Eisenbahn-Bundesamt

EBA Forschungsbericht
Nummer 2019-01

Betrachtung zur Softwareentwicklung im Eisenbahnbereich

EBA Forschungsbericht 2019-01
Projektnummer 2017-I-1-1217

Betrachtung zur Software-Entwicklung im Eisenbahnbereich

von

Fraunhofer-Institut für offene Kommunikationssysteme (FOKUS), Berlin
System Quality Center (SQC)

Prof. Dr. Holger Schlingloff
Dr.-Ing. Jens Gerlach
Dipl.-Math. Marko Fabiunke

Im Auftrag des Eisenbahn-Bundesamtes

Impressum

HERAUSGEBER
Eisenbahn-Bundesamt

Heinemannstraße 6
53175 Bonn

www.eba.bund.de

DURCHFÜHRUNG DER STUDIE
Fraunhofer-Gesellschaft e. V., München
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

ABSCHLUSS DER STUDIE
Dezember 2018

REDAKTION
Andreas Czarski, Referat 22
Ariane Boehmer, Referat 52

PUBLIKATION ALS PDF
https://www.dzsf.bund.de/Forschungsergebnisse/Forschungsberichte

ISSN 2627-9851

doi: 10.48755/dzsf.210022.01

Bonn, Februar 2019

Inhaltsverzeichnis

Kurzbeschreibung

Software wird in immer größerem Umfang in technische und organisatorische Systeme eingebettet und übernimmt darin immer öfter auch sicherheitskritische Aufgaben. Gleichzeitig ist der Innovationsdruck der (europäischen) Bahnindustrie u.a. wegen des großen und (noch) zunehmenden internationalen Wettbewerbs groß und erfordert die Betrachtung von Optimierungspotential auch im Bereich der Software-Entwicklung.

Die vorliegende Studie greift diese Fragestellung auf und beschreibt die aktuellen Tendenzen und Herausforderungen, denen sich die Software-Entwicklung im Eisenbahnbereich in den kommenden Jahren stellen muss. Dabei werden künftige Produktstrukturen, Standards, Methoden und Prozesse, sowie Belange der Ausbildung thematisiert. Ausgehend von einer Analyse der Trends in den benachbarten Domänen Automobilindustrie, Avionik, Telekommunikation und Industrieautomatisierung werden Vorschläge für den Eisenbahnsektor entwickelt. Diese werden konkretisiert an zwei Beispielen: Der Integration von RBC und STW, sowie der Integration OBU und TCMS. Die Ergebnisse lassen sich wie folgt zusammenfassen (siehe auch Abschnitt 1.3):

- Die Digitalisierung des schienengebundenen Verkehrs wird weiter zunehmen. Die Bahntechnik kann hier von vielen Entwicklungen anderer Bereiche profitieren, die sorgsam beobachtet werden sollten.
- Die Aufspaltung der Sicherheitsnormen für Softwareentwicklung gemäß den verschiedenen Anwendungsdomänen ist inhaltlich nicht immer zu rechtfertigen. Es sollte geprüft werden, ob für bahntechnische Anwendungen vermehrt auch Zertifizierungen aus anderen Fachdomänen in Betracht kommen.
- Modellbasierte Entwicklung erfordert eine bewusste Entscheidung für die geeignete Modellierungssprache.
- Auch sicherheitsgerichtete Systeme lassen sich, unter Voraussetzung entsprechender Granularität, mit agilen Methoden entwickeln.
- Funktionale Sicherheit und IT-Sicherheit müssen gemeinsam behandelt werden; die Technologien für IT-Sicherheit bei Datenübertragung und Datenablage sind im Prinzip vorhanden, sind in der Bahnbranche aber noch nicht durchgängig etabliert.
- In der Bahntechnik entsteht ein immer größerer Bedarf an IT-Fachleuten und System-Ingenieuren, insbesondere mit Kenntnissen und Fertigkeiten im Testen.

1 Einleitung

1.1 Aufgabenstellung

Software wird in immer größerem Umfang in technische und organisatorische Systeme eingebettet und übernimmt darin immer öfter auch sicherheitskritische Aufgaben. Dieser allgemeine Trend der Digitalisierung – nicht nur in der Bahnindustrie, sondern zunehmend aller Lebensbereiche – findet seit vielen Jahren statt (man denke z.B. an den Dotcom-Hype im Jahr 2000) und wird über die Jahre mit unterschiedlichen Schwerpunkten verknüpft – so beschreiben aktuell die Schlagworte Industrie 4.0 und Internet der Dinge (IoT) die Thematik.

Der Innovationsdruck der (europäischen) Bahnindustrie ist u.a. wegen des zunehmenden internationalen Wettbewerbs groß und (noch) zunehmend. Dass mit Hochdruck an der Beseitigung von Schwachstellen gearbeitet bzw. nach Kosten- oder Optimierungspotential gesucht wird, ist eine geradezu notwendige Konsequenz. Die Frage, ob in anderen Domänen technische Lösungen, Mechanismen oder Strukturen herausgebildet wurden, die nutzbringend in die Domäne „Bahn“ übertragen werden können, ist naheliegend und mehr als berechtigt.

Die vorliegende Studie greift diese Fragestellung auf und untersucht und bewertet den Stand der Softwareentwicklung im Eisenbahnsektor im Vergleich zu anderen Bereichen eingebetteter Systeme, um geeignete Transferpotentiale aufzuzeigen. Basierend auf einer Analyse aktueller Trends und Entwicklungen in ausgewählten Fremddomänen (Automobilindustrie, Avionik, Telekommunikation und Industrieautomatisierung) wurde untersucht, inwieweit sich aus den dort gemachten Erfahrungen für den Bahnbereich konkrete Vorschläge ableiten lassen, für:

- die Anpassung von Produktstrukturen (Hardware und Software),
- die Anpassung bestehender Standardisierungen,
- die Anpassung der Entwicklungs- und Prüfmethoden/-prozesse und
- die Anpassung der Ausbildung von Softwareentwicklern und Ingenieuren.

Ausgangspunkt der Betrachtung ist zunächst die Frage, inwieweit sich durch Veränderungen der Produktstrukturen eine Verbesserung der Produkte bzw. deren Qualität erreichen lässt. Die wird ergänzt durch die Frage, wie der Produktentwicklungsprozess beschaffen sein muss, um eine adäquate Gesamtqualität der Produkte sicherzustellen. Den Standards kommt dabei eine zentrale Rolle zu, weil damit u.a. auch die Arbeitsweisen (d.h. Methoden und Prozesse) bestimmt werden, die ihrerseits in der Regel auch deutliche Implikationen bezüglich des Aufwands (d.h. Zeit und Kosten) und Qualität nach sich ziehen. Die Regelwerke der Bahn hier sind frühen Ursprungs und zwar so, dass Software noch gar keine und Telekommunikation nur eine sehr geringe Rolle gespielt haben. In der heutigen Situation, in der wesentliche (d.h. leistungsbestimmende) Produkteigenschaften durch Software definiert werden und technische Kommunikation der allgegenwärtige Begleiter ist, erscheinen die Regelwerke zur Software-Erstellung quasi als (z. Zt. allerdings interpretierungsbedürftiger) Leitstrahl. Eine regelmäßige Bestandsaufnahme der Trends in anderen Branchen und der Abgleich mit Erfordernissen der Bahntechnik ist daher geboten.

1.2 Herangehensweise

Ausgehend von den in der Aufgabenstellung genannten vier Problemfeldern wurde ein Fragebogen erarbeitet, der die genannten Problemfelder auf konkrete Sachfragen herunterbricht (vgl. Anhang). Ba-

sierend auf diesem Fragebogen erfolgte zunächst eine Recherche der identifizierbaren relevanten Trends und Entwicklungen innerhalb der zu untersuchenden Fremddomänen (Automobilindustrie, Avionik, Telekommunikation und Industrieautomatisierung). Die Ergebnisse dieser Recherche werden in Kapitel 2 (Entwicklung in anderen Fachdomänen) vorgestellt, wobei hier vorrangig eine reine Darstellung der Trends und Entwicklung in den Fremddomänen erfolgt. Basierend auf den Ergebnissen dieser Recherche wurden die gefunden Trends und Entwicklungen hinsichtlich ihrer Übertragbarkeit auf den Eisenbahnbereich diskutiert, bewertet und es wurden mögliche Vorschläge ausgearbeitet. Das Ergebnis dieser Analyse wird in Kapitel 3 (Vorschläge für den Eisenbahnsektor) ausgeführt. Abschließend wird in Kapitel 4 (Anwendungsbeispiele) an Hand von Beispielen aus dem Bahnbereich das mögliche Verbesserungspotential hinsichtlich der Integration der Funktionalitäten von Stellwerk und Radio Block Centers bzw. der Integration der Funktionalität von Fahrzeugsteuerung/Zugsteuerung und Zugsicherung/Signalisierung vorgestellt.

Für die durchgeführten Recherchen wurde als Informationsquelle die Fachmeinung verschiedener Experten des Forschungsinstitutes (FOKUS Fachgruppe) und von langjährigen Partnern und Kunden eingeholt. Durch die Zusammenarbeit mit Kunden verschiedener Domänen verfügen die befragten Experten nicht nur über profunde Kenntnisse auf ihrem jeweiligen Fachgebiet (bspw. Testautomatisierung), sie haben auch Einblick darin, wie sich die zugehörigen Technologien, Methoden und Prozesse in den einzelnen Fachdomänen (z.T. recht unterschiedlich) entwickeln und auswirken. Dieses Wissen wurde über Befragungen und Interviews erfasst und dabei auch die bestehenden Kontakte zur Industrie genutzt, um (bspw. in Fachgremien) einzelne Aspekte aus dem Themen-/Fragenkatalog mit Vertretern der entsprechenden Industrie zu diskutieren. Um auch die Sichtweise und Erfahrungen des Auftraggebers (Eisenbahnbundesamt) in die Ausarbeitung mit einfließen zu lassen, wurde der Fragebogen auch mit Vertretern des Eisenbahnbundesamtes diskutiert.

Zur Absicherung wurden als sekundäre Informationsquelle auch öffentlich zugängliche Informationsquellen wie branchenspezifische Standards und Normen sowie richtungsweisende Veröffentlichungen einzelner Firmen als auch von Fach- oder Interessenverbänden für die Studie mit herangezogen. Aus diesen lassen sich ebenfalls Erkenntnisse darüber ableiten, welche technologischen Entwicklungen innerhalb einer Fachdomäne gerade stark diskutiert werden.

1.3 Managementzusammenfassung

Die Ergebnisse der Interviews und Recherchen können wie folgt zusammengefasst werden.

Produktstrukturen: Die Digitalisierung des schienengebundenen Verkehrs wird weiter zunehmen. Die Bahntechnik kann hier von vielen Entwicklungen anderer Bereiche profitieren, die sorgsam beobachtet werden sollten. Dementsprechend wird auch der Einsatz von kommerziellen und frei verfügbaren Standardkomponenten in der Bahnbranche weiter zunehmen. Das betrifft sowohl die Software als auch die Hardware. Die Kommunikation wird digitaler, bahnspezifische Kommunikationsstandards (z.B. GSM-R) werden eine geringere Rolle spielen, der Trend ist „all-IP“.

Auch die Intelligenz und Vernetzung der Steuergeräte und anderer informationsverarbeitender Systeme wird steigen. Die wesentlichen Funktionalitäten werden in Zukunft hauptsächlich durch Softwarekomponenten erbracht werden. Neue, flexiblere Architekturen werden es erleichtern, Softwarekomponenten gemäß den Systemanforderungen auf die Hardware zu verteilen. Das kann zu einer Reduktion der Anzahl der Steuergeräte führen; allerdings wirkt der Trend nach immer mehr Systemfunktionen dem entgegen.

Beim Entwurf von Komponenten und Systemen muss sowohl die langfristige Evolutionsfähigkeit als auch die Fähigkeit zur kurzfristigen Aktualisierung der Hard- und Software noch stärker berücksichtigt werden. Dafür gibt es im Wesentlichen drei Gründe:

- neue Anforderungen,
- der Umgang mit Obsoleszenz, und
- Sicherheitslücken in IT-Systemen.

Die Prozesse für die Qualitätssicherung und die Zulassung müssen daher die zügige Aktualisierung von Komponenten noch besser unterstützen.

Standards: Die Aufspaltung der Sicherheitsnormen für Softwareentwicklung gemäß den verschiedenen Anwendungsdomänen ist inhaltlich nicht immer zu rechtfertigen. Es sollte geprüft werden, ob für bahntechnische Anwendungen vermehrt auch Zertifizierungen aus anderen Fachdomänen in Betracht kommen. Insbesondere ist es nicht erforderlich, weitere unterschiedliche Normen innerhalb der Bahndomäne (z.B. für die Leit- und Sicherungstechnik und Fahrzeugsoftware) zu haben. Aus den Interviews ergab sich, dass in den meisten Fällen drei statt fünf Sicherheitsanforderungsstufen (keine, mittel, hoch) genügen.
Wichtige „neue“ Methoden und Techniken wie z.B. IT-Sicherheit, modelbasierte Entwicklung und agile Prozesse sind in der EN 50128 nicht adäquat repräsentiert. Hier bietet sich der "Supplement-Mechanismus" der DO-178C an.

Methoden und Prozesse: Modellbasierte Entwicklung erfordert eine bewusste Entscheidung für die geeignete Modellierungssprache. Auch sicherheitsgerichtete Systeme lassen sich, unter Voraussetzung entsprechender Granularität, mit agilen Methoden entwickeln. Funktionale Sicherheit und IT-Sicherheit müssen gemeinsam behandelt werden (da z.T. gegenläufige Anforderungen erfüllt werden müssen). Die Technologien für IT-Sicherheit bei Datenübertragung und Datenablage sind im Prinzip vorhanden, sind in der Bahnbranche aber noch nicht durchgängig etabliert. Es fehlt zumeist an einer durchgängigen IT-Sicherheitsarchitektur und den dazugehörigen Prozessen.

Ausbildung: Bahnsysteme wandeln sich immer mehr in IT-Produkte, daher entsteht ein immer größerer Bedarf an IT-Fachleuten und System-Ingenieuren. Softwaresicherheitsaspekte werden im allgemeinen Ingenieursstudium wenig berücksichtigt. Noch gravierender ist nach Aussagen von Industrievertretern das Fehlen von Kenntnissen und Fertigkeiten im Testen. Im Wettbewerb um die besten Köpfe steht die Bahnindustrie in Konkurrenz zu den IT-Unternehmen; in diesem Wettbewerb muss die Bahnbranche auch auf zeitgemäße Arbeitsmethoden und Softwaretechnologien setzen.

2 Entwicklung in anderen Fachdomänen

Im Rahmen der durchgeführten Recherche in den Domänen Automobilindustrie, Avionik, Telekommunikation und Industrieautomatisierung wurden Fachexperten der jeweiligen Domäne hinsichtlich ihrer Sichtweise auf Trends und Entwicklungen in ihrer Domäne befragt (siehe Anhang: Fragebogen). Die nachfolgenden Abschnitte dieses Kapitels enthalten jeweils einen groben Überblick über die aktuellen Themen in diesen Domänen, sowie eine kumulierte Zusammenfassung der in den Interviews geäußerten Meinungen dazu. Sie basieren daher sowohl auf relevanten Literaturquellen, als auch auf Fachwissen von Domänenexperten.

2.1 Automobilindustrie

In der über 125 Jahre alten Automobilbranche betrachtete man lange Zeit das Fahrzeug als metallene Maschinen auf Rädern, die sich vor allem durch den ihr eigenen Verbrennungsmotor auszeichnet. Entsprechend waren es zunächst vor allem Maschinenbauer, welche die Entwicklung des Industriezweiges und der Fahrzeuge bestimmt haben. Inzwischen beinhalten Fahrzeuge auch eine Vielzahl elektronischer Systeme, mit denen vor allem die Sicherheit und der Komfort im Fahrzeug deutlich erhöht wurden. Zwar hat seit den 1970er Jahren die Zahl der in den Fahrzeugen verbauten elektronischen Baugruppen stetig zugenommen (die Elektronik wiegt inzwischen mehr als der Motor), es dauerte aber erheblich länger bis auch Elektroingenieure eine größere Rolle beim Systementwurf des Fahrzeuges spielten. In noch stärkerem Maße gilt diese Aussage für Softwareingenieure und das Thema Software-Entwicklung für Fahrzeuge.

2.1.1 Produktstrukturen

Heutzutage ist das Auto ein nach innen und außen immer stärker vernetztes IT-System. Auch die Hinwendung zu alternativen Antriebstechnologien (Elektro- oder Hybridantriebe) beeinflusst die Entwicklung bzw. Veränderung im Fahrzeugbau maßgeblich. So wird etwa die bisherige mechanische Kraftübertragung vom Motor auf die Räder auf im Fahrzeug verteilte und elektronisch gesteuerte Antriebssysteme umgestellt. Das damit verbundene Potenzial für die Reduktion des Anteils der beweglichen Teile im Fahrzeug sowie das starke Interesse am autonomen Fahren haben dazu geführt, dass heute führende IT-Unternehmen wie Google, Apple, Intel, Infineon oder NVIDIA in den Markt eingetreten sind. Die gesamte Branche erlebt daher gerade, dass nicht mehr Maschinenbauer und Elektroingenieure, sondern maßgeblich Softwareingenieure die Architektur des Gesamtsystems Fahrzeugs bestimmen. Die traditionellen Hersteller haben diesen Trend erkannt, auch weil sie befürchten müssen, dass sie bei Verweigerung gegenüber diesen Trends sich schnell in einer Position wie die ehemals dominierenden Hersteller von Mobiltelefonen, Nokia und BlackBerry wiederfänden: Nach Einführung des iPhones durch Apple (Touchscreen Technologie und das von Unix herrührende Betriebssystem iOS) im Jahre 2007 und dem Nachziehen durch Google mittels der Android-Plattform sind die „Könige der Tastentelefone“ in kürzester Zeit komplett aus dem Markt verdrängt worden.

Straßenfahrzeuge werden im Gegensatz zu Eisenbahnen in sehr großer Stückzahl und im Wesentlichen für individuelle Kunden hergestellt. Die vom Marketing getriebene Konfigurierbarkeit von Automobilen ist mittlerweile so groß, dass es nur sehr selten vorkommt, dass zwei baugleiche Fahrzeuge das Band verlassen. Einhergehend damit werden die Produktentwicklungszyklen immer kürzer und die Verzahnung der Automobilhersteller mit ihren Zulieferern immer enger, da wesentliche Teile der für den Fahrzeughalter sichtbaren Funktionen von Zulieferern in Hard- und/oder Software implementiert werden.

Die Lebensdauer eines Autos ist in der Regel kürzer als die eines Eisenbahnfahrzeuges, der Produkt-Lebenszyklus beträgt lediglich 15-20 Jahre. Dennoch ist *Obsoleszenz* auch in der Automobilindustrie ein wichtiges Thema, da die Lebensdauer und Verfügbarkeit vieler elektronischer Komponenten deutlich kürzer ist als diese Zeitspanne. Da die elektronischen Bauteile die materielle Basis für die Fahrzeugsoftware bilden, ergeben sich daraus weitreichende Auswirkungen auf die Software-Entwicklungsprozesse in der Automobilindustrie. So besteht beispielsweise ein sehr großes Interesse, die Gerätesoftware so zu entwickeln, dass sie möglichst unabhängig von der Gerätehardware ist.

