses Prozesses besteht darin, einen Lieferanten für einen Vertrag oder eine Vereinbarung basierend auf relevanten Kriterien auszuwählen und zu beauftragen. Bewertungskriterien für Lieferanten werden festgelegt und Lieferanten werden anhand der definierten Kriterien bewertet. Es wird eine Anfrage zur Angebotsabgabe an Lieferantenkandidaten ausgestellt und Verträge, Maßnahmen- und Risikominderungs-Pläne werden mit dem Lieferanten abgestimmt und vertraglich vereinbart.

Zur Prozessgruppe der Managementprozesse (MAN) wird der folgende Prozess hinzugefügt:

■ **MAN.7 Cybersecurity-Risikomanagement**

Der Zweck dieses Prozesses ist es, Risiken für Schäden an relevanten Stakeholdern zu identifizieren, zu priorisieren und zu analysieren sowie die entsprechenden Risikobehandlungs-Optionen kontinuierlich zu überwachen und zu steuern. Der Umfang des durchzuführenden Risikomanagements wird festgelegt und geeignete Risikomanagement-Praktiken werden definiert und implementiert. Potenzielle Risiken werden identifiziert und priorisiert, analysiert und bewertet. Risikobehandlungs-Optionen werden bestimmt, Risiken werden kontinuierlich überwacht und für relevante Änderungen identifiziert und Korrekturmaßnahmen werden bei relevanten Änderungen durchgeführt.

19.2 Agile SPICE

In den letzten Jahren haben immer mehr Entwicklungsprojekte agile Prinzipien für ihren Arbeitsansatz übernommen. Das Automotive SPICE®-Prozessreferenz- und Prozessassessment-Modell geht jedoch nicht explizit auf diese innovativen Ansätze ein, was zu unterschiedlichen Interpretationen sowohl bei Assessoren als auch bei Verbesserungsteams führt.