2.1.2 Standards

Der maßgebliche Standard für funktionale Sicherheit im Automobilbereich ist die ISO 26262 („Road vehicles – Functional safety"), die als Anpassung der allgemeiner Sicherheitsnorm IEC 61508 an die Anforderungen im Automobilbereich aufgefasst werden kann. Im zuständigen Normierungsgremium TC22/SC3/WG16 sind allerdings kaum Zulassungsbehörden vertreten. Die Norm ist seit 2011 in Kraft, eine neue Version ist in Vorbereitung (2019). Die Sicherheitsaspekte von Fahrzeugsoftware werden, analog zur IEC 61508, in einem Teil 26262-6 behandelt. Durch die Nähe der ISO 26262 zur IEC 61508 gibt es auch viele Gemeinsamkeiten mit den Bahnnormen EN 5012x. Ein prinzipieller Unterschied ist aber, dass bei der Zulassung von Fahrzeugen staatliche Stellen eine geringere Rolle spielen als im Eisenbahnbereich.

2.1.3 Methoden und Prozesse

Gegenwärtig werden die Methoden und Prozessen der Softwareentwicklung in der Automobilindustrie maßgeblich durch AUTOSAR (Automotive Open System Architecture) beeinflusst. AUTOSAR ist eine im Jahre 2003 gegründete weltweite Entwicklungspartnerschaft von Automobilherstellern, -zulieferern und Unternehmen der Software-, Halbleiter und Elektronikindustrie. Mit AUTOSAR soll eine standardisierte und offene Softwarearchitektur für vernetzte eingebettete Steuergeräte im Fahrzeug geschaffen werden, die umfassend konfigurierbar und für verschiedene Fahrzeugplattformen verfügbar ist. Auf methodischer Ebene gibt AUTOSAR vor, wie Systeminformationen und Steuergeräte beschrieben werden müssen, damit sie modulare, austauschbare funktionale Einheiten darstellen.

Übergreifendes Ziel von AUTOSAR ist es, durch die verbesserte Austauschbarkeit und damit einhergehende Wiederverwendung von Softwarekomponenten die Komplexität elektronischer Systeme im Fahrzeug besser zu beherrschen. Dadurch werden auch die Grundlagen für ein besseres Obsoleszenz-Management geschaffen, da es mit AUTOSAR leichter ist, Softwarekomponenten auf neuere Versionen der Steuergeräte zu übertragen.

Die erste Ausprägung von AUTOSAR bezeichnet man heute als *Classic AUTOSAR*, um sie leichter von der seit 2016 erfolgenden Entwicklung von *Adaptive AUTOSAR* abgrenzen zu können.

Classic AUTOSAR (ab 2003)

Auf unternehmerischer Seite war die Entwicklung von AUTOSAR vor allem vom Wunsch der Endhersteller getrieben,

- durch Synergien in der Systementwicklung, die Kosten für Steuergeräte insgesamt zu senken,
- Softwarekomponenten unabhängig von der konkreten Hardware zu beschreiben und insgesamt die Software portabler zu entwickeln,

- die Zahl der Steuergeräte im Fahrzeug zu reduzieren, und
- die Zulieferer austauschbarer machen.

Gerade der vierte Punkt hat, insbesondere zu Beginn der Entwicklung von AUTOSAR, den Druck auf die Zulieferer erhöht. Gleichzeitig gilt jedoch, dass eine einheitliche Architektur mit standardisierten Schnittstellen und einer einheitlichen Methodik auch für die Zulieferer Vorteile durch Kosteneinsparung bietet. Aus diesem Grund waren an der Definition von AUTOSAR auch große Zulieferer wie Bosch oder Continental von Anfang an maßgeblich beteiligt.

Eine wesentliche Motivation für die im zweiten Punkt genannte, leichtere Portierbarkeit von Softwarekomponenten war und ist ein verbessertes Obsoleszenz-Management durch die Hersteller, aber auch durch die Zulieferer. Die wesentlichen Vorteile von AUTOSAR für das Obsoleszenz-Management gründen sich auf einer klareren Trennung der Anwendungssoftware von der Hardware (siehe Abbildung 1: Übersicht von Classic und Adaptive AUTOSAR (nach **Bechter** (2015)). Diese Trennung wird im Wesentlichen dadurch erreicht, dass die Anwendungssoftware gegen die Softwareschnittstellen der AUTOSAR-Laufzeitumgebung programmiert wird wodurch man zu einer viel größeren Unabhängigkeit von der unterliegenden Hardware gelangt. Software, die weitgehend unabhängig von der Hardware entwickelt wurde, ist leichter auf neue Plattformen zu portieren, falls die ursprüngliche Ziel-Hardware abgekündigt oder nicht mehr verfügbar ist. Dadurch können Systeme, deren Lebenserwartung zu Ende geht, leichter durch neue ersetzt werden.

Technische Merkmale von AUTOSAR

Ein wesentlicher Aspekt der mehrschichtigen Softwarearchitektur von AUTOSAR ist die standardisierte Laufzeitumgebung, die zum einen als *Portabilitätsschicht* für die Softwareentwicklung dient und zum anderen als *Middleware* den Datenaustausch zwischen den Softwarekomponenten innerhalb eines Fahrzeugs ermöglicht. Dabei spielt es, aus softwaretechnischer Sicht, für die Kommunikation keine Rolle, ob die Softwarekomponenten auf einem oder verschiedenen Steuergeräten laufen. Diese *ortstransparente Kommunikation* bildet die Grundlage für die in AUTOSAR relativ flexible Zuordnung von Softwarekomponenten zu Steuergeräten. Sie ist die entscheidende Voraussetzung, um bei steigenden Rechenkapazitäten der Hardware durch Kollokation von Softwarekomponenten auf einem Steuergerät die Zahl der Steuergeräte im Fahrzeug zu verringern.

Die Laufzeitumgebung stellt Basisdienste nicht nur für die Kommunikation, sondern auch für die Speicherverwaltung oder die Systemdiagnose bereit. Generell lässt sich sagen, dass die Softwarekomponenten durch die Laufzeitumgebung von vielen Details der zugrundeliegenden Hardware oder des Betriebssystems isoliert werden.

Die Erstellung der Laufzeitumgebung für ein AUTOSAR-Steuergerät erfolgt heute weitestgehend mit *generativen* Ansätzen, was den Anteil von manueller, und damit fehlerträchtiger, Codeerstellung drastisch reduziert. Die Aufwände sind jedoch nicht völlig verschwunden, sondern treten nunmehr in Form der relativ komplexen Konfiguration der Laufzeitumgebung wieder auf.

Zusammengefasst lässt sich festhalten, dass die AUTOSAR-Laufzeitumgebung eine standardisierte Portabilitätsschicht ist, die es erleichtert, Softwarekomponenten auf andere Steuergeräte zu übertragen. In den Zeiten vor AUTOSAR hatte jeder Zulieferer seine eigene Laufzeitumgebung definiert, was zu teuren Mehrfachentwicklungen führte. Mit AUTOSAR können sich die Hersteller bei der Entwicklung, dem Test und der Zulassung stärker auf die eigentliche Funktionalität ihrer Softwarekomponenten bzw. Steuergeräte konzentrieren.

AUTOSAR-Methodik

Ein wichtiger Vorteil von AUTOSAR besteht darin, dass die Methodik neue Wege beim Systementwurf und dabei insbesondere bei der funktionalen Dekomposition erlaubt. Vor AUTOSAR wurde die Entscheidung, welche Systemfunktionen auf welchen Steuergeräten laufen, oft durch die Zulieferer definiert. Dies führte zu einer sehr großen Anzahl von Steuergeräten im Fahrzeug, was nicht nur das Gewicht und den Energieverbrauch erhöhte, sondern auch die Integration, Wartung und Lebensdauer des Fahrzeugs erschwerte. Mit AUTOSAR ist es einfacher, Softwarekomponenten vom Steuergerät *unabhängig* zu beschreiben und zu entwickeln. Die Zuordnung von Softwarekomponenten zu den Steuergeräten erfolgt heutzutage nicht mehr allein gemäß den Bedürfnissen des Zulieferers, sondern nach übergeordneten Systemkriterien. Ein wesentlicher Erfolg der AUTOSAR-Methodik ist, dass es heutzutage üblich ist, mehrere Softwarekomponenten auf einem Steuergerät auszuführen.

In der Praxis ist aber zu beobachten, dass die angestrebte Reduktion der Zahl der Steuergeräte im Automobil nicht in dem erhofften Maße eingetreten ist. Das ist jedoch kein Zeichen für das Scheitern von AUTOSAR. Vielmehr ist festzustellen, dass die Systemfunktionen im Fahrzeug deutlich komplexer geworden sind und mehr Softwarekomponenten für deren Umsetzung benötigt werden. Der Erfolg von AUTOSAR zeigt sich mithin dadurch, dass es den Hersteller und Zulieferer gelungen ist, die gewachsene Komplexität der Fahrzeugfunktionen weiterhin zu beherrschen.

Implementierungsaspekte

Die AUTOSAR-Laufzeitumgebung bietet eine Programmierschnittstelle für die Programmiersprache C an. Dementsprechend werden AUTOSAR-Softwarekomponenten typischerweise in C programmiert oder der Code ihrer Komponenten wird aus abstrakteren Modellen (z.B. MATLAB/Simulink) nach C generiert. Die ausführbare Software, bestehend aus Anwendungssoftware, AUTOSAR-Laufzeitumgebung und eingebettetem Betriebssystem, ist am Ende ein durch den Integrator erstelltes monolithisches Kompilat, das direkt auf der Hardware des Steuergeräts läuft. Konkret bedeutet letzteres, dass es nicht einfach möglich ist, einzelne Softwarekomponenten nach der Auslieferung zu ersetzen. In der Regel muss bei Fehlerbehebungen die gesamte AUTOSAR-Software eines Steuergerätes neu kompiliert und aufgespielt werden.

Die Benutzung von C als Implementierungssprache war für die Akzeptanz von AUTOSAR sehr wichtig, denn C ist wegen seiner vergleichsweisen hardwarenahen Konzepte für die Programmierung eingebetteter Systeme seit langem etabliert. Andererseits macht es gerade das geringe Abstraktionsniveau von C nicht einfach, komplexere Anwendungskonzepte adäquat in Software abzubilden.

Adaptive AUTOSAR (ab 2016)

Im Laufe des mehr als zehnjährigen Einsatzes von Classic AUTOSAR ist aus den Erfahrungen und durch neue Anforderungen das Bedürfnis entstanden, die AUTOSAR Architektur umfassend zu modernisieren. Obwohl AUTOSAR viele positive Auswirkungen auf den Systementwurf und den Softwareentwicklungsprozess in der Automobilindustrie hatte und hat, ist es durch neue Anforderungen deutlich geworden, dass AUTOSAR weiterentwickelt werden muss. Die wesentlichen Treiber sind dabei:

- das autonome Fahren,
- der Einzug von Elektro- und Hybridantrieben,
- die stärkere Vernetzung der Fahrzeuge mit ihrer Umgebung, sowie
- das Bedürfnis, bestimmte Softwarekomponenten leichter installieren und aktualisieren zu können.

Die aufwändige Verarbeitung zahlreicher Sensordaten, wie sie z.B. beim autonomen Fahren entstehen, macht es erforderlich, dass statt vieler, relativ kleiner eingebetteter Steuergeräte, jetzt auch leistungsstarke Rechner mit POSIX Betriebssystemen (typischerweise Linux) ein zentraler Bestandteil der Netzwerktopologie im Fahrzeug sind. Für die Übertragung der stark angewachsenen Datenmengen spielen neben den klassischen Fahrzeugbussen wie CAN oder Flex Ray jetzt auch „neuere" Ansätze wie Ethernet zentrale Rollen im Fahrzeug. Diese Hinwendung zu Standardbetriebssystemen unterstützt auch den Übergang zu moderneren und flexibleren Ansätzen bei der Software-Architektur. In erster Linie ist hierbei der Ansatz der service-orientierten Architekturen zu nennen, bei dem Softwarekomponenten als *Dienste* gekapselt werden. Die Aufgabe des Systemarchitekten besteht dann in der *Koordinierung* der Dienste, während die technischen Details der Dienste weitgehend hinter standardisierten Schnittstellen versteckt werden. Ein Vorteil des service-orientierten Ansatzes ist, dass Softwarekomponenten leichter aktualisiert werden können, da sie nur über die Schnittstellen der Services mit anderen Komponenten kommunizieren.

Während sich Classic AUTOSAR auf die statisch konfigurierte Software-Architektur von eingebetteten Steuergeräten konzentriert und die Architekturen von Infotainment-Geräten weitgehend außer Acht lässt, sieht Adaptive AUTOSAR ergänzend einen Bereich mit einer dynamischeren Software-Architektur vor (Abbildung 1: Übersicht von Classic und Adaptive AUTOSAR (nach **Bechter** (2015))). In der Abbildung sind auch die wesentlichen Bestandteile von Classic AUTOSAR dargestellt. Das Ziel der Erweiterung von AUTOSAR um Adapative AUTOSAR ist es, in einem langfristigen Prozess, geeignete Softwarekomponenten aus Classic AUTOSAR bzw. dem Infotainment-Bereich, in die neue, flexiblere Architektur von Adaptive AUTOSAR zu überführen.

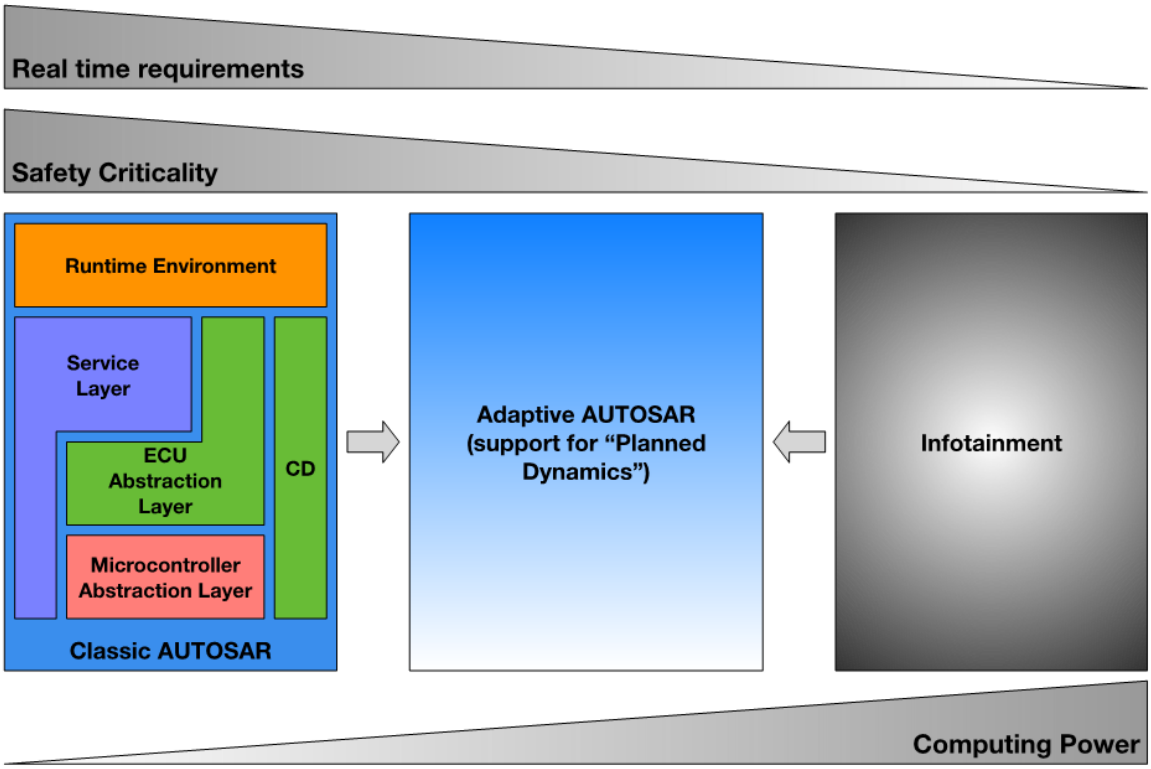

Abbildung 1: Übersicht von Classic und Adaptive AUTOSAR (nach **Bechter** (2015))

Für die Softwareentwicklung bedeutet das im Unterschied zu Classic AUTOSAR unter anderem:

- Softwarekomponenten werden zur Laufzeit in (POSIX-) Prozessen ausgeführt. Die Adressräume der verschiedenen Prozesse sind voneinander getrennt, was einen besseren Schutz der Softwarekomponenten zur Laufzeit garantiert.
- Eine noch stärkere Trennung von Softwarekomponenten erlaubt die ebenfalls vorgesehene Virtualisierung der Hardware.
- Die Multi-Core-Verarbeitung ist in Adaptive AUTOSAR ebenfalls vorgesehen. Damit steht im Fahrzeug noch mehr Rechenleistung zur Verfügung, ohne dass zusätzliche Computer eingesetzt werden müssen.
- Die Interprozesskommunikation wird über serviceorientierte Ansätze (SOA) erfolgen. Damit verbunden ist die eine stärkere Betonung der Schnittstellenspezifikation und des dynamischen Bindens anstelle einer direkten, statischen Referenzierung konkreter Komponenten.

Als Implementierungssprache für Adaptive AUTOSAR ist C++ vorgehen. Diese Entscheidung ist aus den folgenden Gründen sinnvoll:

- C++ ist im hohen Maße mit der Programmiersprache C kompatibel. Letztere wird schon länger in der Automobilindustrie eingesetzt.
- C++ erlaubt ebenso wie C, hardwarenahe Konzepte effizient abzubilden.
- Gleichzeitig ist es mit C++ deutlich einfacher als mit C, domänenspezifische Abstraktionen zu entwerfen.
- C++ wird nicht von einer einzelnen Firma kontrolliert, ganz im Gegensatz zu anderen weitverbreiteten Programmiersprachen wie Java (Oracle), C# (Microsoft) oder Swift (Apple).

Trotz dieses radikal anderen Ansatzes muss man im Auge behalten, dass Adaptive AUTOSAR die klassische Ausprägung von AUOTSAR nicht verdrängen soll. Gerade die Interoperabilität mit Software-Komponenten des Classic AUTOSAR spielt bei der aktuell stattfindenden Entwicklung von Adaptive AUTOSAR eine wichtige Rolle.

Durch die stärkere Vernetzung des Fahrzeugs mit externen IT-Systemen, die insgesamt größere Dynamik der Softwarekomponenten sowie die sich durch Multicore-Verarbeitung und Virtualisierung ändernden Systemarchitekturen (vergleiche **Fuchsen** (2018)) stellt sich in Adaptive AUTOSAR in einem noch stärkeren Maße die Frage nach der Sicherstellung von funktionaler Sicherheit (Safety) und IT-Sicherheit (Security). An der Beantwortung dieser Fragen, die auch für die Bahnindustrie sehr wichtig sind, wird gegenwärtig intensiv gearbeitet.

2.1.4 Ausbildung

Die Automobilindustrie in Deutschland war und ist stark innovationsgetrieben und ein Magnet für Talente aus vielen Ingenieursdisziplinen. Es bestehen sehr enge Forschungskooperationen zwischen Automobilherstellern, Zulieferern sowie Universitäten und außeruniversitären Forschungseinrichtungen. Die TU München hat beispielsweise einen Studiengang „Automotive Software Engineering“ eingerichtet, siehe **TU München** (2018).

Diese Kooperationen sind international ausgerichtet und keineswegs auf Deutschland beschränkt. Die Branche genießt eine hohe Reputation bei Entwicklern, was auch, aber nicht nur, an den attraktiven Gehältern liegt, die die Automobilindustrie zu bezahlen in der Lage ist.

Aktuell investiert die Automobilindustrie massiv in die Entwicklung von Technologien für das autonome Fahren und führt parallel dazu elektrische Antriebstechniken ein. Neu ist, dass sie dabei sowohl in tech-

nologischer als auch personeller Konkurrenz zu Unternehmen wie Google, Apple oder Intel steht. Es handelt sich dabei um die innovativsten IT-Unternehmen der Welt, denen es nicht schwerfällt, die besten Softwareingenieure zu rekrutieren.

Der Übergang von Classic zu Adaptive AUTOSAR verlangt, dass aus C-Programmierern für eingebettete Systeme in Zukunft C++-Programmierer für komplexe und dynamische Softwarekomponenten werden müssen. Dazu müssen auch entsprechende Weiterbildungsprogramme entwickelt werde. In diesem Zusammenhang sind die vom AUTOSAR Konsortium vorgelegten C++-Programmierrichtlinien zu erwähnen, deren Ziel es ist, die große Anzahl von neuen Merkmalen der modernen C++ Standards sinnvoll und sicher einzusetzen.

2.2 Avionik

Üblicherweise wird die Luftfahrtindustrie als Teil der größeren Domäne Avionik (Luft- und Raumfahrt) betrachtet. Wir konzentrieren uns in diesem Abschnitt auf die zivile Luftfahrt, da funktionale Sicherheit hier eine wesentlich größere Rolle spielt als in der Raumfahrt. Der Hauptgrund hierfür ist, dass in der Raumfahrt, ebenso wie in der militärischen Luftfahrt, der Transport von Zivilpersonen praktisch nicht vorkommt. Dass sich dies in Zukunft vielleicht ändern wird, ist für diese Studie nicht relevant.

2.2.1 Produktstrukturen

Flugzeuge sind, ebenso wie Züge, langlebige und sehr teure Investitionsgüter, die extrem hohen Sicherheitsanforderungen unterliegen. Das gilt selbstverständlich auch für die Systeme der Luftraumüberwachung. Die wenigen großen Hersteller sind meist auch global tätig. Vom Airbusmodell A350, zum Beispiel, sind bisher mehr als 850 Exemplare von über 50 Fluggesellschaften (von allen Kontinenten) bestellt worden. Auch die Zulassung von Flugzeugen ist deutlich globaler organisiert als in der Bahnindustrie: ist ein Flugzeug in Europa und den USA zugelassen, kann es praktisch weltweit eingesetzt werden.

Ebenso wie in der Automobil- und Bahnindustrie hängt ein immer größerer Teil der Funktionalität im Flugzeug von vernetzten, eingebetteten Steuergeräten ab. Im Luftfahrtbereich spielen zum einen die extrem hohen Sicherheitsanforderungen sowie die Beschränkungen an die Masse und den Energieverbrauch der Steuergeräte eine entscheidende Rolle. Dazu kommt durch die sehr lange Lebensdauer der Flugzeugtypen das Problem der Obsoleszenz von Bauteilen, Software und Prozessen. (Der Jungfernflug beispielsweise der Boeing 737 fand 1967 statt. Diese Flugzeugfamilie wird heute noch bei mehr als 500 Fluggesellschaften eingesetzt.)

2.2.2 Methoden und Prozesse

Analog zu den Methoden und Prozessen der Automobilindustrie (Abschnitt 2.1.1) betrachten wir in diesem Abschnitt kurz neuere Entwicklungen bei der Soft- und Hardwarearchitektur in der Domäne. Dabei muss man beachten, dass es erhebliche Unterschiede zwischen diesen Domänen gibt. Da sowohl die Eisenbahn als auf die Luftfahrt im Vergleich zur Automobilindustrie nur geringe Stückzahlen produzieren, ist der Kostendruck bei ersteren nicht so hoch. Andererseits sind die Themen Gewicht, Anzahl und Energieverbrauch von Steuergeräten bei Luftfahrt und der Automobilindustrie von viel größerer Bedeutung als bei der Eisenbahn.

Line-Replaceable Units (LRU)

Flugzeuge fliegen Ziele an, die sehr weit von größeren Wartungsbasen entfernt sein können. Zur Vereinfachung der Wartung von Flugzeugen im Feld, das heißt, außerhalb von Werkstätten, wurde das System der *Line-Replaceable-Unit* (LRU) entwickelt. Man versteht darunter Bauteile, die über standardisierte Steckkontakte und eine klar definierte Funktionalität verfügen. Solche Bauteile können mit einfachen Handgriffen ausgebaut und ersetzt werden. Oft können LRUs unterschiedlicher Hersteller eingesetzt werden.

Integrated Modular Avionics (IMA)

Das System der *Integrierten Modularen Avionik* (englisch Integrated Modular Avionics, IMA) stellt eine Weiterentwicklung des LRU-Konzepts dar. Man versteht darunter modulare Steuergeräte für den Einsatz im Flugzeug, die aus standardisierten Komponenten aufgebaut sind. Im Unterschied zu AUTOSAR geht bei IMA nicht nur um die Softwarearchitektur, sondern auch um standardisierte Hard- und Softwareschnittstellen zur Kommunikation zwischen den verschiedenen Teilsystemen eines Flugzeugs. Die wesentlichen Vorteile gegenüber dem LRU-Konzept bestehen in

- der Portabilität der zu entwickelnden Software und
- dem erheblichen Potenzial zur Gewichtseinsparung durch die Zusammenführung bisher verschiedener LRUs auf einem Steuergerät.

Ein besonderer Aspekt ist dabei, dass IMA-Module echtzeitfähige Steuergeräte sind, die es prinzipiell erlauben, Applikationen unterschiedlicher Kritikalitätslevel sicher auszuführen. Letzteres wird über Anforderungen an das zugrundeliegende Betriebssystem erreicht. Bei der Zertifizierung von IMA-Systemen stellt der Einsatz von Mehrkernprozessoren immer noch eine große Herausforderung dar. Das liegt vor allem darin begründet, dass die Performanz eines Kerns eines Multiprozessorsystems, durch die gemeinsame Benutzung des Hauptspeichers, stark von der Auslastung eines anderen Kerns abhängen kann. Daher ist insbesondere der Nachweis von Echtzeiteigenschaften eines Mehrprozessorsystems oft mit Schwierigkeiten verbunden.

2.2.3 Standards

Die DO-178C (Software Considerations in Airborne Systems and Equipment Certification) aus dem Jahr 2012 ist in der zivilen Luftfahrt der grundlegende Standard für die Zertifizierung softwarebasierter Systeme. Er ersetzte die Version DO-178B aus dem Jahre 1992.

Neben verschiedenen Klarstellungen und einer genaueren Terminologie, im Vergleich zur DO-178B, bestehen die wesentlichen Änderungen in einem separaten Standard für die Qualifizierung von Softwarewerkzeugen, nämlich der DO-330 (Software Tool Qualification Considerations).

Daneben gibt es noch die drei folgenden, ergänzenden Standards (sogenannte *Supplemente*)

- DO-331: Modellbasierte Entwicklung und Verifikation
- DO-332: Objektorientierung und verwandte Technologien
- DO-333: Formale Methoden

Darin werden moderne Softwaretechnologien in Bezug auf die DO-178C dargestellt. Generell lasst sich sagen, dass die Detailtiefe dieser Standards weit über die der Anhänge der EN 50128 hinausgeht. Ande-

rerseits ist festzuhalten, dass die ergänzenden Standards weder modellbasierte Entwicklung noch formale Methode von sich aus effizienter als traditionelle Methoden der Softwareentwicklung und -verifikation ansehen. Vielmehr gelten sie als neue Entwicklungs- oder Verifikationsmittel, die dazu beitragen können, die Qualitätssicherungsziele der DO-178C zu erreichen.

Im Folgenden gehen wir kurz auf die einzelnen Supplemente ein.

Model-Based Development and Verification Supplement (DO-331)

Unter einem Modell versteht die DO-331 *eine abstrakte Darstellung von Softwareaspekten eines Systems.* Der Vorteil von Modellen liegt unter anderem in:

- der eindeutigen Darstellung von Anforderungen und der Architektur,
- der Unterstützung der automatischen Codegenerierung,
- der Unterstützung der automatisierten Testgenerierung und
- der Unterstützung des Einsatzes von Analysewerkzeugen zur Verifikation der Anforderungen und der Architektur.

Es wird auch klargestellt, dass

- Abbildungen ohne Syntax bzw. Semantik sowie
- Gleichungen, die sich auf Sätze einer natürlichen Sprache beziehen

keine Modelle im Sinne der Norm sind. Die DO-331 erläutert welche Punkte beim Einsatz von modellbasierter Entwicklung beachtet werden müssen, damit die Sicherheits- und Integritätsziele erreicht werden.

Die DO-331 verfügt im Anhang, ebenso wie die DO-332 und DO-333, einen Abschnitt über häufig gestellte Fragen (FAQ) zur Modellierung. Solche informellen Ergänzungen sind sowohl für Praktiker als auch für Zulassungsbehörden von großer Hilfe.

Object-Oriented Technology and Related Techniques Supplement (DO-332)

Die DO-332 beginnt mit der Beobachtung, dass objektorientierte Technologien in unkritischen Softwareentwicklungsprojekten weit verbreitet seien und dass der Einsatz dieser Technologie für kritische Softwareanwendungen in der Luftfahrt zugenommen habe. Ziel des Supplements ist es, Hinweise für den Einsatz von objektorientierten und verwandten Technologien zu geben, damit die Sicherheits- und Integritätsziele erreicht werden.

Hervorgehoben wird, dass viele Eigenschaften der Objektorientierung von deren konkreter Umsetzung in den einzelnen Programmiersprachen abhängen. Außerdem werden grundlegende Begriffe und Techniken kompakt erläutert, was insbesondere für die Kommunikation zwischen Softwareherstellern und Zertifizierungsbehörden von großem Vorteil ist.

Zu diesen Begriffen zählen unter anderem

- Klassen und Objekte,
- Typen, Typsicherheit und Typumwandlungen,
- hierarchische Kapselung,
- Polymorphismus (einschließlich parametrischem Polymorphismus), und

- Behandlung von Ausnahmen.

Zusammenfassend lässt sich sagen, dass die DO-332 mit ihren fast 150 Seiten Umfang, die auch Antworten auf häufig gestellte Fragen enthält, in ihrer Detailtiefe die Ausführung der EN 50128 weit übertrifft.

Formal Methods Supplement (DO-333)

Die DO-333 betont, analog zur EN 50128, dass formale Methoden mathematisch fundierte Techniken für die Spezifikation, Entwicklung und Verifikation von Softwareaspekten von Systemen sind. Die mathematischen Grundlagen formaler Methoden bestehen aus formaler Logik, diskreter Mathematik und maschinell verarbeitbaren Sprachen. Der Einsatz formaler Methoden ist durch die Erwartung motiviert, dass mittels mathematischer Analysen die Korrektheit oder andere wichtige Eigenschaften eines Softwaresystems definitiv nachgewiesen werden können.

Als Beispiele für solche wichtigen Eigenschaften nennt die DO-333

- die Abwesenheit von Ausnahmebedingung und Verklemmungen (deadlocks),
- die Nicht-Interferenz zwischen verschiedenen Kritikalitätsstufen,
- die Worst-Case-Laufzeit sowie Schranken für die Größe des Programmstacks zur Laufzeit,
- die Abwesenheit von unbeabsichtigter Funktionalität, und
- korrektes Synchronisationsverhalten.

Es handelt sich dabei um Eigenschaften, deren Nachweis mit Tests allein notorisch schwierig ist.

Im Gegensatz zur DO-333 hat die Darstellung von formalen Methoden in der EN 50128 (2011) mehrere wesentliche Probleme:

- Es handelt sich um eine bloße Fortschreibung des Stands von 2001, die neuere Entwicklungen nicht berücksichtigt.
- Es ist nach wie vor unklar, wie sich die Ergebnisse einer formalen Verifikation zu funktionalen Tests und Testüberdeckungskriterien der Norm verhalten.

Im Gegensatz dazu diskutiert die DO-333 in einem Anhang verschiedene *konkrete* Anwendungen von formalen Methoden. Dabei wird an Hand eines Beispiels dargestellt, wie

- informale Anforderungen mittels Prädikatenlogik erster Stufe formalisiert,
- Unittests, d.h. Modultests, durch formale Unit-Beweise eines Verifikationswerkzeugs ersetzt, und
- Nachweise für die integrierte Software mittels Tests erbracht

werden können.

2.2.4 Ausbildung

Auch die Luftfahrtindustrie pflegt ähnlich wie die Automobil- und Bahnindustrie enge Kooperationen mit Universitäten und Forschungseinrichtungen. Wegen der Bedeutung der Avionik für die nationale Sicherheit, gibt es in verschiedenen Staaten spezielle Organisationen für die Forschung auf diesem Gebiet. Man denke beispielsweise an die NASA in den USA, die ESA in Europa beziehungsweise die DLR auf nationaler Ebene. Andererseits sind diese Aktivitäten oft nicht auf den Luftfahrtbereich beschränkt; so ist die DLR auch stark in der Forschung für die Bahn- und Automobilindustrie vertreten.

Analog zur Bahnindustrie stellen die sehr langen Produktlebenszyklen die Luftfahrindustrie vor große Herausforderungen beim Gewinnen und Halten von qualifiziertem Personal. Wie schon im Abschnitt 2.1.4 erläutert, verfügen bestimmte Unternehmen der IT-Branche über eine Reputation und Finanzkraft, die es anderen Branchen immer schwerer macht, hinreichend attraktiv für sehr gute Softwareentwickler zu sein.

2.3 Telekommunikation

Kommunikation ist heutzutage aus unserem alltäglichen Leben, insbesondere aber auch aus technischen Systemen nicht mehr wegzudenken. Die nachfolgende Bestandsaufnahme zum Telekommunikationssektor entstammt zu wesentlichen Teilen einer im Auftrag des BMVI erstellten Studie zu Netzinfrastrukturen der Gigabitgesellschaft („Gigabit-Studie“, **Fraunhofer FOKUS** (2016)) und wurde hier um Bezüge zum Bahnsektor und nicht zuletzt auch zu zukünftigen Entwicklungen ergänzt.

2.3.1 Produktstrukturen

„Das Internet“ erscheint oftmals verkürzt als ein homogenes Netz, das die Dienste aller traditionellen Kommunikationsnetze anbietet und diese ursprünglichen Netze (z.B. das Telefonnetz) sukzessive ersetzt. Somit bliebe am Ende nur ein großes, vereinheitlichtes Netz übrig. Die Praxis stellt sich jedoch vielschichtiger dar: Inzwischen wird in fast allen Netzen das Internetprotokoll (IP) eingesetzt oder zumindest darauf geachtet, dass eine Umsetzung zum Internetprotokoll an Netzübergängen leicht möglich ist. Neben dem offenen Internet – das sich durch den globalen Zusammenschluss von Netzen und einen einheitlichen, öffentlichen Adressraum konstituiert – gibt es geschlossene Weitverkehrsnetze ohne direkten Übergang in das Internet als Basis für IT-Sicherheit. Beispiele dafür sind Firmen- oder Verwaltungsnetze, z.B. auch das Netz für Behörden und Organisationen mit Sicherheitsaufgaben (BOS) oder GSM-R und bbIP der Deutschen Bahn. Dazwischen gibt es eine Reihe von Abstufungen. In der folgenden Darstellung (aus **Fraunhofer FOKUS** (2016), S. 13) ist exemplarisch die Nutzung eines Spezialnetzes dargestellt, das bspw. für die Verteilung von Videodatenströmen zum Einsatz kommen könnte, um Videos effizient und qualitativ hochwertig zur Verfügung zu stellen.

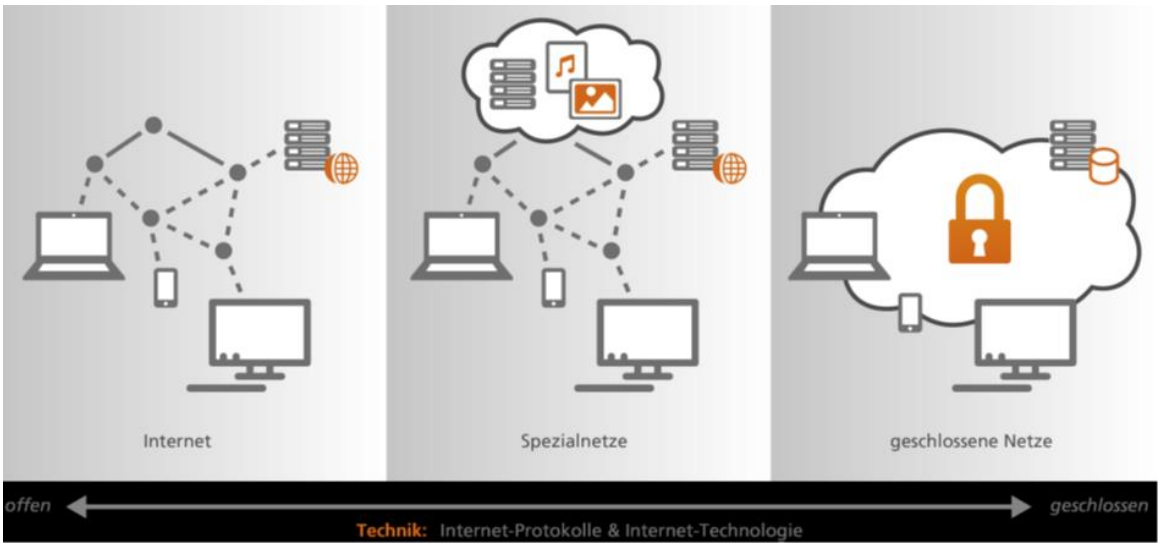

Abbildung 2: Netztypen (Internet, Spezialnetz, privates Netz) (aus **Fraunhofer FOKUS** (2016))

Diese verschiedenen Netzkonfigurationen werden in der Praxis den jeweiligen Anforderungen an Offenheit und Sicherheit entsprechend eingesetzt. Die drei in der Abbildung dargestellten Netztypen (Internet, Spezialnetz, privates Netz) zeigen die Spannbreite der Möglichkeiten, die Übergänge sind dabei fließend.