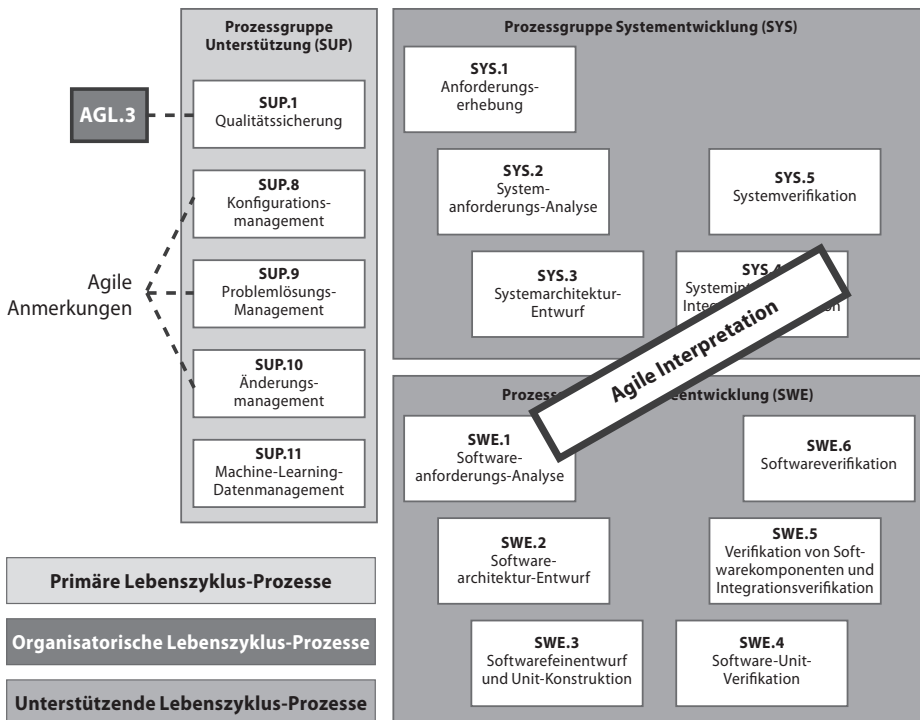


Abb. 19-2 Agile SPICE im Kontext des Automotive SPICE®-Prozessmodells

Inhaltsübersicht

1	Einleitung	1
2	Einstieg in das Thema	5
3	Aufbau von Automotive SPICE®	15
4	Neuerungen in Automotive SPICE® 4.0	25
5	Wesentliche Konzepte	35
6	Prozessübergreifende Themen	53
7	Prozessgruppe der Akquisitionsprozesse	77
8	Prozessgruppe der Bereitstellungsprozesse	91
9	Prozessgruppe zur Systementwicklung	103
10	Prozessgruppe zur Softwareentwicklung	169
11	Prozessgruppe zur Validierung	239
12	Prozessgruppe Machine Learning Engineering	251
13	Prozessgruppe zur Hardwareentwicklung	295
14	Prozessgruppe der Unterstützungsprozesse	343
15	Prozessgruppe der Managementprozesse	403
16	Prozessgruppe der Verbesserungsprozesse	443
17	Prozessgruppe zur Wiederverwendung	457
18	Capability-Dimension von Prozessen	469
19	Modellerweiterungen	519
20	Angrenzende Standards	531
21	Assessments	543

Anhang	557
A Abschließende Worte	559
B Glossar	561
C Abkürzungsverzeichnis	569
D Literaturverzeichnis	573
Index	577

Inhaltsverzeichnis

1	Einleitung	1
1.1	Wir stehen auf den Schultern von Riesen	1
1.2	Zielsetzung und Verwendung dieses Buches	2
1.3	Kapitelüberblick	3
2	Einstieg in das Thema	5
2.1	Einordnung	5
2.2	Zertifizierung für Assessoren und Experten	8
2.3	Prozessverbesserung mit Automotive SPICE®	11
2.4	Automotive SPICE® und Agilität	13
3	Aufbau von Automotive SPICE®	15
3.1	Prozessdimension und Struktur der Prozessbeschreibungen . . .	15
3.1.1	Modellerweiterungen (Extensions)	20
3.2	Capability-Dimension	21
3.2.1	Das Bewertungsschema (NPLF)	22
3.3	Automotive SPICE® Guidelines	24
4	Neuerungen in Automotive SPICE® 4.0	25
4.1	Klarheit in der Terminologie	26
4.2	Erzeugte Informationsobjekte	27
4.3	Änderungen bei den Basispraktiken	28
4.4	Automotive SPICE® 4.0 und Strategien	29
4.5	Unabhängigkeit der Prozesse	30
4.6	Hardwareprozesse	30
4.7	Machine Learning	31
4.8	Validierung	34

5	Wesentliche Konzepte	35
5.1	Der Begriff des Systems	35
5.1.1	Systemebenen	36
5.2	Systematischer Ansatz und Strategie	38
5.3	Angemessenheit der Umsetzung	39
5.4	Trennung von Fragestellung und Lösung	39
5.5	Analysen	41
5.6	Reviews	43
5.7	Bidirektionale Rückverfolgbarkeit und Konsistenz	45
5.8	Übergänge zwischen verschiedenen Verständnisebenen	47
5.9	Kommunikation und Transparenz	47
5.9.1	Kommunikation der vereinbarten Ergebnisse	50
5.9.2	Zusammenfassen und Kommunizieren	50
5.10	Nachvollziehbarkeit	51
6	Prozessübergreifende Themen	53
6.1	Integration, Verifikation und Validierung (IVV)	53
6.1.1	Integration von Teilsystemen	56
6.1.2	Integrationsstufen und funktionales Wachstum	58
6.1.3	Methoden zur Verifikation und Validierung	59
6.1.4	Planung der Verifikation und Validierung	65
6.2	Modellbasiertes Systems Engineering (MBSE)	67
6.3	Applikationsparameter	70
6.4	Besonderheiten bei der Umsetzung der Automotive SPICE®-Konformität beim OEM	73
6.5	Einstufung der Prozesse für das Assessment	75
7	Prozessgruppe der Akquisitionsprozesse	77
7.1	ACQ.4 Lieferantenüberwachung	77
7.1.1	Prozessbeschreibung	77
7.1.2	Basispraktiken	79
7.1.3	Erzeugte Informationsobjekte	86
7.1.4	Zusätzliche Überlegungen	88
8	Prozessgruppe der Bereitstellungsprozesse	91
8.1	SPL.2 Produktfreigabe	91
8.1.1	Prozessbeschreibung	91
8.1.2	Basispraktiken	93
8.1.3	Erzeugte Informationsobjekte	98
8.1.4	Zusätzliche Überlegungen	99