Mit dem Internet steht eine offene, transparente und globale Kommunikationsinfrastruktur zur Verfügung, die aus einem Zusammenschluss vieler voneinander unabhängiger Netze besteht, die nur eine geringe Anzahl von technischen und administrativen Regeln teilen. Aus technischer Sicht gibt es nur wenige zentrale Netzfunktionen; auf einfachste Weise ist eine spontane, weltweite Kommunikation mit kleinen oder großen Kommunikationsteilnehmern möglich. Diese Offenheit ist eine wichtige Grundlage für Innovation und Wachstum von Netz und Anwendungen. So kann bspw. eine neue Anwendung mit einer kleinen Anzahl von Nutzern weltweit gestartet werden, ohne dass irgendwelche Vorkehrungen im Netz getroffen werden oder besondere Infrastrukturen oder Anschlüsse zur Verfügung stehen müssen.

Die Nutzung des offenen Internets hat allerdings auch zwei Nachteile, die von Bedeutung sind:

- Nutzer setzen sich Sicherheitsrisiken aus und
- bei der Übertragung kann die Dienstqualität (QoS) nur sehr eingeschränkt sichergestellt werden.

Ist ein Gerät direkt über das Internet erreichbar, so können Sicherheitslücken dieses Geräts von jedem Ort der Welt aus ausgenutzt werden. Selbst wenn man nur gut überprüfte und damit sehr sichere Systeme einsetzen würde, kann zumindest eine Dienstblockade aufgrund mutwilliger Überlastung (Denial of Service-Angriff) auf Grund der weltweiten Erreichbarkeit prinzipiell nicht ausgeschlossen werden. Zudem stellt das IP-Protokoll nur wenige Möglichkeiten bereit, um die Qualität der Übertragung durch den Nutzer eines Diensts (also technisch durch das Endgerät) zu steuern.

Diese Eigenschaften gelten im Internet als „störend“, und werden bei der Abwägung zwischen Nutzen und Bedrohung in Kauf genommen. In einem Netz der kritischen Infrastruktur sind solche „Störungen“ nicht tolerierbar, ohne wesentliche Eigenschaften bezüglich Verlässlichkeit und Integrität aufzugeben. In Abschnitt 2.3.2 wird aufgezeigt, dass diese offensichtlichen Nachteile durch die nächste Generation des Internets wirkungsvoll angegangen werden.

Diese nächste Generation der Mobilfunkstandards – allgemein unter der Bezeichnung 5G bekannt – gliedert das künftige Netz in zwei Bereiche; einen Bereich Zugangsnetz, dessen Eigenschaften und Technologien den Schwerpunkt der erwähnten Gigabit-Studie bilden und hier auch nicht weiter behandelt werden und ein sogenanntes Kernnetz, welches im Kontext der Bahnindustrie eher relevant ist. Beide Bereiche werden im Rahmen von 5G neu aufgebaut und auch beide Bereiche tragen zur Funktionserweiterung und Geschwindigkeitssteigerung künftiger Netze bei.

Es sind bei der Entwicklung der Funktionalitäten des Kernnetzes – wie in vielen anderen Anwendungsbereichen von Informationstechnik Technologien (IT-Technologien) – langfristige technische Entwicklungen zu sehen, bei denen zunächst

1. die Herausbildung einer Plattformarchitektur erfolgt, die sich an der Hardware – Software Schnittstelle orientiert wie z.B. auch in der Luftfahrt beim IMA Standard oder in der Automobilindustrie beim AUTOSAR Standard geschehen.
2. Anschließend sind die wesentlichen Funktionalitäten der Anwendungslogik in Software repräsentierbar und
3. die ursprünglich durch Hardware definierte Plattformfunktionalität wird immer mehr durch Softwarefunktionalitäten wie z.B. Virtualisierung angereichert. Auch diese Schritte sind bei der Weiterentwicklung von Classic AUTOSAR zu Adaptive AUTOSAR nachvollziehbar (siehe Abschnitt 2.1).

Was ein 5G Kernnetz damit gegenüber seinen Vorgängern charakterisiert, ist die Verlagerung der quasi eigenen Managementfunktionen in Softwarekomponenten und anschließend die optional freizügige Verteilung im Netz. Das ‚sichtbare' Anwendernetz ist virtualisiert; man spricht von Software Defined Networks (SDN). Beim *Cloud-Computing* wird die komplette IT-Infrastruktur über ein Rechnernetz zur Verfügung gestellt, ohne dass diese auf dem lokalen Rechner installiert sein muss. Beim *Edge Computing* werden Anwendungen, Daten und Dienste von zentralen Knoten (Rechenzentren) weg zu den äußeren Rändern eines Netzwerks verlagert. Datenströme werden ressourcenschonend zumindest teilweise an Ort und Stelle (z. B. direkt am Endgerät oder innerhalb einer Fabrik) verarbeitet. Statt Edge Computing wird gelegentlich auch von *Fog Computing* gesprochen. Bei Fog Computing liegt der Fokus allerdings weniger auf den Endgeräten, sondern vielmehr darauf, die Cloud-Ressourcen näher zu den Anwendungen zu bringen (Dezentralisierung). Für den Anwender sind die Grenzen zwischen "Cloud Computing", "Edge Computing" oder "Fog Computing" bei SDN fließend; Anwendungslogik kann nach beliebigen Kriterien alloziert und verschoben werden. Von der physischen Netzebene kann praktisch komplett abstrahiert werden; es spielt keine Rolle welches konkrete Kommunikationsmedium zur Anwendung kommt. Die Eigenschaften und Fähigkeiten des Netzes werden durch Software definiert und zum großen Teil auch ausgeführt.

Mit dieser Entwicklung einher geht natürlich auch der Verlust bisher bekannter Produktstrukturen. So sind z.B. Telefonanlagen als eigenständige Geräte im Zeitalter von VoIP obsolet und durch Vermittlungssoftware ersetzt worden, die per Web-Interface frei konfigurierbar irgendwo in der Cloud ausgeführt wird. Oder es werden z.B. Storage und Computing Ressourcen - sogar in komplexen Formen - in der Regel (nur noch) als ‚Cloud'-Dienste angeboten.

Übertragen auf den Bahnbereich sind die Entwicklungen im Bereich der Leit- und Sicherungstechnik (LST) sehr gut auf „sichere, verteilte Rechenzentren" auf der Basis von 5G Plattformen abbildbar; dies bedeutet dann aber, Stellwerke werden auf die minimale logische Stellwerksfunktion reduziert und sind in Software repräsentierbar. Vor diesem Hintergrund muss die Definition und Etablierung tragfähiger Plattformstandards im Bahnbereich dringend angemahnt werden um einheitliche (herstellerübergreifende, internationale) Produktstrukturen in diesem Segment zu ermöglichen.

2.3.2 Standards

Der Bereich der Telekommunikation ist auf allen Ebenen sehr stark von Standards geprägt. Die Notwendigkeit der Interoperabilität zwischen verschiedenen Herstellern und der weltweite Markt machen dies unumgänglich. Das klassisch zu nennende OSI Schichtenmodell gibt hier seit mehreren Jahrzehnten einen klaren logischen Ordnungsrahmen vor. Quasi als Rückgrat hat sich das Internetprotokoll (IP) basierend auf den Standards der IEEE 802.3 Familie herausgebildet. Zum Stand der Standardisierung im Einzelnen:

Netzebene

Die Standardisierung der nächsten Mobilfunkgeneration 5G ist noch im Gange. Standards werden 2018-2020 erwartet (Phasenansatz). Bisher haben sich 3GPP, ITU, ETSI, CTIA, 4G/5G Americas, NIST, GSMA, Small Cell Forum, IEEE und NGMN zur Entwicklung dieser Standards bekannt. Viele weitere Standards Development Organizations (SDOs) sind zusätzlich mit der Standardisierung von Enablern für 5G befasst. Wie unter dem Punkt Produktstruktur schon ausgeführt bietet insbesondere das Management von Netzen durch ‚Slicing' das Potential sehr viele (bisherige) Spezialnetze in das „Internet" zu integrieren.

Die Relation zu den im Bahnbereich genutzten Netzen wird an dieser Stelle sehr offensichtlich. Betrachtet man GSM-R und bbIP als die aktuellen Standardtechnologien und vergleicht dies mit den Entwicklungen im Bereich 5G, wird schnell deutlich, dass hier erhebliche Synergien möglich sind. Dies ist im Bahnbereich inzwischen erkannt und die Entwicklung wird unter dem Begriff FRMCS (Future Railway Mobile Communication System) von verschiedensten Organisationen aus dem Telekom und dem Bahnbereich bspw. im MISTRAL Projekt (vgl. Abbildung 3) gemeinsam vorangetrieben (**3GPP** 2018). Ein im Juli 2018 von der ETSI durchgeführter Workshop (**ETSI** 2018) hat hier Ideen für eine sehr weitgehende Adaption und Integration gezeigt und es wird interessant sein zu sehen, wie weit bzw. wie tiefgreifend die Synergien eines künftigen Bahnnetzes hier tatsächlich gehen werden. Die Ausführungen des Vertreters der DB sind hier durchaus als sehr ambitioniert zu werten (**Marsch**, Patrick 2018) und zwar sowohl von der Vision Betriebsführung („relativen Bremswegabstand“) bis hin zum technischen Lösungsansatz („Future-proof architecture: Have a clear split between application and communication layer, and enable a future-proof, service-based Cloud architecture“). Am meisten überraschte hierbei der geplante Rollout in 2023, was bedeuten würde, dass die Bahnspezifische Telekomlösung – verglichen mit GSM-R – nur mit einem sehr geringen Nachlauf zur aktuellen Mobilfunk-Generation nutzbar wäre.

MISTRAL Projekt
Communication Systems for Next-Generation Railways
Shift2Rail 2016-2018 - www.mistral-s2r-project.eu/
Das Ziel des MISTRAL-Projekts ist es, angesichts des derzeit veralteten GSM-R die technische Spezifikation des zukünftigen Kommunikationssystems für den Eisenbahn-Sektor zu erarbeiten. Das neue Funksystem wird die Breitbandkapazität der IP-basierten drahtlosen Kommunikation nutzen, um innovative Dienste sowohl für Benutzer als auch für die Zugautomatisierung / -steuerung zu ermöglichen.

Abbildung 3: MISTRAL Projekt (Kurzbeschreibung)

Protokollebene

Wie schon erwähnt hat sich das IP (Internetprotokoll) als die zentrale Protokollfamilie für praktisch jede Art von Kommunikation herauskristallisiert. Im Folgenden sind zwei Ergänzungen ausgeführt.

In den letzten Jahren erfolgte unter dem Begriff Time-Sensitive Networking (TSN) bzw. IEEE 802.1 die Erweiterung der Internet Standards. Diese Erweiterungen betrafen hauptsächlich Funktionen, die die Übertragung mit sehr geringer Übertragungslatenz und hoher Verfügbarkeit sicherstellen. Die wesentlichen Schlüsselkomponenten von TSN sind (1) Zeitsynchronisation, zur Herstellung einer einheitlichen Zeitbasis, (2) Scheduling und Traffic Shaping, um verschiedene Verkehrsklassen sicher und verlässlich zu trennen, und (3) Auswahl von Kommunikationspfaden, um Reservierungen und Fehlertoleranz implementieren zu können. Die Anwendungsbereiche dieser Standards dürften sich dadurch erheblich erweitern. Perspektivische Anwendungsbereiche sind konvergente Netzwerke mit Echtzeit-Übertragung wie Audio/Video-Streams sowie insbesondere Realtime-Controls, im Automobil oder in der Automatisierung (Industrie 4.0).

Das Internet der Dinge (IoT) hat in den letzten Jahren als weiteres Segment des Internets stark an Bedeutung gewonnen, eine Vielzahl von Anwendungen entstehen lassen und etablierte Prozess- und Wertschöpfungsketten entweder obsolet gemacht oder zumindest mit neuen Qualitäten versehen und dadurch gravierend verändert. Die hierfür verwendeten Messaging Standards sind teilweise neue Definitionen; teilweise wurden schon bekannte Protokolle quasi zweckentfremdet. Die Nutzungshäufigkeiten der IoT Protokolle sind in der nachfolgenden Abbildung 4: Nutzungshäufigkeiten der IoT Protokolle ersichtlich.

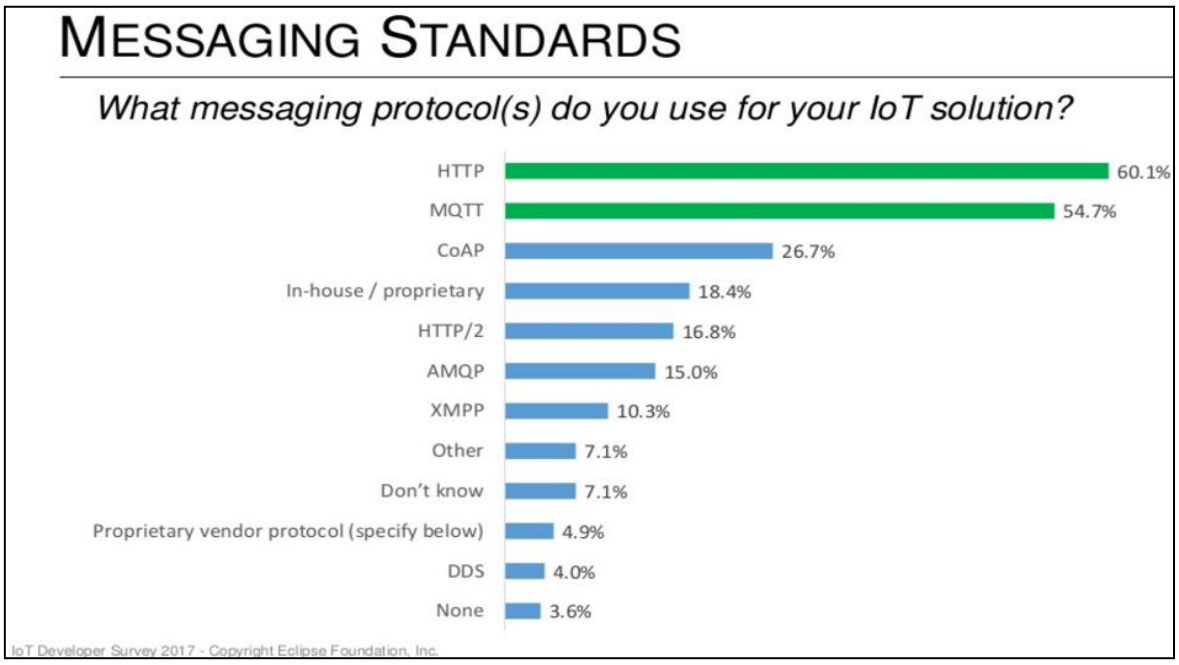

Abbildung 4: Nutzungshäufigkeiten der IoT Protokolle

Plattformebene

In der gegenwärtigen öffentlichen Wahrnehmung werden 5G-Infrastrukturen oftmals auf den Teilbereich der drahtlosen Zugangsnetze reduziert. Wie oben schon ausgeführt umfasst 5G jedoch auch die Konvergenz von Mobilfunk- und Breitbandnetzen und insbesondere eine softwarebasierte Gesamtnetzarchitektur. 5G-Technologien und Netzinfrastrukturen müssen daher in engem Zusammenhang und in ihren Wechselbeziehungen betrachtet werden. Es existiert sowohl eine große Schnittmenge auf Anwendungsseite als auch Übereinstimmung zwischen den Infrastrukturen, sowohl in Hinblick auf den Netzzugang als auch auf die intelligenten Netztechnologien. Intelligente Netzfunktionen bilden dabei die gemeinsame Basis zur effizienten Realisierung flexibler, spezialisierter Anwendungsnetze und unterstützen sowohl bestehende als auch völlig neuartige intelligente Ende-zu-Ende-Steuerungs- und Dienstplattformen.

Stichwörter hierzu sind auf der einen Seite Software Defined Network bzw. Edge-Computing zur Dezentralisierung und auf der anderen Seite Cloud-Computing zur Zentralisierung. Gesamtarchitekturen sind daher durch die Wahl der einen oder der anderen Technik unter Beibehaltung einer transparenten Schnittstelle besser an die spezifische Anwendung anpassbar. Kostengründe mögen eine Zentralisierung nahelegen, während Echtzeitanforderungen und Dienstgüte eine dezentrale und prozessnahe Ausführung verlangen.

Anwendungsebene

Die Standardisierungsdichte auf der Anwendungsebene ist insgesamt betrachtet eher rückläufig, da sich die Plattformebene (mit IP als Basiskommunikationsdienst) als Enabler für eine Vielzahl von Spezialdiensten bzw. der entsprechenden Ecosysteme herausgebildet hat. Man denke hierbei z.B. nur an den Bedeutungsverlust des SMS Dienstes durch diverse Messenger (WhatsApp,...), die durch das gleichzeitige Anbieten von Sprachdiensten eine analoge Entwicklung auch bei dieser Dienstgruppe in der Zukunft begünstigen.

Im Bahnbereich ist solch eine Entwicklung jedoch nicht zu beobachten. Ganz im Gegenteil wird die Standardisierung auch im Anwendungsbereich vorangetrieben, um hier überhaupt erst Strukturen aufzubauen, die das Gesamtsystem Bahn wirtschaftlicher machen. Als Beispiel verweisen wir die Standardisierungsbemühungen im Rahmen des euLyNX Projekts (siehe Abbildung 5: euLyNX Projekt (Kurzbeschreibung)).

euLyNX Projekt
https://www.eulynx.eu/
Ziel ist die Definition und Standardisierung von Schnittstellen im zukünftigen digitalen Zugsteuerungs-, Melde- und Automatisierungssystem mit dem Ziel, die Lebenszykluskosten von Signalanlagen deutlich zu senken. Das Steuerungs- und Automatisierungssystem bildet den Kern der digitalen Eisenbahn. Die Digitalisierung bringt große Vorteile für das Bahnsystem, wie z.B. eine kontinuierliche Überwachung für die zustandsorientierte Instandhaltung. Es gibt eine Reihe von Herausforderungen mit einem weit verteilten Sicherheitssystem, die aufgegriffen wurden und zu einer weiteren Harmonisierung der Zulassungsverfahren in der EU führen werden.

Abbildung 5: euLyNX Projekt (Kurzbeschreibung)

2.3.3 Methoden und Prozesse

Da die Domäne sehr stark von Standardisierung und Interoperabilität geprägt wird, sind viele Aktivitäten hiermit verbunden. So sind mit der Verabschiedung der Standards auch in vielen Fällen bereits Referenzimplementierungen erfolgreich getestet und die Referenztestsuiten- und Qualifizierungsumgebungen für die weitere Entwicklung sind verfügbar. Die im Rahmen von Standardisierungen durchzuführende Entwicklung neuer Protokolle wird durch umfangreiche Netzwerksimulationen unter Einbeziehungen von Emulation (Nutzung echter virtualisierter Komponenten) durchgeführt.