9	Prozessgruppe zur Systementwicklung	103
9.1	SYS.1 Anforderungserhebung	103
9.1.1	Prozessbeschreibung	103
9.1.2	Basispraktiken	107
9.1.3	Erzeugte Informationsobjekte	113
9.1.4	Zusätzliche Überlegungen	114
9.2	SYS.2 Systemanforderungs-Analyse	116
9.2.1	Prozessbeschreibung	116
9.2.2	Basispraktiken	118
9.2.3	Erzeugte Informationsobjekte	127
9.2.4	Zusätzliche Überlegungen	128
9.3	SYS.3 Systemarchitektur-Entwurf	130
9.3.1	Prozessbeschreibung	130
9.3.2	Basispraktiken	132
9.3.3	Erzeugte Informationsobjekte	141
9.3.4	Zusätzliche Überlegungen	142
9.4	SYS.4 Systemintegration und Integrationsverifikation	145
9.4.1	Prozessbeschreibung	145
9.4.2	Basispraktiken	147
9.4.3	Erzeugte Informationsobjekte	154
9.4.4	Zusätzliche Überlegungen	155
9.5	SYS.5 Systemverifikation	157
9.5.1	Prozessbeschreibung	157
9.5.2	Basispraktiken	159
9.5.3	Erzeugte Informationsobjekte	165
9.5.4	Zusätzliche Überlegungen	166
10	Prozessgruppe zur Softwareentwicklung	169
10.1	SWE.1 Softwareanforderungs-Analyse	169
10.1.1	Prozessbeschreibung	169
10.1.2	Basispraktiken	170
10.1.3	Erzeugte Informationsobjekte	177
10.1.4	Zusätzliche Überlegungen	178
10.2	SWE.2 Softwarearchitektur-Entwurf	180
10.2.1	Prozessbeschreibung	180
10.2.2	Basispraktiken	182
10.2.3	Erzeugte Informationsobjekte	188
10.2.4	Zusätzliche Überlegungen	189

10.3	SWE.3 Softwarefeinentwurf und Unit-Konstruktion	191
10.3.1	Prozessbeschreibung	192
10.3.2	Basispraktiken	194
10.3.3	Erzeugte Informationsobjekte	200
10.3.4	Zusätzliche Überlegungen	201
10.4	SWE.4 Software-Unit-Verifikation	204
10.4.1	Prozessbeschreibung	204
10.4.2	Basispraktiken	206
10.4.3	Erzeugte Informationsobjekte	211
10.4.4	Zusätzliche Überlegungen	212
10.5	SWE.5 Verifikation von Softwarekomponenten und Integrationsverifikation	214
10.5.1	Prozessbeschreibung	214
10.5.2	Basispraktiken	218
10.5.3	Erzeugte Informationsobjekte	226
10.5.4	Zusätzliche Überlegungen	227
10.6	SWE.6 Softwareverifikation	229
10.6.1	Prozessbeschreibung	229
10.6.2	Basispraktiken	231
10.6.3	Erzeugte Informationsobjekte	235
10.6.4	Zusätzliche Überlegungen	236
11	Prozessgruppe zur Validierung	239
11.1	VAL.1 Validierung	239
11.1.1	Prozessbeschreibung	239
11.1.2	Basispraktiken	241
11.1.3	Erzeugte Informationsobjekte	247
11.1.4	Zusätzliche Überlegungen	248
12	Prozessgruppe Machine Learning Engineering	251
12.1	MLE.1 Anforderungsanalyse für Machine Learning	251
12.1.1	Prozessbeschreibung	251
12.1.2	Basispraktiken	252
12.1.3	Erzeugte Informationsobjekte	259
12.1.4	Zusätzliche Überlegungen	260
12.2	MLE.2 Machine-Learning-Architektur	262
12.2.1	Prozessbeschreibung	262
12.2.2	Basispraktiken	263
12.2.3	Erzeugte Informationsobjekte	269
12.2.4	Zusätzliche Überlegungen	270

12.3	MLE.3 Machine-Learning-Training	272
12.3.1	Prozessbeschreibung	273
12.3.2	Basispraktiken	274
12.3.3	Erzeugte Informationsobjekte	279
12.3.4	Zusätzliche Überlegungen	280
12.4	MLE.4 Testen des Machine-Learning-Modells	282
12.4.1	Prozessbeschreibung	282
12.4.2	Basispraktiken	284
12.4.3	Erzeugte Informationsobjekte	290
12.4.4	Zusätzliche Überlegungen	291
13	Prozessgruppe zur Hardwareentwicklung	295
13.1	HWE.1 Hardware-Anforderungsanalyse	295
13.1.1	Prozessbeschreibung	295
13.1.2	Basispraktiken	297
13.1.3	Erzeugte Informationsobjekte	303
13.1.4	Zusätzliche Überlegungen	304
13.2	HWE.2 Hardwareentwurf	307
13.2.1	Prozessbeschreibung	308
13.2.2	Basispraktiken	309
13.2.3	Erzeugte Informationsobjekte	317
13.2.4	Zusätzliche Überlegungen	319
13.3	HWE.3 Verifikation des Hardwareentwurfs	322
13.3.1	Prozessbeschreibung	322
13.3.2	Basispraktiken	324
13.3.3	Erzeugte Informationsobjekte	330
13.3.4	Zusätzliche Überlegungen	331
13.4	HWE.4 Verifikation der Hardwareanforderungen	334
13.4.1	Prozessbeschreibung	334
13.4.2	Basispraktiken	336
13.4.3	Erzeugte Informationsobjekte	340
13.4.4	Zusätzliche Überlegungen	341
14	Prozessgruppe der Unterstützungsprozesse	343
14.1	SUP.1 Qualitätssicherung	343
14.1.1	Prozessbeschreibung	343
14.1.2	Basispraktiken	345
14.1.3	Erzeugte Informationsobjekte	351
14.1.4	Zusätzliche Überlegungen	352

14.2	SUP.8 Konfigurationsmanagement	354
14.2.1	Prozessbeschreibung	354
14.2.2	Basispraktiken	356
14.2.3	Erzeugte Informationsobjekte	367
14.2.4	Zusätzliche Überlegungen	368
14.3	SUP.9 Problemlösungs-Management	370
14.3.1	Prozessbeschreibung	370
14.3.2	Basispraktiken	372
14.3.3	Erzeugte Informationsobjekte	377
14.3.4	Zusätzliche Überlegungen	378
14.4	SUP.10 Änderungsmanagement	381
14.4.1	Prozessbeschreibung	381
14.4.2	Basispraktiken	383
14.4.3	Erzeugte Informationsobjekte	388
14.4.4	Zusätzliche Überlegungen	389
14.5	SUP.11 Machine-Learning-Datenmanagement	391
14.5.1	Prozessbeschreibung	391
14.5.2	Basispraktiken	392
14.5.3	Erzeugte Informationsobjekte	399
14.5.4	Zusätzliche Überlegungen	400
15	Prozessgruppe der Managementprozesse	403
15.1	MAN.3 Projektmanagement	403
15.1.1	Prozessbeschreibung	403
15.1.2	Basispraktiken	405
15.1.3	Erzeugte Informationsobjekte	414
15.1.4	Zusätzliche Überlegungen	418
15.2	MAN.5 Risikomanagement	420
15.2.1	Prozessbeschreibung	420
15.2.2	Basispraktiken	423
15.2.3	Erzeugte Informationsobjekte	429
15.2.4	Zusätzliche Überlegungen	430
15.3	MAN.6 Messung	432
15.3.1	Prozessbeschreibung	433
15.3.2	Basispraktiken	434
15.3.3	Erzeugte Informationsobjekte	439
15.3.4	Zusätzliche Überlegungen	440

16	Prozessgruppe der Verbesserungsprozesse	443
16.1	PIM.3 Prozessverbesserung	443
16.1.1	Prozessbeschreibung	443
16.1.2	Basispraktiken	446
16.1.3	Erzeugte Informationsobjekte	453
16.1.4	Zusätzliche Überlegungen	454
17	Prozessgruppe zur Wiederverwendung	457
17.1	REU.2 Management der Produktwiederverwendung	457
17.1.1	Prozessbeschreibung	457
17.1.2	Basispraktiken	459
17.1.3	Erzeugte Informationsobjekte	466
17.1.4	Zusätzliche Überlegungen	467
18	Capability-Dimension von Prozessen	469
18.1	Prozessattribut PA 1.1 Prozessdurchführung (CL 1)	470
18.2	Prozessattribut PA 2.1 Prozessdurchführungs- Management (CL 2)	472
18.3	Prozessattribut PA 2.2 Arbeitsprodukt-Management (CL 2)	482
18.4	Prozessattribut PA 3.1 Prozessdefinition (CL 3)	489
18.5	Prozessattribut PA 3.2 Prozesseinsatz (CL 3)	495
18.6	Prozessattribut PA 4.1 Quantitative Analyse (CL 4)	500
18.7	Prozessattribut PA 4.2 Quantitative Steuerung (CL 4)	507
18.8	Prozessattribut PA 5.1 Prozessinnovation (CL 5)	510
18.9	Prozessattribut PA 5.2 Umsetzung der Prozessinnovation (CL 5)	514
19	Modellerweiterungen	519
19.1	Automotive SPICE® for Cybersecurity	519
19.2	Agile SPICE	522
19.3	SPICE for Mechanical Engineering	524
19.4	Data Management SPICE	529
20	Angrenzende Standards	531
20.1	ISO 26262 – Funktionale Sicherheit	531
20.1.1	Ein Einblick in ISO 26262	532
20.1.2	Zusammenspiel mit Automotive SPICE®	533
20.1.3	Automotive SPICE®-Assessment vs. Functional Safety Assessment	534
20.1.4	Automotive SPICE®-Assessment vs. Functional Safety Audit	535
20.1.5	Synergie von Automotive SPICE® und ISO 26262	536