Es ist offensichtlich, dass solche Systeme eine sehr hohe Komplexität aufweisen und die Entwicklungs- und Testphasen daher mit vertretbarem Aufwand nur noch in virtualisierenden Umgebungen durchgeführt werden können. Alle Entwicklungs-, Test- und Qualitätssicherungsschritte sind dabei hochgradig automatisiert (Continuous Integration und Continuous Delivery). Es gibt eine Vielzahl unterschiedlicher Testziele und Testmethoden. Acceptance Tests und Usability Tests wird z.B. in sehr großem Masse auf Crowdtesting zurückgegriffen. Andere Testziele sind z.B. Last- und Performancetests

Im Wesentlichen werden die Stufen

- Entwickler-Test
- Eigenverantwortliche Tests
- Gesamt-Integrations-Test und
- Abnahmetests

unterschieden. Interoperabilität wird oft im Rahmen von Interop Events in konkreten Konfigurationen praktisch getestet. Für kritische Komponenten ist eine Zertifizierung durch verschiedene Einrichtungen (TüV, IPv6 Forum, ETSI, BSI, ...) erforderlich.

Auch hier sei wieder ein kurzer Bezug zum Bahnsektor eingefügt. Für das KISA System wurde eine Zertifizierung von DB Netz z.B. mit dem BSI (BSI-CC-PP-0085-2016) durchgeführt.

Da der Zertifizierungsprozess im Bahnbereich eine sehr wichtige Rolle spielt und zu erwarten ist, dass Technologien (und damit auch deren Zertifizierungsprozesse) aus dem Bereich Telekommunikation in

Zukunft auch im großen Maß im Bahnbereich adaptiert werden, soll am Beispiel von IPv6 dieser Zertifizierungsprozess im Folgenden sehr detailliert und nachvollziehbar beschrieben werden.

Das IPv6 Ready Logo Programm (siehe **IPv6** (2017, 2018)) ist ein vom IPv6 Forum initiiertes weltweites Zertifizierungsprogramm für Netzwerkgeräte hinsichtlich ihrer IPv6-Reife. Gesteuert wird das IPv6 Ready Logo Programm vom IPv6 Ready Logo Committee (v6LC). Dessen Aufgabe umfasst neben der Herausgabe von Testspezifikationen für Konformitäts- und Interoperabilitätstest, sowie der Bereitstellung von IPv6-Testing-Werkzeugen zur Testdurchführung, auch den Aufbau und die Zertifizierung von Testlaboren (Approved Labs).

Das IPv6 Ready Logo Programm besteht derzeit aus drei Phasen, die zeitlich und inhaltlich aufeinander aufbauen. Phase I lief von September 2003 bis November 2011. Phase II startete im Januar 2005 und löste Phase I im September 2011 ab. Phase I testet grundlegende IPv6-Funktionalitäten (Core Protocols) und die Interoperabilität von Netzwerkgeräten (d.h. Device under Test) untereinander, abgedeckt durch bis zu 170 Tests. Phase II dagegen testet darüber hinaus weitergehende Funktionalitäten, wie IPSec, DHCPv6, SNMP, NEMO und Customer Edge Router. Für das Testen der erweiterten IPv6 Funktionalität auf Basis von speziellen Protokollen ist ein erfolgreiches Testen der Core-Funktionalitäten die Voraussetzung. Für diese sieht die Phase-II bis zu 450 Tests vor, je nach Funktionsumfang des zu testenden Gerätes. Das v6LC empfiehlt ausdrücklich das Durchlaufen von Phase II. Ein Ablösen von Phase II durch Phase III ist bisher nicht terminiert, ein Testen der IPSec-Funktionalität wird für das Erlangen des Phase-III-Logos aber verpflichtend sein.

Das IPv6 Ready Logo kann auf zwei unterschiedliche Arten erlangt werden. Eine Möglichkeit besteht in der Einsendung des zu testenden Gerätes an eines der Approved Labs. Der Test durch ein Lab ist auf mehrere Wochen ausgelegt, hängt aber von vielen Faktoren ab, wie den gewählten Protokollen, Konfiguration und dem Debugging-Aufwand im Falle von Problemen. Dabei wird verstärkt auf die Virtualisierung der komplexen Testbed-Infrastrukturen gesetzt, die in einem Lab praktikabel nur durch die Nutzung entsprechender Virtualisierungsplattformen umsetzbar sind. Diese besteht aus mehreren Knoten verschiedener IPv6 Implementierungen, die im Zusammenspiel mit dem zu testenden Gerät verschiedene Szenarien durchgehen, und dabei die allgemeine Interoperabilität des Testobjektes prüfen. Zusätzlich beinhaltet sie auch entsprechende Testtreiberknoten und Monitoring Komponenten, die für die Ausführung und Protokollierung der Testläufe zuständig sind. Die Komplexität und die Steuerung dieser Infrastruktur sind durch Virtualisierungstechniken (z.B. Docker-Container oder VirtualBox) leicht in den Griff zu bekommen – dadurch kann sich der Tester/Testautomatisierer auf die wesentliche Materie des IPv6 Protokolltests fokussieren und die erforderlichen Test Evidence erzeugen.

Die zweite Möglichkeit ist die Testdurchführung mit Hilfe eines bereitgestellten Self-Test-Werkzeuges (für IPv6 vom InterOperability Laboratory, siehe **IPv6** (2017)) und das anschließende Einreichen der Testergebnisse (Test Evidence) bei einem der Approved Labs. In beiden Fällen muss das Testergebnis zu 100% PASSED sein, um als IPv6-Ready-zertifiziert zu werden. Auf beiden Wegen behält das IPv6 Ready Logo Committee die Hoheit über Inhalt und Format der benötigten Testprotokolle. Das Self-Test-System gibt Format und Inhalt des Test Evidences (Testprotokoll) vor. Dadurch ist von vornherein sichergestellt, dass diese den Anforderungen der zertifizierenden Stelle entsprechen. Zum aktuellen Zeitpunkt durchliefen bereits circa 2200 Geräte erfolgreich eine Phase des IPv6-Ready-Programms (**IPv6** Forum (2018)).

2.3.4 Ausbildung

Die Telekommunikation ist, von den hier betrachteten Domänen, diejenige welche am stärksten durch die Digitalisierung transformiert worden ist. Aus diesem Grund ist es auch etwas irreführend zu sagen, dass umfassende IT-Kenntnisse ein wesentlicher Bestandteil bei der Ausbildung in der Telekommunika-

tion seien. Genauer gesagt bilden Informations- und Kommunikationstechnik (IKT) heute in der Regel schon bei der Ausbildung eine Einheit.

2.4 Industrieautomatisierung

Industrieautomatisierung umspannt ein weites Feld, angefangen von der Steuerung einzelner Fertigungsvorgänge an einer Maschine über bis hin zur Softwarelösungen zur Ressourcenplanung eines Unternehmens bzw. einer Organisation (Enterprise Resource Planning – ERP). Diese Funktionen sind zunehmend über eine gemeinsame Datenbasis und durchgehende Kommunikationskanäle miteinander verbunden. Dadurch ist die Planung der Prozesse über sämtliche Unternehmensebenen hinweg möglich – egal ob es sich dabei um verschiedene Abteilungen oder verschiedene Werke handelt.

2.4.1 Produktstrukturen

Die klassische Automatisierungstechnik entwickelt sich weiter und wird heute maßgeblich durch den allgemeinen Trend zur umfassenden Digitalisierung mitbestimmt. Mit Hilfe moderner Informations- und Kommunikationstechnik soll die industrielle Produktion zukünftig eine weitestgehend selbstorganisierte Produktion ermöglichen. In der industriellen Produktion wird dieser Trend zunehmend mit dem Schlagwort „Industrie 4.0“ belegt. Verwandte Begriffe wie Machine-to-Machine Kommunikation (M2M) oder das Internet der Dinge fokussieren zwar auf etwas andere Themenbereiche, adressieren aber letztlich das gemeinsame Ziel, durch bessere Vernetzung, zunehmende Miniaturisierung und sinkende Hardwarekosten den Grundstein für sich selbst verwaltende Systeme zu legen.

Grundlage von Industrie 4.0 ist die (auch in anderen Bereichen stattfindende) zunehmende Vernetzung und die sich daraus ergebenden neuen Möglichkeiten der direkten Kommunikation zwischen Menschen, Maschinen und (zukünftig auch) Werkstücken. Der Informationsaustausch soll dabei nicht nur innerhalb eines Produktionsstandortes, sondern weltweit, über verschiedene Produktionsstandorte hinweg, stattfinden, egal ob innerhalb des Unternehmens oder mit den IT-Systemen von Zulieferern oder Kunden.

Industrie 4.0 wird das Potential zugesprochen, die Produktion individueller und effizienter zu gestalten. Die miteinander vernetzten Maschinen und Werkstücke tauschen – ähnlich wie Menschen in sozialen Netzwerken – Informationen direkt untereinander in Echtzeit aus. Produktionsanlagen sollen sich so selbstständig organisieren können und Abläufe und Termine untereinander koordinieren. Die Produktion wird flexibler, dynamischer und effizienter. Zudem kommunizieren die Maschinen direkt mit den IT-Systemen und Mitarbeitern des Unternehmens. Letztlich wird ein durchgängiger Informationsfluss über die gesamte Fertigungskette (Produktion, Vertrieb, Entwicklung usw.) angestrebt.

Kommunikation basiert auf Vernetzung. Die meisten Industriebetriebe nutzen für ihre Netzwerkinfrastruktur kabelgebundene Systeme, die zwar hohe Übertragungsgeschwindigkeiten und ein hohes Übertragungsvolumen ermöglichen, andererseits aber auch hohe Wartungskosten erzeugen und problematisch sind, wenn Produktionsstätten weit voneinander entfernt sind. In der Vergangenheit gab es zwei Faktoren, die eine Verbreitung kabelloser Systeme in der Industrie verhindert haben: Kosten und Sicherheit. Die Entwicklung von Technologien des Internet of Things (IoT) begünstigt zukünftig die Implementierung kabelloser Industriesysteme, da kabellose Sensoren, Aktoren und IoT-Geräte immer erschwinglicher werden. Problematisch bleibt das Thema IT-Sicherheit. Zwar haben zahlreiche Sicherheitsvorfälle eine stärkere Sensibilisierung für das Thema bewirkt, noch aber basieren die Abwehrstrategien häufig nur auf der Idee der physikalischen Abschottung, was im Gegensatz zum Trend der weltweiten Vernetzung steht. Industriesysteme müssen zukünftig stärker die bereits auf anderen Systemen (Computer, mobile Endgeräte) erfolgreich erprobten Best Practice Ansätze zur IT-Sicherheit umsetzen.

Mit der Einführung von IoT-Technologien werden in den Fertigungsanlagen riesige Datenmengen verfügbar. Über Cloudbasierte Softwarewerkzeuge können Hersteller damit in Echtzeit einen genauen Überblick über ihre Fertigungsanlagen erhalten und diese Erkenntnisse nutzen, um bessere Entscheidungen zu treffen und ihre Fertigungsverfahren zu verfeinern. Während in den letzten Jahren das Sammeln und Darstellen von Betriebs- und Anlagedaten im Vordergrund stand, wird es zukünftig zunehmend darum gehen, den ausgehobenen Datenschatz auch tatsächlich in effektiven Nutzen umzuwandeln. Die Datenanalyse wird sich als Entscheidungshilfe bei der Optimierung von Betriebs-, Instandhaltungs- und Geschäftsprozessen etablieren. Hierbei werden neue Technologien im Bereich Künstliche Intelligenz und Maschinelles Lernen in die Prozessleittechnik Einzug finden.

Nachdem in den vergangenen Jahren vor allem die Motorik im Vordergrund stand, wird die Entwicklungen im Bereich der künstlichen Intelligenz und des maschinellen Lernens zukünftig die Roboterentwicklung maßgeblich prägen. Sie werden eine neue Innovationsstufe einleiten; Roboter der neuen Generation werden ihr Verhalten selbstständig durch Lernen an veränderte Situationen anpassen, statt immer dieselben starren Bewegungsprogramme auszuführen.

2.4.2 Standards

Derzeit kommen zahlreiche unterschiedliche, miteinander nicht kompatible technische Lösungen zum Einsatz, wenn Maschinen miteinander kommunizieren. Um die Vision von Industrie 4.0 Wirklichkeit werden zu lassen, bedarf es einer Vereinheitlichung der Kommunikationsstrukturen. Mit OPC UA (Open Platform Communications Unified Architecture) soll der Datenaustausch zukünftig vereinheitlicht werden.

OPC UA ist eine Sammlung von IEC-Standards für die Kommunikation und den Datenaustausch im Umfeld der Industrieautomation, für die die OPC Foundation als global agierende Non-Profit-Organisation die Koordination und Weiterentwicklung übernimmt. Hierbei arbeitet sie eng mit Anwendern, Herstellern und Forschungseinrichtungen zusammenarbeitet (vergleiche **Litzel,** Nico (2018)). Mithilfe von OPC UA werden sowohl der Transport von Machine-to-Machine-Daten als auch Schnittstellen, Sicherheitsmechanismen und der semantische Aufbau der Daten spezifiziert. Die vollständige Architektur ist serviceorientiert aufgebaut.

Die erste Version der OPC Unified Architecture erschien Ende des Jahres 2006. Im Jahr 2009 wurde eine überarbeitete Version veröffentlicht. Es folgten viele weitere Spezifikationen in Standards wie EC 62541-11 (OPC Unified Architecture – Part 11), IEC 62541-12 (OPC Unified Architecture – Part 12) oder IEC 62541-13 (OPC Unified Architecture – Part 13).

Die serviceorientierte Architektur von OPC UA basiert auf mehreren Grundprinzipien. Diese sind:

- Bereitstellung einfacher Schnittstellen
- Bereitstellung eines einheitlichen Nachrichtenformats
- Bereitstellung flexibler Erweiterungsmöglichkeiten
- Implementierung hoher Sicherheitsstandards und verschiedener Sicherheitslevel

Um die Anzahl der Schnittstellen möglichst gering zu halten, erfolgt eine Beschreibung der Semantik innerhalb der Nachrichten. Aufgrund des einheitlichen Formats, der definierten Struktur und dem gemeinsamen Vokabular können alle Anwendungen im OPC-UA-Umfeld die Nachrichten verstehen. Die Beschränkung der Nachrichten auf definierte Formate reduziert die Komplexität So kann ein hohes Maß an Interoperabilität zwischen den Anwendungen sichergestellt werden. Besonders hervorzuheben ist in diesem Zusammenhang die Arbeiten zur Spezifikation von OPC UA TSN (*Time Sensitive Networking*), bei welcher die Kommunikation über Ethernet um Echtzeiteigenschaften erweitert wird. Diese Standardi-

sierung erfolgt in Rahmen des Ethernet-Standards IEEE 802.1. Damit soll es möglich werden, weiche Echtzeit-Kommunikationsanforderungen oberhalb der Maschinenebene, umzusetzen.

Ein weiteres Grundprinzip ist die IT-Sicherheit der Kommunikation. Die OPC-UA-Spezifikationen sehen Mechanismen und Methoden vor, hohe Sicherheitsstandards zu implementieren. Dabei ist es den Anwendungen überlassen, auf welches Sicherheitsniveau sie zurückgreifen möchten.

OPC UA beinhaltet Sicherheitsmechanismen wie Autorisierung, Authentifizierung und Verschlüsselung. Die Integrität von Daten lässt sich durch das Signieren mit beispielsweise digitalen X.509-Zertifikaten sicherstellen. Die Funktionen und Methoden orientieren sich an den Web-Service-Security-Spezifikationen und den Verschlüsselungsstandards der Public Key Infrastruktur (PKI). Die Architektur von OPC UA gestattet eine mehrschichtige Implementierung von Sicherheitsmechanismen. Damit basiert die Sicherheit in OPC UA zum Teil auf den bewährten und bereits bekannten Internet- und Service-Sicherheitsverfahren.

Vom Grundsatz her ist IT-Sicherheit zentraler Bestandteil der OPC-UA-Spezifikationen. OPC UA ist aber so flexibel gestaltet, dass jede Sicherheitsanforderung erfüllbar ist. Je nach Applikation, beteiligten Systemen und dem benötigten Sicherheitsniveau lassen sich verschiedene Sicherheitsstufen in OPC UA bereitstellen. Im Bedarfsfall können auch niedrigere Sicherheitsstufen für unkritische Anwendungen verwendet werden. Die Architektur bietet die Grundlage, für jede Anwendung den gewünschten Sicherheitslevel zu verwenden, ohne dass Applikationen neu aufgebaut werden müssen.

Zusammengefasst bietet OPC UA eine Vielzahl an Vorteilen gegenüber den Kommunikationstechnologien in der klassischen Automatisierungstechnik:

- es steht eine transparente und plattformneutrale Architektur für die industrielle Kommunikation zur Verfügung,
- die Standardisierung sorgt für ein hohes Maß an Interoperabilität zwischen verschiedenen Anwendungen und Herstellern,
- die Architektur ist flexibel, zukunftsfähig und erweiterbar,
- Anwendungen sind einfach zu konfigurieren und zu betreiben,
- einheitliche Schnittstellen ermöglichen den einfachen Zugriff auf Anwendungen und Daten,
- die komplette Architektur ist serviceorientiert und transparent,
- es lassen sich Anwendungen mit hoher Performance realisieren und
- die Sicherheit der Kommunikation ist gewährleistet.

2.4.3 Methoden und Prozesse

Wie in anderen Branchen auch setzen die Unternehmen zunehmend auf Agilität, um Entwicklungsprozesse nachhaltig auf den Kunden auszurichten. Methoden wie Scrum erobern dabei nicht nur den Bereich der Softwareentwicklung, sondern werden auf den gesamten Entwicklungsprozess vom Entwurf über die Produktion bis zur Logistik ausgeweitet.

Die Einführung agiler Methoden führt in der Regel dazu, dass die Entwicklungszeit verkürzt werden kann. Außerdem können mithilfe agiler Methoden veränderte Rahmenbedingungen noch während der Entwicklung besser berücksichtigt werden und in den Entwicklungsprozess einfließen. Dadurch werden Produkte erhalten, die den individuellen Anforderungen des Kunden bzw. Markts schon bei der Einführung gerecht werde. Methoden wie Scrum schaffen Transparenz und direkte Kundenbindung, indem sie die unmittelbare Kommunikation zwischen dem Scrum-Team, dem Kunden und weiteren Stakeholdern fördern. In Meetings mit dem Kunden wird offengelegt, was gerade entwickelt wird, so sieht der regel-

mäßig den Fortschritt der Arbeiten, kann diesen bewerten und gegebenenfalls neue Anforderungen einfließen lassen.

2.4.4 Ausbildung

Wir befinden uns in einer Phase, in der die klassische Automatisierungstechnik sich erheblich weiterentwickelt und immer stärker von der Digitalisierung beeinflusst wird. Software gewinnt dabei im Produktionsumfeld eine immer stärkere Bedeutung, sei es für die Produktions-Planung und -Überwachung oder das -Engineering. Allerdings wird es nur durch eine disziplinübergreifende Zusammenarbeit zwischen Design, Mechanik, Automatisierung und IT zukünftig noch möglich sein, weitere Produktivitätspotenziale zu heben.

Die Digitalisierung der Industrie mit Verbindung von virtueller und realer Welt verlangt dabei nicht nur 'Big Data', sondern 'Smart Data', also die intelligente Nutzung von Daten sowie Automatisierungs- und Prozess-Know-how. Dabei sind für den Ingenieur der Zukunft das technische Verständnis beider Welten und der spezifische Blick auf die Geschäftsmodelle der Industrie nötig. Mit wachsender Kommunikationsfähigkeit werden neue Services in der Cloud möglich, und bei zunehmender Vernetzung steigt auch der Bedarf an effektiven Schutzmechanismen wie Industrial Security.