20.2	ISO/IEC 15288	536
20.2.1	Systems Engineering in Automotive SPICE®	537
20.2.2	Zusammenspiel mit Automotive SPICE®	538
20.3	ISO/SAE 21434:2021	539
20.3.1	Zusammenspiel mit Automotive SPICE®	540
20.4	SOTIF	540
20.4.1	Zusammenspiel mit Automotive SPICE®	541
20.5	AUTOSAR	541
20.5.1	Zusammenspiel mit Automotive SPICE®	542
21	Assessments	543
21.1	Arten von Assessments	543
21.2	Assessment-Scope	544
21.3	Bewertung und Durchführung	545
21.4	Vertraulichkeit der Informationen	546
21.5	Dauer eines Assessments	546
21.6	Rollen im Assessment	547
21.7	Vorbereitung und Durchführung von Assessments	549
21.8	Assessmentbericht	551
21.9	Best Practices zur Vorbereitung auf Assessments	553
21.9.1	Wie man sich auf ein Assessment vorbereitet	553
21.9.2	Was man bei der Assessment-Vorbereitung nicht tun sollte	554
21.10	Potenzialanalysen	554
	Anhang	557
A	Abschließende Worte	559
B	Glossar	561
C	Abkürzungsverzeichnis	569
D	Literaturverzeichnis	573
	Index	577

Handbuch Automotive SPICE® 4.0

- Umfassende Darstellung
- Inkl. Modellerweiterungen wie Plug-ins und Add-ons
- Mit praxisnahen Beispielen und konkreten Hinweisen

Dieses Buch bietet einen strukturierten Einstieg in die praktische Anwendung der vierten Generation des Prozessreferenz- und -assessment-modells Automotive SPICE®. Es gibt einen tiefen Einblick sowohl in die Zusammenhänge von Automotive SPICE® im Kontext der Entwicklung von Automobilelektronik als auch in die Verknüpfung mit mechatronischen Disziplinen und neuen Vorgehensweisen. Ein besonderer Fokus liegt auf den neu hinzugekommenen Prozessen des Machine Learning.

Im Einzelnen werden behandelt:

- Automotive SPICE® im Lebenszyklus, Struktur und Elemente, Plug-in-Konzept, Guidelines des VDA
- Prozessdimension und Prozesse im Detail
- Capability-Dimension und Capability Levels 1 bis 5
- Modellerweiterung innerhalb und ergänzend zu Automotive SPICE® (Hardware- und Mechanikentwicklung, Machine Learning Engineering, Agilität)
- Assessments (Vorbereitung, Durchführung, Rollen, Bericht)
- Synergieeffekte (z. B. zu funktionaler Sicherheit, Cybersecurity)

Das Buch richtet sich in erster Linie an Praktiker, die einen verständlichen Einstieg in Automotive SPICE® suchen, und dient als Hilfestellung bei der Umsetzung von Automotive SPICE®-konformen Prozessen sowie als Wissensbasis für Assessoren.

Alexander Levin ist Lead Consultant bei UL Solutions, vormaliger Leiter der Expert Area »SPICE« und aktiv in der iNTACS-Arbeitsgruppe »Functional Safety«.

Christina Stathatou ist Lead Consultant bei UL Solutions und hat das Plug-in »Machine Learning for Automotive SPICE®« mitentwickelt.

Volker Lehmann ist Principal Consultant, und People Lead bei UL Solutions und iNTACS® Certified Principal Assessor and Instructor.

Josefin A. Benning ist Lead Consultant und People Lead bei UL Solutions und leitet die iNTACS-Arbeitsgruppe »New Training Courses and Coordination«.