Elektroniker für Automatisierungstechnik wirken an der Entwicklung und Erprobung hochentwickelter Steuerelektronik mit, richten also komplexe, rechnergesteuerte Industrieanlagen ein. Ihre Aufgabe ist dabei, dafür zu sorgen, dass die jeweiligen Einzelkomponenten ein automatisch arbeitendes Gesamtsystem bilden. Dabei können sie zwischen klassischen Speicherprogrammierbaren Steuerungen (SPS) und Lösungen basierend auf Industrie-PCs (IPC) wählen. Die klassische SPS unterstützen zwar Ausführung und Kommunikation in Echtzeit, bleiben aber der klassischen Regelungstechnik verhaftet. Ein (häufig Windows-basierter) IPC zeichnet sich dagegen durch hohe Geschwindigkeit und Leistungsfähigkeit aus, ist aber per se kein Echtzeit-System. Dafür ist er einfacher ihn in die Systemlandschaft von Industrie 4.0 bzw. IoT integrierbar. Dies bedeutet aber auch, dass IPCs (wie herkömmliche PCs) anfällig sind für Computerviren, Würmer und Trojaner, denn die herkömmlichen PC-Hacker-Werkzeuge funktionieren auch in der Produktion – mit unter Umständen gravierenden Folgen. SPS sind dagegen bisher vergleichsweise sicher, denn sie verfügen über einen wesentlich höheren Schutz vor unerlaubten Zugriffen von außen.

In der Ausbildung lernen Ingenieure für beide Systemarten Steuerungsprogramme zu erstellen, diese in Netzwerke einzubinden, Automatisierungsgeräte zu programmieren und Komponenten der Automatisierungstechnik zu konfigurieren. Für die SPS Programmierung hat sich hier ein genormter Standard etabliert (DIN EN 61131-3), welcher fünf gängige domänenspezifische Programmiersprachen spezifiziert. Diese orientieren sich vom Aufbau her an der Fachdomäne (Regelungstechnik) und sind damit auch für Fachkräfte ohne spezifischen IT-Hintergrund leicht erlernbar. Wegen ihrer Beschränkung auf den Kontext klassischer Regelungsaufgaben ist deren Bedeutung jedoch abnehmend. Stattdessen finden die Hochsprachen C, C++ und C# zunehmend Verbreitung, da sie auch die Programmierung von hochkomplexen Abläufen in der Automatisierung erlauben.

3 Vorschläge für den Eisenbahnsektor

In diesem Kapitel betrachten wir, unter Berücksichtigung der Erkenntnisse aus den vorangegangenen Kapiteln, Entwicklungen und Herausforderungen für die Software-Entwicklung im Eisenbahnsektor.
Die Bahntechnik hat mit den vorgenannten Domänen etliche Gemeinsamkeiten, weist aber auch einige Besonderheiten auf, die eine direkte Übernahme der erwähnten Technologien erschweren bzw. unmöglich machen. Ähnlich wie bei der zivilen Luftfahrt sind Züge und Infrastrukturanlagen langlebige Investitionsgüter, die hohe Sicherheitsanforderungen zu erfüllen haben. Ähnlich wie im Automobilbau basiert der kundensichtbare Mehrwert hauptsächlich auf Innovationen der IT-Systeme an Bord. Ähnlich wie Industrie 4.0 Anlagen sind bahntechnische Anlagen verstärkt Forderungen nach Flexibilisierung, Individualisierung, und Automatisierung bzw. Autonomie ausgesetzt. Künftige Bahn-Infrastrukturen sind in weit stärkerem Maße als heutige von Mobilfunk und IP-basierten Netzen abhängig. Allen genannten Branchen gemeinsam ist die zunehmende Komplexität der Steuerungen und speziell der Software, bedingt durch steigende Anforderungen und Möglichkeiten.
Besonderheiten der Bahn-Branche sind die lang etablierten Betriebs- und Zulassungsverfahren und ein „gewachsenes Ökosystem“ nationaler Betreiber und Behörden basierend auf historisch gewachsenen Staatsbahnsystemen, die nun liberalisiert werden.

3.1 Produktstrukturen

Informationstechnische Produkte im Bahnsektor sind streckenseitig Anlagen zur Zugbeeinflussung und kontinuierlichen Überwachung von Zügen und Fahrstraßen, zugseitig Steuergeräte zur Einstellung und Kontrolle des Fahrtzustands und des Fahrzeugzustands. Anlagen für das Infotainment im Zug, Fahrgastinformation, Fahrplanerstellung, Fahrtenplanung und Fahrkostenabrechnung sollen hier nicht weiter berücksichtigt werden. (Ein anhaltender Trend ist in diesem Bereich sicherlich, mehr und mehr Funktionalität in Web-Dienste und somit auf Endkundengeräte (Mobiltelefone, Tablets, Laptops) zu verlagern.)

Diese Trends und Aufgaben betreffen sowohl die Entwicklung der fahrzeugseitigen Software als auch die Softwaresysteme der streckenseitigen Ausrüstung:

- Integration verschiedener Geräte und Funktionalitäten
- Elektronische Überwachung von Fahrzeugparametern
- Fahrerassistenzsysteme, autonome Fahrfunktionen
- Variantenmanagement, globale Produktlinien
- Commercially-off-the-shelf (COTS) Komponenten
- Standardisierte Steuergerätearchitektur und Middleware
- Obsoleszenzmanagement

Im Bereich der streckenseitigen Ausrüstung bestehen zusätzlich folgende Herausforderungen:

- Ersatz analoger und End-to-end Verbindungen durch all-IP-Netze
- Ersatz von GSM-R durch LTE-R/FRMCS
- Automatische Regelung des Bahnverkehrs

Nachfolgend gehen wir auf diese Punkte kurz ein.

Integration verschiedener Geräte und Funktionalitäten: Zur Verringerung von Kosten und Raumbedarf ist es sinnvoll, mehrere bisher von verschiedenen Steuergeräten wahrgenommene Funktionalitäten in einem Gerät zu integrieren. Zur Integration der ETCS On-Board Unit (OBU) und des Train Control and

Monitoring Systems (TCMS) siehe weiter unten Kapitel 4.1.2. Während dies ein sehr spezielles Beispiel ist, da Systeme unterschiedlicher Kritikalität integriert werden müssen, gibt es eine Vielzahl weiterer Funktionen, die in gemeinsamen Geräten untergebracht werden können: Heizung und Klima, Ventilation, Beleuchtung, Innentüren, Sitze, WC-Anlagen sowie Antrieb, Stromregelung, Pantograph, Energieversorgung, Bremsen, Hydraulik / Pneumatik, Fahrtenschreiber usw. Letztlich ist hier eine Beschränkung auf wenige Steuergeräte sinnvoll, die aber mit „intelligenter" Sensorik und Aktorik kommunizieren. Abhängig von der Art der Integration (siehe unten in Kapitel 4.1.2) ergeben sich unterschiedliche Anforderungen an die Validierung und Zertifizierung.

Elektronische Überwachung von Fahrzeugparametern: Großes Potential zur Optimierung des Betriebs liegt in der Möglichkeit, viele unterschiedliche Fahrzeugparameter mit preisgünstigen Sensoren (z.B. Rad- oder Drehgestell-Sensoren) zu überwachen und die so entstehenden großen Datenmengen zu analysieren. Künftige Produktstrukturen werden also – ähnlich wie im Paradigma „Internet der Dinge" aus intelligenten Sensoren plus einer dahinterliegenden Big-Data-Auswertungsebene bestehen. Konditionale Wartung bezeichnet die Möglichkeit, an Hand von Messwerten frühzeitig zu erkennen, dass ein bestimmtes Bauteil getauscht werden muss. Dadurch entfällt die Notwendigkeit, diese Teile „vorsorglich" bei der Routine-Wartung zu ersetzen. Unter prädiktiver Wartung versteht man den Einsatz von Methoden der künstlichen Intelligenz, um aus der Kombination von Messreihen die Restlebensdauer bestimmter Bauteile vorherzusagen. Zur Kommunikation der Sensoren mit den entsprechenden Auswertungseinheiten werden vornehmlich drahtlose Verbindungen (Bluetooth Low Energy, Zigbee, etc.) verwendet werden, was „Energy Harvesting" Methoden zur Energieversorgung der Sensoren erforderlich macht. Möglichkeiten zur batterielosen Stromversorgung bestehen z.B. in der Umwandlung von Bewegungsenergie (kinetische Energiewandler), Licht (Photovoltaik-Zellen), und Temperatur (Peltier-Elemente). Hier steckt noch ein hohes Potential für Forschung und industrienahe Entwicklung für die Software. Zur Auswertung sind vor allem auch Methoden des maschinellen Lernens einsetzbar. Da die resultierenden Prognosen immer nur stochastischer Art sind, ergibt sich einerseits die Frage nach der Validität der Aussagen, andererseits die Frage, wie die Wahrscheinlichkeiten zur Zulassung zu bewerten sind.

Fahrerassistenzsysteme, autonome Fahrfunktionen: Dies ist ein aktuell beherrschendes Thema für die Software-Entwicklung, nicht nur im Automobilbereich. Die zahlreichen Aufgaben eines Lokführers werden nach und nach durch automatisierte oder ferngesteuerte Software-Systeme ersetzt. Die UITP (Union internationale des transports publics) unterscheidet in IEC 62267:2009 „Railway applications – Automated Urban Guided Transport – Safety Requirements" fünf Automatisierungsgrade (GoA – Grades of Automation), die auch in die Norm IEC 62290-1:2014 (Railway applications – Urban guided transport management and command/control systems – Part 1: System principles and fundamental concepts) aufgenommen wurden:

- GoA 0: Fahrzeugführer fährt auf Sicht
- GoA 1: Fahrzeugführer beschleunigt und bremst, kontrolliert die Türsteuerung und bewältigt Notfallsituationen oder ungeplante Umleitungen (manueller Betrieb, NTO)
- GoA 2: automatisierte Beschleunigungs- und Bremsvorgänge, Fahrzeugführer rüstet den Zug auf und ab, kontrolliert die Türsteuerung, kann bei Bedarf den Zug selbst steuern und bewältigt Notfallsituationen (halbautomatischer Betrieb, semi-automatic train operation, STO)
- GoA 3: automatisierter Zugbetrieb, der Zugbegleiter kontrolliert die Türsteuerung und steuert den Zug in Notfallsituationen (driverless train operation, DTO)
- GoA 4: automatisierter Zugbetrieb, automatisierte Türsteuerung und automatisierte Bewältigung von Notfällen. Im Zug befindet sich keinerlei Personal (unattended train operation, UTO)

Zugbeeinflussungssysteme (PZB, LZB) sind seit den 1970-er Jahren Stand der Technik, STO und DTO wird vor allem im U-Bahn-Bereich eingesetzt. Fahrerlose People-Mover und U-Bahnen gibt es seit Beginn der 2000-er Jahre. Da sich durch die Automatisierung die Zuverlässigkeit und Kapazität des Be-

triebs steigern lässt (Pünktlichkeit, Varianz, Energieverbrauch), ist mit einem Einsatz im Nahverkehr (S-Bahnen) in absehbarer Zeit zu rechnen. Migrationsschritte zur Automatisierung des Fahrens werden in dem Tagungsband **TU Darmstadt** (2017) erörtert. Der Einsatz im Fernverkehrs- und Hochgeschwindigkeitsbereich erfordert jedoch größere Infrastrukturmaßnahmen (Gleisschutzmaßnahmen, Trennwände); wir sehen ihn daher nicht vor 2030. Allerdings wurde im Bereich der SBB (Schweizerischen Bundesbahn) im Projekt Smart Rail 4.0 im Dezember 2017 ein auf ETCS L2 aufbauendes „Assistenzsystem für Lokführer" getestet, welches auf der Strecke zwischen Bern und Olten einen Stadler Doppelstockzug automatisch bremsen und bis auf 160 km/h energieeffizient und sicher beschleunigen konnte

Variantenmanagement, globale Produktlinien: Ein großes Problem für die Hersteller ist die Vielfalt länderspezifischer Regeln und Kundenwünsche, die zu einer immer größeren Diversifizierung des Produktangebots führt. Dieses Problem wird verstärkt durch den Trend zu immer größeren Zusammenschlüssen der Herstellerunternehmen, bei der immer weniger Konzerne immer größere Kundenkreise zu bedienen haben. Eine Möglichkeit, dieses Problem in den Griff zu bekommen, besteht in der Entwicklung generischer Komponenten, die nach dem Baukastenprinzip zusammengesetzt werden können. Ungeklärt ist jedoch weitgehend, wie mit der sogenannten Feature-Interaction-Problematik umzugehen ist: die gegenseitigen Abhängigkeiten verschiedener Software-Merkmale können zu unvorhersehbaren Auswirkungen führen.

Commercial-off-the-shelf (COTS) Hardware: Viele Steuergeräte der Bahntechnik sind maßgefertigte Speziallösungen für kleine und kleinste Serien. Dies ist sehr kostenintensiv; wünschenswert wäre eine stärkere Verwendung von Standardkomponenten, die in großen Stückzahlen auch für andere Domänen gefertigt werden. Ein Problem bei der Verwendung von COTS Hardware ist, dass die internen Strukturen für den Anwender unbekannt sind (Black-Box) und sich daher aus Anwendersicht nichtdeterministisch verhalten können. Dies erschwert den Test und die Validierung. Andererseits gibt es bereits seit 2015 SPS-Stellwerke auf Basis industrieller COTS-Hardware (**S**peicher-**P**rogrammierbarer **S**teuerungen), welche die Lifecycle-Kosten senken können. Im BMWi-Projekt NeGSt (2011-2013) wurde die Projektierung von COTS-Produkten für die Leit- und Sicherungstechnik untersucht (**NeGST** 2013). Als Problem wurde dabei die domänenübergreifende Zulassung und Übertragung der Zulassung anderer Domänen (z.B. Straßenfahrzeuge oder chemische Prozessindustrie) identifiziert.

Standardisierte Steuergerätearchitektur und Middleware: Im Aerospace-Bereich ist mit der IMA Architektur (siehe Abschnitt 2.2.2) eine einheitliche Steuergeräte-Plattform geschaffen worden, auf die sämtliche Anwendungen aufbauen können. In der Bahntechnik gibt es zwar herstellerspezifische Steuergeräte (z.B. bei Bombardier Transportation), aber keinen Industriestandard. Durch die zunehmende Öffnung der Märkte wird dieses Thema jedoch in Zukunft auch für die Bahntechnik relevant.
Die Automobilindustrie hat mit dem AUTOSAR Vorhaben (siehe Abschnitt 2.1.3) auch eine einheitliche Middleware für automobile Steuergerätesoftware geschaffen. Ähnlich wurde in der Industrieautomatisierung mit OPC-UA eine einheitliche Kommunikationsschicht für die Maschine-zu-Maschine-Kommunikation definiert. Diese Middleware-Schichten sind sicherlich nicht ohne weiteres auf bahntechnische Geräte übertragbar. Die zugrundeliegenden Ideen (Hardware-Abstraktion, Konfigurierbarkeit, transparente Kommunikation) wären jedoch auch für die Bahntechnik relevant. Im Safe4Rail Projekt (siehe Abbildung 7: SAFE4RAIL Projekt (Kurzbeschreibung)) wurde ein „TCMS functional distribution framework" definiert und in Form von drei „design instantiations" auf die PikeOS, AUTOSAR, und Integrity Plattform instantiiert. Obwohl unseres Wissens nach derzeit keine herstellerübergreifenden Initiativen zur Definition eines einheitlichen „EISENSAR" existieren, ist zu erwarten, dass die Rolle der Middleware in bahntechnischen Steuergeräten immer weiter zunimmt.

Obsoleszensmanagement: Im Allgemeinen haben Züge eine längere Lebenszeit als die eingebauten Steuerungen. Elektronische Bauteile und Komponenten werden jedoch oftmals bereits nach wenigen Jahren abgekündigt, die Software ist nicht mehr ohne weiteres auf neueren Plattformen ablauffähig. Daraus ergibt sich das Problem der Bauartzulassung nachträglich veränderter Hard- und Softwaresys-

teme. Dieses Problem besteht auch in vergleichbaren Industriezweigen wie z.B. der Luftfahrt-Branche. Dort versucht man, das Problem mit LRU (Line-Replaceable Units) und IMA (integrierter modularer Avionik) zumindest abzumildern, siehe Abschnitt 2.2.2. In anderen Branchen, z.B. bei Aufzügen, wird im Zuge einer „Modernisierung" in einem so genannten „Refurbishment" sämtliche elektronischen Steuerungselemente ausgetauscht, mechanische Elemente wie Kabine, Türen und Motoren bleiben erhalten.

Ersatz analoger und End-to-end Verbindungen durch all-IP-Netze: Was in der Telekommunikation schon längst Realität ist, wird in der Bahntechnik erst langsam erprobt. Weichen, Signale, Balisen, Achszähler usw. werden künftig aus informationstechnischer Sicht nur noch Knoten in einem „Internet der Bahn-Dinge" sein. Hier müssen dezentrale Stellteile als Adaptoren geschaffen werden, die es erlauben, herkömmliche Technik in diese Welt zu integrieren.

Ersatz von GSM-R durch LTE-R/FRMCS: Das heutige GSM-R System, welches seit den 1990-er Jahren in Betrieb ist, ist am Ende seiner Lebensdauer angelangt. GSM wird als zweite Mobilfunkgeneration bezeichnet, die dritte Generation ist UMTS. GSM nutzt die vorhandenen knappen Frequenzbänder vergleichsweise ineffizient und wird von den Mobilfunkbetreibern nur noch wenige Jahre unterstützt. Zurzeit laufen Entwicklungen zur Adaption des aktuellen Mobilfunkstandards LTE (vierte Generation) für den Bahnbereich. Jedoch ist zu bemerken, dass im Mobilfunkbereich bereits die fünfte Generation vor der Einführung steht. Hier sollte überlegt werden, wie der Anschluss an solche neueren Entwicklungen kontinuierlich gewährleistet werden kann. Vergleiche hierzu auch die Ausführungen in Abschnitt 2.3.

Automatische Regelung des Bahnverkehrs: Die Hauptaufgabe des Personals in Stellwerken ist es den Betrieb zu überwachen und Ausnahmesituationen zu behandeln. In wie weit diese Aufgaben, zum Beispiel mit Methoden der künstlichen Intelligenz, weiter automatisiert werden können, soll hier nicht weiter erörtert werden. Je mehr die Steuerungs- und Überwachungsaufgaben durch Software realisiert werden, stellt sich aber die grundsätzliche Frage nach der Verteilung der Verantwortung zwischen Zug und Betriebszentrale. Hier gibt es zwei gegenläufige Tendenzen:

a) das Zentrale trägt die Verantwortung für den Fahrplan und -betrieb, die Züge sind „nur" unselbständige Befehlsempfänger und werden vollständig ferngesteuert.
b) jeder Zug trägt die vollständige Verantwortung für seine Fahrt, die Zentrale ist „nur" für Ausnahmebehandlung zuständig. Weichen und Signale werden kollaborativ von den Zügen gestellt (ähnlich wie bei Straßenbahnen).

Für Variante b) ist neben einer sicheren und genauen Ortung (siehe **Düpmeier** (2017)) eine sichere Zug-zu-Zug-Kommunikation erforderlich (vergleiche auch die Car-to-Car Initiativen der Automobilindustrie). Dies ist ein Thema, was bereits für andere Domänen von der Informatik untersucht wird. In der Indoor-Transportlogistik gibt es bereits Systeme, bei denen sich autonome fahrerlose Transportfahrzeuge untereinander abstimmen, wer welchen Transportauftrag übernimmt (siehe **Schlingloff** (2017)). Übertragen auf den Schienenverkehr würde ein System fahrerloser autonomer Schienenfahrzeuge einige Vorzüge gegenüber den heutigen Betriebsmodellen bieten:

- Flexible, bedarfsgerechte, dynamische Fahrpläne
- Bessere Ausnutzung des Schienennetzes durch Güterverkehr zu Randzeiten
- Vermeidung von Engpässen durch automatische Wahl von Alternativen
- „Individualisierter Massenverkehr", „Call a Train"

Ein erster Schritt in diese Richtung wäre es, Kolonnenfahrten zu ermöglichen, d.h., dass sich Züge dynamisch zu virtuellen Consists koppeln. Die Fahrzeuge fahren dann mit einem Abstand unterhalb des Notbremsabstands. Für den Automobilbereich gibt es diesbezügliche Initiativen, z.B. die European Truck Platooning Challenge **EUTPC** (2016), das EU-Projekt **SARTRE** (2012) oder das BMBF-Projekt **CrESt** (2017). Vorteile von Platooning sind

- Erhebliche Reduktion des Luftwiderstands
- Entlastung des folgenden Triebfahrzeugführers bzw. führerloses Fahren
- Bessere Ausnutzung des Schienennetzes durch kleinstmögliche Abstände

Erfahrungen aus der „European Truck Platooning Challenge 2016“ sind unter anderem:

- Unterschiedliche nationale und regionale Zulassungsverfahren und Zuständigkeiten erschweren den grenzüberschreitenden Platooning-Verkehr.
- Von den Zulassungsbehörden wurden verschiedene Risiken sehr unterschiedlich bewertet und verschiedene Risikominderungsmaßnahmen gefordert: Kennzeichnung des Platoons, Entkopplung in bestimmten Situationen, Geschwindigkeitseinschränkungen, gleichmäßige Lastverteilung usw.

Interessanterweise wurden keinerlei Probleme mit der Software berichtet, so dass Kolonnenfahrten aus softwaretechnischer Sicht auch für den schienengebundenen Verkehr realistisch machbar sind.

3.2 Standards

Im Gegensatz zu den Domänen Automotive und Industrieautomatisierung sind die Domänen Schienenverkehr und Luftfahrt stark reglementiert. Im Vergleich zur Luftfahrt gibt es bei der Bahntechnik aus historischen Gründen viele länderspezifische Regelungen. Allein in Europa gibt es über 20 verschiedene Zugbeeinflussungssysteme. Die Vielzahl unterschiedlicher Standards und Vorschriften haben zu Interoperabilitätsproblemen geführt, die sich mehr und mehr als Innovationshemmnis herausstellen. Aus diesem Grund hat auf dem Gebiet der Leit- und Sicherungstechnik die Europäische Union Anfang der 1990-er Jahre mit der Definition von ETCS als einheitlichem **Produkt-Standard** begonnen.

ETCS soll die Investitionskosten für Betreiber senken, die Zulassung vereinfachen und Umrüstzeiten vermeiden. Auf Grund unterschiedlicher Interpretationen der Spezifikation war die Interoperabilität von Geräten verschiedener Hersteller jedoch nicht immer gegeben. Ein Versuch die Ungenauigkeiten von ETCS durch die Entwicklung einer maßgeblichen und unter einer freien Softwarelizenz verfügbaren Implementierung zu beseitigen, wurde im *openETCS* Projekt unternommen. An diesem Projekt, das von 2012–2015 lief waren neben der Deutschen Bahn auch führende europäische Bahnhersteller und Forschungseinrichtungen beteiligt. Das openETCS Projekt kann als ein Beispiel für den Einsatz von Referenzarchitekturen, beziehungsweise Referenzimplementierungen, gesehen werden.

In vielen Domänen weisen die **Prozess-Normen** für die funktionale Sicherheit große Ähnlichkeit zur allgemeinen Sicherheitsnorm IEC 61508 auf. Aus der Reihe fällt hier eigentlich nur die Norm DO-178 für Avionik-Software. Die Spezialisierung der Normen gemäß den verschiedenen Domänen ist zurzeit eine gegebene Tatsache. Zwingend notwendig ist sie aber nicht, denn die meisten Aspekte sicherer Softwareentwicklung beziehen sich auf Programmiersprachen und Modellierungskonzepte, die domänenübergreifend anwendbar sind. Die Zersplitterung der Normenlandschaft führt dazu, dass neue Herausforderungen und Erkenntnisse in jeder domänenspezifischen Norm separat aufgearbeitet werden müssen. Konkret gilt das zum Beispiel für die Behandlung von Fragen der IT-Sicherheit. Um die Themen Funktionale Sicherheit und IT-Sicherheit in Zukunft auf einander abgestimmt zu behandeln, sollte man überlegen, ein *IT-Sicherheits-Supplement* zur EN 50128 zu definieren. Eine Möglichkeit der weiteren Fragmentierung der Normenlandschaft entgegenzuwirken, besteht darin, dass sich die Bahnbranche wieder enger an die allgemeine Sicherheitsnorm IEC 61508 anlehnt. Mittelfristig könnte man überlegen, dass in Hinblick auf Softwareentwicklung die IEC 61508-3 die maßgebliche Norm wird. Wo aus zwingenden Gründen bahnspezifische Regelungen nötig wären, sollte man diese als fokussierte Supplemente zur IEC 61508-3 formulieren.

Eine weitere Frage ist, ob es sich lohnt, die Unterscheidung in fünf Softwaresicherheitsanforderungsstufen (SSAS 0-4) aufrecht zu erhalten. Da die EN 50128, im Gegensatz zu anderen Normen, bei den empfohlenen Maßnahmen weder zwischen SSAS 1 und 2 noch zwischen SSAS 3 und SSAS 4 unterscheidet, könnte man sich inhaltlich auf die drei Stufen 0, 2 und 4 beschränken.

Erforderlich ist in jedem Fall eine Aktualisierung der Empfehlungen für modellbasierte Entwicklung und formale Methoden auf den in den letzten 20 Jahren gewachsenen Stand der Technik.

3.3 Methoden und Prozesse

Ein genereller Trend aller betrachteten Industriezweige seit den 1980-er Jahren ist, dass immer mehr Funktionen in Software realisiert werden. Gründe dafür liegen in der Flexibilität, Änder- und Erweiterbarkeit, und quasi kostenlosen Vervielfältigbarkeit. Dies führt zu einer stetig zunehmenden Komplexität der Softwaresysteme. Als immaterielles Gut altert Software im Prinzip nicht; jedoch sind durch Änderungen der Hard- und Softwareumgebung Aktualisierungen meist unumgänglich. Die Methoden und Prozesse für den Entwurf und die Qualitätssicherung der immer komplexer werdenden Softwaresysteme im Bahnbereich müssen stärker darauf abzielen, deren *langfristige Wartung* und *kurzfristige Anpassbarkeit* sicherzustellen. Unter langfristiger Wartung verstehen wir ausdrücklich die Fähigkeit, Soft- und Hardware über Jahrzehnte hinweg kontinuierlich weiter zu entwickeln.

Für die Sicherstellung sowohl der langfristigen Wartbarkeit als auch der kurzfristigen Anpassbarkeit bedarf es der konsequenten Umsetzung von Konzepten serviceorientierter Architekturen sowie der Virtualisierung von Hardware und ganzen Kommunikationsnetzen. Eine große Herausforderung wird es dabei sein, auch bei diesen, deutlich dynamischeren Ansätzen, höchste Qualitäts- und Sicherheitsansprüche sicherzustellen. Hier bedarf es verstärkter Forschungsarbeiten auf dem Gebiet dynamischer und statischer Softwareanalysen, die selbstverständlich nicht unabhängig von verwandten Domänen durchgeführt werden sollten. Generell lässt sich sagen, dass die in der Bahntechnik angewendete Methoden und Prozesse zur Software-Entwicklung sich nicht grundlegend von denen der anderen der hier betrachteten Domänen unterscheiden. Auch die aktuellen Trends und Herausforderungen sind durchaus ähnlich. **Stecklina** und Passeck (2015) betonen: „Neben den klassischen Software-Entwicklungsmethoden spielen seit längerer Zeit Vorgehensweisen wie Objektorientiertes Entwickeln, Agile Software-Entwicklung und Modellbasierte Entwicklung auch in der sicherheitsgerichteten Software-Entwicklung eine Rolle“. Genc (2014) nennt modellbasierte Entwicklung, agile Prozesse und COTS-Produkte als aktuelle Trends der Bahntechnik. Wir gehen nachfolgend speziell auf die Themen modellbasierte Entwicklung, agile Entwicklungsprozesse und IT-Sicherheit ein.

3.3.1 Modellbasierte Entwicklung

Die wesentliche Idee der modellbasierten Entwicklung besteht darin, die Erfassung, Bewertung und Generierung von Informationen im Entwicklungsprozess zu systematisieren. Ein Entwicklungsprozess umfasst stets informelle Aspekte – die Erwartungen der Nutzer oder organisatorische Randbedingungen, und formale Aspekte – den programmiersprachlichen Code, den Binär-Code und mehr. Modelle sollen dazu dienen, die Kluft zwischen natürlicher Sprache einerseits und formaler Sprachen andererseits zu überbrücken. Natürliche Sprachen sind angemessen für die Formulierung der informellen Schritte und Aspekte; sie sind ausdrucksstark, flexibel und werden von allen Beteiligten beherrscht. Formale Sprachen sind präzise und maschinenlesbar. Keine der beiden sprachlichen Ebenen kann in einem Entwicklungsprozess vermieden werden. Vor allem in den frühen Phasen eines Prozesses sind Programmiersprachen nicht geeignet, um den Entwicklungsstand festzuhalten. Da sich Kundenanforderungen, Benutzungsszenarien, Software-Architektur und Hardware-Deployment programmiersprachlich

nicht gut erfassen lassen, werden oftmals Beschreibungen in natürlicher Sprache oder bildliche Darstellungen verwendet. Diese bieten allerdings wenig Unterstützung für eine systematische Analyse und Bewertung des Entwicklungsstands. Ebenso wird die Verbindung mit den formalsprachlichen Artefakten – Programme, Test, Konfigurationsdateien, etc. – nur wenig unterstützt und ist nicht automatisierbar.

Um diese Lücke zu schließen, können formale oder semiformale Modelle verwendet werden. (Anm.: In einem formalen Modell sind Syntax und Semantik eindeutig festgelegt, Beispiel: endlicher Automat und reguläre Sprache; bei einem semiformalen Modell ist nur die Syntax festgelegt, Beispiel: SysML Requirements-Diagramm). Die Modelle beschreiben noch keine fertigen Lösungen, sondern stellen einen Rahmen zur Verfügung, bestimmte Fragen systematisch zu untersuchen. Die wesentlichen Bestandteile eines Systems können zum Beispiel in einem Architekturdiagramm dargestellt werden. Ein solches Diagramm zu erstellen bedeutet, die Komponenten zu benennen, deren hierarchische Struktur zu definieren und die Kommunikation zwischen den Komponenten festzulegen. Das Diagramm bleibt durch die hierarchische Struktur überschaubar; gleichzeitig lässt es formale Prüfungen zu:

- Sind alle Komponenten verbunden?
- Kann der Daten- und Informationsfluss gemäß der gewünschten Nutzung abgebildet werden?
- Sind alle Schnittstellen implementiert worden?
- Entspricht die Kommunikation zwischen den Programmmodulen den in der Architektur erfassten Regeln?

Ein Architekturdiagramm kann in der weiteren Entwicklung genutzt werden, um Module und Klassen zu definieren und deren Konsistenz zu überprüfen. Ein wichtiger Aspekt dabei ist, dass das Modell nicht nur zu Dokumentationszwecken verwendet wird, sondern als „first class citizen“ ständig in den Prozess eingebunden ist. Bei der modellbasierten Software-Entwicklung wird frühzeitig im Prozess ein abstraktes Systemmodell erstellt, welches im Entwicklungsprozess schrittweise zu ausführbarem Code verfeinert wird.

Universelle versus domänenspezifische Sprachen

Die Ansätze, Methoden und Techniken der modellbasierten Entwicklung können in zwei Hauptströmungen unterteil werden:

- Der universelle Ansatz, maßgeblich vertreten durch Sprachen wie UML (Unified Modeling Language) und SysML (Systems Modeling Language).
- Der domänenspezifische Ansatz, vertreten durch die Technik der Domain Specific Languages (DSL).

Universeller Ansatz (UML/SysML): Im universellen Ansatz wird versucht, allen Anforderungen durch eine universelle, einheitliche Sprache gerecht zu werden. UML und SysML sind von der OMG (Object Management Group) standardisierte Sprachen, die aus objekt- und komponentenorientierten Ansätzen abgeleitet sind. Der wesentliche Vorteil besteht darin, dass UML- und SysML-Modelle mit marktgängigen Werkzeugen erstellt und über Unternehmensgrenzen hinweg ausgetauscht werden können. UML und SysML sind aber nur Sprachen, keine Methoden. D.h. es lässt sich zwar dank der Universalität so gut wie alles beschreiben, es gibt aber keine Vorgaben dafür, wie das geschehen soll, was genau die Modelle bedeuten oder wie sie verwendet werden sollen.

Domänenspezifische Sprachen: Der domänenspezifische Ansatz ist geleitet von der Idee, dass jede Branche individuelle Anforderungen und Entwicklungsprozesse hat und die Modellierungsmittel diesen Umständen gerecht werden müssen. Ganz im Gegensatz zu den universellen Sprachen zielt der Ansatz der Domain Specific Languages (DSL) darauf ab, eine technische Infrastruktur zur Verfügung zu stellen, mit der domänenspezifische Sprachen schnell entwickelt und im Entwicklungsprozess nutzbar gemacht werden können. Diese Technik steht mittlerweile in guter Qualität zur Verfügung. Es gibt Werkzeuge,

mit denen Sprachen definiert und daraus leistungsstarke Editoren sowie Code- und Dokumentengeneratoren generiert werden können. So kann zum Beispiel eine Sprache definiert werden, in der die Nutzung eines Systems genau so beschrieben werden kann, wie es der Prozess erfordert. Aus der Sprachdefinition wird ein Editor generiert, der die Eingaben auf syntaktische und semantische Korrektheit prüft. Damit wird bereits früh im Prozess eine systematische, formale, automatisierbare Qualitätskontrolle eingeführt. Da die Sprache prozessspezifisch definiert wird, können die Betroffenen an der Gestaltung der Sprache mitwirken und sicherstellen, dass ein adäquates Detaillierungs- und Formalisierungsniveau erreicht wird. Aus den Modellen, die mit dem Editor erstellt wurden, können weitere Artefakte generiert werden, z.B. Dokumente für die Zertifizierung oder Code- bzw. Codefragmente für die weitere Entwicklung. Auch die Entwicklung dieser Generatoren wird durch die DSL-Technik wirksam unterstützt.

Nachfolgend seien drei gängige Werkzeugumgebungen genannt.

- Die besonders in der Automobilbranche weit verbreitete Modellierungssprache Matlab/Simulink/Stateflow™ von The Mathworks© verfolgt ein hybrides Konzept, bei der eine universelle Datenfluss-Sprache durch die Einbindung spezieller Bibliotheken („Toolboxes“) an eine bestimmte Domäne angepasst werden kann.
- Die Modellierungssprache Scade von Ansys basiert auf der formalen Sprache Lustre und ist deshalb besonders für sicherheitskritische Anwendungen einsetzbar. Die Scade Suite integriert SysML- und Lifecycle-Management-Werkzeuge mit formalen Verifikationswerkzeugen und zertifizierten Codegeneratoren.
- Die Modellierungsumgebung Ascet von ETAS enthält neben Blockdiagrammen für den Signalfluss und Zustandsautomaten für den Kontrollfluss eine Java-ähnliche Sprache zur Modellierung physikalischer Zusammenhänge und wird in der Automobilindustrie vor allem wegen der zertifizierten Codegenerierung nach MISRA-C eingesetzt.

Bei der Entscheidung zwischen universeller und domänenspezifischer modellbasierter Entwicklung sollten folgende Kriterien berücksichtigt werden.

- ***Einführung der modellbasierten Entwicklung***: Für die Entwicklung von UML- und SysML-Modellen stehen hinreichend viele industrietaugliche Werkzeuge zur Verfügung. Auch methodische Schulungen werden in hinreichendem Umfang und Qualität angeboten. Für die Erstellung einer domänenspezifischen Infrastruktur (Sprachdefinition, Editoren, Generatoren) ist in nicht unwesentlichem Umfang technisches Know-how zu entwickeln oder zu erwerben.
- ***Verwendung der modellbasierten Entwicklung***: Die Verwendung von UML- und SysML-Modellen ist nicht standardisiert. Der Nutzen hängt von der unternehmensspezifischen Verwendung ab und kann daher nicht allgemein quantifiziert werden. Domänenspezifische Modelle sind, wie oben skizziert, geeignet, informelle Aspekte systematisch und formalisiert zu erfassen, zu bewerten, an die formalen Artefakte (Programme) anzuschließen und damit eine durchgehende und nachvollziehbare Entwicklung und Qualitätssicherung effektiv zu unterstützen.

3.3.2 Agile Entwicklungsprozesse

Die EN 50128 verlangt, dass ein dokumentiertes Software Lebenszyklus-Modell in der Entwicklung angewendet wird, und gibt als Beispiele ein wasserfallartiges und ein V-Modell-artiges Phasenmodell. Daher werden in der Bahntechnik oft diese oder ähnliche Modelle eingesetzt. In der Literatur wurden etliche weitere Vorgehensmodelle für die Software-Entwicklung vorgeschlagen, z.B. Spiralmodell, modellgetriebene Software-Entwicklung (s.o.), OOAD (Object-oriented Analysis and Design), Kanban, Rational Unified Process, V-Modell XT und W-Modell, und viele andere. Oftmals ist die Überlegenheit einer Methode gegenüber anderen nicht wissenschaftlich belegt; ein prinzipielles Problem des so genannten

empirischen Software Engineering ist die Unmöglichkeit, dasselbe Produkt unter denselben Rahmenbedingungen mit zwei verschiedenen Vorgehensweisen zu entwickeln. Daher ist man hier auf Erfahrungswerte und subjektive Einschätzungen angewiesen.

In nicht sicherheitskritischen Domänen (z.B. bei der Entwicklung von Webshops) werden traditionelle Softwareentwicklungsprozesse oft als schwergewichtig und bürokratisch angesehen. Agile Methoden versuchen, mit geringerem Managementaufwand und weniger starren Regeln auszukommen, um sich schnell an Veränderungen der Anforderungen oder Rahmenbedingungen anpassen zu können. Agile Softwareentwicklungsmethoden sind eine Reaktion auf die Schwierigkeiten, die sich bei der Entwicklung komplexer Systeme ergeben. Komplexität bedeutet, dass sich die Auswirkung einer Entwurfsentscheidung auf das fertige System in der Regel nicht vollständig abschätzen lässt. Es ist bei komplexen Systemen schwierig, sämtliche Entwurfsentscheidungen von einer Anforderungserfassung und -analyse komplett zu trennen und in zwei aufeinander folgende Prozessphasen zu aufzuteilen. Jede Festlegung einer Systemeigenschaft – d.h. jede Anforderung, die über die erste Systemidee hinausgeht – ist eine Entwurfsentscheidung; und Entwurfsentscheidungen führen in der Regel zu weiteren Aufgaben, die analysiert und bewertet – d.h. in Anforderungen überführt – werden müssen.

Die systemtechnische Antwort auf die Frage nach der Beherrschung von Komplexität ist das Einbinden von Steuerungsmöglichkeiten in den Systementwicklungsprozess. In agilen Modellen werden Mechanismen bereitgestellt, mit denen Entwurfsentscheidungen revidiert oder korrigiert werden können. Diese Maßnahmen begleiten und steuern den Prozess von Anfang an. Sie sind nicht für Notfälle reserviert, sondern gehören zum regulären Prozess.

Ein weiterer Aspekt der systemtechnischen Herangehensweise, der speziell im Ansatz des Design Thinking ausformuliert wurde, ist die Einbindung des Kunden bzw. einer Kundenrolle in den Entwicklungsprozess. Oft kann ein Kunde seine Erwartungen erst dann präzisieren, wenn ein gewisser Entwicklungsstand des Systems bereits zur Verfügung steht. Dies führt zu kurzen, bisweilen sehr kurzen, Entwicklungszyklen, in denen der Kunde seine Vorstellungen mit dem Entwicklungsteam präzisiert, ausbaut und erweitert.

Agile Entwicklungsmethoden haben mittlerweile eine Reife (und Eigenkomplexität) erreicht, die ihren zuverlässigen Einsatz in der industriellen Praxis ermöglichen. Bestimmend für den Erfolg eines agilen Projekts ist allerdings, die Methode genau zu kennen und sehr diszipliniert zu befolgen. Eine nur oberflächliche Kenntnis und laxe Durchführung führen in der Regel schnell zu Problemen, die schwerwiegend sein können und im Ernstfall auch das Scheitern des Projekts zur Folge haben. Es gibt mittlerweile aber hinreichend viele und gute Schulungsmöglichkeiten für ein seriöses agiles Projektmanagement.

Die Frage ist, in wie weit sich mit agilen Verfahren mit geringerem Aufwand ein gleiches oder höheres Qualitätsniveau erreichen lässt wie mit herkömmlichen Methoden. In der Literatur werden hierzu teilweise unterschiedliche Meinungen vertreten: **Cawley** et al. (2010) schreiben „Drawing on the results of a systematic literature review we find that evidence is sparse for Lean/Agile adoption in [the rigorous environment of safety-critical embedded software development].“ **Turk** et al. (2002) argumentieren „that agile and formal software development are not incompatible“, und **Jonsson** et al (2012) betonen, dass viele agile Praktiken Ziele der EN 50128 unterstützen. **Auch Gary** et al (2011) widersprechen der gängigen Auffassung, dass agile Methoden und open-source Entwicklungen ungeeignet für sicherheitskritische Software seien. Im Wesentlichen sind agile Entwicklungszyklen kurz und Kunde bzw. Kundenrolle und Entwicklerteam eng verbunden. Die bekannten Techniken der Sicherheitsanalyse und -entwicklung zum Beispiel lassen sich aber problemlos auch in kurze Zyklen integrieren.

Noch offen sind hingegen Fragen nach der Größe der Systeme bzw. der Entwicklerteams, die sich gut mit agilen Methoden vertragen. Wie können übergreifende Architekturentscheidungen agil, in kurzen, ggf. revidierbaren Zyklen getroffen werden? Und wie verhalten sich diese Architekturentscheidungen

dann zu dem bereits entwickelten Stand des Systems? Wie kann ein großes Team so aufgeteilt werden, dass die Vorteile des agilen Teamentwickelns trotz Spezialisierung und Einteilung bestehen bleiben? In der Praxis werden hier bereits verschiedene Verfahren erprobt. Eine vereinheitlichte Methode scheint aber noch nicht etabliert zu sein.

3.3.3 IT-Sicherheit (Cyber Security)

Die mit der Digitalisierung einhergehende zunehmende Vernetzung aller Branchen und Lebensbereiche eröffnet nicht nur Geschäftsmöglichkeiten, es steigt gleichzeitig auch die Gefahr für (weltweit ausgeführte) Cyberattacken über diese Netzwerke. Cyberattacken betreffen alle Branchen der Wirtschaft, besonders häufig jedoch sind der Finanzsektor, das Gesundheitswesen, Handel, Telekommunikation, Produktion und Behörden betroffen. Allerdings wechseln Fokus und Schwerpunkt der Attacken häufig. Im „Cyber Security Intelligence Index 2016“ (einer von IBM jährlich erhobene Statistik über Cyberattacken) wird bspw. auch die Transportindustrie als eine der fünf häufigsten betroffenen Branchen im Jahr 2015 genannt.

Unternehmen müssen sich auf diese Bedrohungslage vorbereiten. Sie sollten dabei nicht nur von Anfang an Sicherheitslösungen in ihre IT-Technologie implementieren, sondern auch ein geschärftes Bewusstsein in der Unternehmenskultur entwickeln. Die Technologien zur sicheren Übertragung von Informationen über Netzwerke sind vorhanden und alltagstauglich. Die Systeme am jeweiligen Ende der Kommunikationskette haben dagegen häufig noch Defizite bei der Absicherung gegen kriminelle Cyberangriffe, insbesondere, wenn es sich um Systeme handelt, die bereits seit längerem auf dem Markt sind. Nachrüsten gestaltet sich oftmals als schwierig, dass auch IT-Sicherheit nicht ohne Einsatz von Ressourcen gewährleitstet werden kann, die aber in Altsystemen häufig nicht zur Verfügung stehen.

Obwohl im Markt leistungsfähige, für den unternehmensweiten Einsatz geeignete Lösungen z.B. für das Identitätsmanagement und den gesicherten Zugriff auf Daten und Funktionen zur Verfügung stehen, scheitert deren Verwendung viel zu häufig noch an deren bestimmungsgerechten Einsatz durch Mitarbeiter, Entwickler und Kunden. Unsichere, leicht ableitbare oder auslesbare, nicht vergebene oder gedankenlos weitergegebene Passwörter sind keine Seltenheit, sondern ein massives Alltagsproblem, dass jede Form technologische Absicherung unterminieren kann. Hier besteht noch ein großer Bedarf hinsichtlich der Sensibilisierung von Entwicklern und Anwendern für das Thema IT-Sicherheit.

Aus Produktsicht ist die Einbruchssicherheit eine Eigenschaft, die nicht „nachträglich hinzugefügt“ werden kann, sondern es müssen Vorkehrungen getroffen werden, wie Produkte kontinuierlich auch gegen neu auftretende Schwachstellen immunisiert werden können. Das bedeutet vor allem, dass sichere Mechanismen geschaffen werden müssen, die es erlauben, Produkte zu aktualisieren, ohne die Ausfallsicherheit (Safety) zu gefährden.

Die Bahninfrastruktur setzt derzeit noch auf branchenspezifischen, geschlossenen Kommunikationslösungen, deren Weiterentwicklung jedoch zeitaufwändig und teuer ist. Mit der zunehmenden Digitalisierung (etwa durch die Einführung digitaler Stellwerke) findet aber auch hier ein Paradigmenwechsel weg von geschlossenen Lösungen und hin zur Verwendung von Standardkomponenten und der Nutzung des Internets statt. Dadurch werden die Datennetze der Leit- und Sicherungstechnik (LST) leistungsfähiger, gleichzeitig werden sie jedoch auch Hackerangriffen ausgesetzt. Die potentiellen Folgen der Angriffe können von ärgerlichen Verspätungen bis hin zu kritischen Störungen reichen, die Auswirkungen auf Leib und Leben der Zuginsassen haben. Deshalb ist es (auch im Sinne der funktionalen Sicherheit) essentiell, geeignete IT-Sicherheitslösungen in die LST zu integrieren. Die funktionalen Sicherheits-Anforderungen dürfen dabei nicht durch Funktionen der IT-Sicherheit beeinträchtigt werden.

Die gleichzeitige Gewährleistung der Funktions- und IT-Sicherheit ist eine wesentliche Voraussetzung, um den Paradigmenwechsel in der Bahnindustrie umzusetzen. Dies ist bspw. Ziel des Forschungsprojekts „Hardwarebasierte Sicherheitsplattform für Eisenbahn Leit- und Sicherungstechnik“ (Projekt HASELNUSS, siehe **BMBF** (2017) sowie Abbildung 6: HASELNUSS Projekt (Kurzbeschreibung)). Die im Projekt angestrebte IT-Sicherheitsarchitektur basiert auf einem modernen Hardware-Sicherheits-Modul, das als nicht manipulierbarer Sicherheitsanker dient, sowie einem Softwarekern, dessen Sicherheit aufgrund der geringen Größe des Programmcodes mit formalen Methoden vollständig überprüft werden kann. Weiterhin werden sichere Boot- und Update-Mechanismen für die Sicherheitsplattform erforscht, um einen sicheren Softwarelebenszyklus zu gewährleisten. Die Fähigkeit, nicht nur selektiv und partikulär, sondern regelmäßig und flächendeckend Softwareupdates durchführen zu können ist essentiell für die Gewährleistung von IT-Sicherheit. Diesen Prozess qualitätsgesichert so durchführen zu können, dass die Anforderungen an die funktionale Sicherheit (Safety) und die damit verbundenen Zulassungsprozesse dadurch nicht beeinträchtigt werden, wird eine der zentralen Herausforderungen für die Bahn in der Zukunft werden.

BMBF-Projekt HASELNUSS
IT-Sicherheit für die Bahn der Zukunft
Laufzeit 01/2017-12/2019 - https://www.haselnuss-projekt.de/
Ziel des Projekts „Hardwarebasierte Sicherheitsplattform für Eisenbahn Leit- und Sicherungstechnik“ (HASELNUSS) ist die Entwicklung einer IT-Sicherheitsplattform, die an die speziellen Anforderungen von LST-Anlagen angepasst ist. Dazu gehören die Einhaltung kurzer Reaktionszeiten etwa bei Notbremsungen genauso wie die Gewährleistung der funktionalen Sicherheit (Safety). Die Safety-Anforderungen dürfen dabei nicht durch Funktionen der IT-Sicherheit (Security) beeinträchtigt werden. Die Architektur basiert auf einem „Trusted Platform Module (TPM) 2.0“, welches als Sicherheitsanker dient, sowie dem Mikrokern-basierten Betriebssystem PikeOS, das eine sichere Koexistenz von kritischen und unkritischen Anwendungen erlaubt und einfach zu verifizieren ist. Auf dieser Grundlage werden Dienste für sicheres Patch- und Update-Management, Health Monitoring, Anomalie- und Angriffserkennung umgesetzt.

Abbildung 6: HASELNUSS Projekt (Kurzbeschreibung)

3.4 Ausbildung

Ein eingebettetes System ist ein Informatiksystem, welches fester Bestandteil eines technischen Systems ist. In diesem Sinne sind alle in dieser Studie referenzierten Bahnsysteme eingebettete Systeme. Ein charakteristisches Merkmal der Entwicklung eingebetteter Systeme ist die Notwendigkeit der Zusammenarbeit von Informatikern und Ingenieuren. **Genc** (2014) betont die Bedeutung interdisziplinärer Qualifikationen in Eisenbahnprojekten. Leider gibt es erst wenige Universitäten, die entsprechende Curricula anbieten. Ein Beispiel ist in **Schlingloff** (2015) zu finden. Ob solche Studiengänge als interdisziplinäre Masterstudiengänge angeboten werden, liegt in der Verantwortung der einzelnen Hochschulen.

Je mehr das System Bahn die Charakteristiken eines IT-Produkts annimmt, desto mehr stehen die Hersteller und Betreiber aber auch in einem Konkurrenzkampf um die besten IT-Spezialisten. Ebenso wie andere Firmen sicherheitskritischer Domänen, konkurriert die Bahnindustrie daher immer mehr mit den großen Unternehmen der IT-Industrie um die besten Talente. Auch wenn die Entwicklung von Steuergeräten und Web-Applikationen auf den ersten Blick unterschiedliche Anforderungen an den Entwickler stellen, kommt es durchaus vor, dass Softwareentwickler eines Herstellers von Steuergeräten im Bahnbereich etwa zu Google wechseln.

Um die wachsende Komplexität des Systems Bahn auch in Zukunft beherrschen zu können, werden aber talentierte Entwickler benötigt. Die Eisenbahn ist zwar bekannt dafür Menschen zu begeistern, dies allein wird aber in Zukunft nicht genügen, um für die besten Köpfe auch gegen IT-Unternehmen konkurrieren zu können. Ein Ausbau der oft schon langfristig bestehenden Kooperationen mit Universitäten und Forschungseinrichtungen ist eine Möglichkeit, IT-Experten frühzeitig mit den Anforderungen des Systems Bahn vertraut zu machen. Andererseits würde eine konsequente Hinwendung zu modernen Softwareentwicklungsmethoden im Bahnbereich es auch leichter machen, IT-Fachleute aus anderen Domänen anzuwerben. Für die Vermittlung des erforderlichen Domänenwissens haben viele Unternehmen der Bahnindustrie bereits Aus- und Weiterbildungsprogramme etabliert.

Qualitätssicherung im Allgemeinen und Softwarequalitätssicherung im Besonderen sind wesentliche Bedingungen, um sowohl die funktionale als auch die IT-Sicherheit der Bahnsysteme zu gewährleisten. IT-Sicherheit ist im Gegensatz zu funktionaler Sicherheit ein gut etabliertes Thema im Informatikstudium. Die Defizite von Berufseinsteigern auf dem Gebiet der funktionalen Sicherheit werden in der Regel durch unternehmensinterne Aus- und Weiterbildungsprogramme kompensiert. Spezialisten für IT-Sicherheit sind zunehmend gefragt auf dem Arbeitsmarkt. Gemäß **Cybersecurity Ventures** (2018) soll es bis 2021 rund 3,5 Millionen unbesetzte Stellen in diesem Bereich geben. Die Herausforderung dabei ist, die Spezialisten für IT-Sicherheit so zu schulen, dass sie sich so schnell wie möglich die nötigen Fähigkeiten aneignen und diese auch weiterentwickeln. So wie sich die Technik weiterentwickelt, müssen sich auch die Kenntnisse für Sicherheitslösungen den veränderten Bedürfnissen anpassen.

4 Anwendungsbeispiele

In diesem Abschnitt reflektieren wir die Anwendung der oben genannten Vorschläge an Hand konkreter Anwendungsbeispiele. Wir konzentrieren uns dabei auf Möglichkeiten der Integration verschiedener Informatiksysteme in der Domäne des schienengebundenen Verkehrs. Dazu definieren wir zunächst verschiedene Integrationsstufen von Hardware und Software. Sodann betrachten wir beispielhaft die Möglichkeiten zur Integration von Streckenzentrale und elektronischem Stellwerk. Anschließend behandeln wir die Integration der Funktionalität der Fahrzeugsteuerung und Zugsteuerung, Zugsicherung und Signalisierung hinsichtlich Anzeige und Sensorik. Dabei spielt die Behandlung unterschiedlicher Kritikalitäten eine wichtige Rolle.

4.1 Integration

Integration bezeichnet allgemein den Zusammenschluss von einzelnen Funktionseinheiten bzw. Bauelementen zu einem komplexen System oder in ein bestehendes System. Dadurch kann die Anzahl der nach außen sichtbaren Schnittstellen der einzelnen Einheiten reduziert und die Dopplung von Teilfunktionen vermieden werden.

In einer prozessorientierten Sicht steht der Begriff für den Vorgang der Zusammenfügung bei der Entwicklung des Gesamtsystems, in der Produktsicht für das Ergebnis.

Bei der Integration von Informatiksystemen sind grundsätzlich zwei Dimensionen zu unterscheiden, die voneinander weitgehend unabhängig sind: Hardware und Software.

Hardware: Bei der Integration der Hardware ist vor allem die räumliche Nähe und elektrische Verbindung relevant. Wir unterscheiden folgende Ebenen:

- Keine Integration: Die Teilsysteme sind in getrennten Räumen und auf getrennten Boards untergebracht, es gibt keine elektrische Verbindung. Übertragung von Information geschieht durch mechanische (z.B. Taster), elektromagnetische (Funk) oder optische (z.B. Optokoppler) Weise. Beispiel: OBU und RBC
- Lose Integration: Die Teilsysteme sind in räumlicher Nähe (z.B. im selben Gehäuse) untergebracht und haben elektrische Verbindung, z.B. gemeinsame Stromversorgung. Sie haben jedoch keine direkte elektronische Verbindung (Datenleitungen) und keine gemeinsame Peripherie (Sensoren, Aktuatoren, Speicher). Beispiel: ESTW und GSM-R Gerät im Stellwerk, redundante Switches im RBC
- Mittlere Integration: Die Teilsysteme sind in räumlicher Nähe und haben vielfältige elektrische und elektronische Verbindungen (Bussysteme, Hintergrundspeicher, ...). Beispiel: EVC und DMI
- Enge Integration: Die Teilsysteme sind in enger räumlicher Nähe (z.B. auf derselben Platine) untergebracht, mit gemeinsamer Stromversorgung und Peripherie. Jedes Teilsystem hat jedoch eigene Prozessoren und Speicher. Beispiel: integriertes ETCS – PZB/LZB-Gerät
- Vollständige Integration: Die Teilsysteme sind auf demselben Chip untergebracht (z.B. ICs, FPGAs, Multicore etc.), und können auf dieselben Speicherzellen und System-Busse zugreifen. Beispiel: integriertes Entertainment-/Fahrgastinformationssystem

Natürlich ist diese Einteilung nur als grobe Orientierung zu verstehen und gibt keine endgültige Klassifikation. Zwischenstufen und Mischformen sind möglich, z.B., wenn Teilsysteme in räumlicher Nähe untergebracht sind und nur wenige oder indirekte Datenverbindungen haben.

Software:

- Keine Integration: Getrennte Programme, getrennt entwickelt, kompiliert und installiert; keine Synchronisation; Datenaustausch über externe Medien oder Dateien mit vordefinierten Formaten oder Import/Export-Filtern. Beispiel: OBU und Türsteuerung
- Lose Integration: Getrennte Programme, Datenaustausch und Synchronisation über Nachrichten (z.B. SOA, Web Services, Micro Services); standardisierte Nachrichtenformate und –protokolle. Beispiel: TCMS und Türsteuerungssoftware
- Mittlere Integration: Ein gemeinsames Grundprogramm, zu dem Komponenten (zur Entwicklungs-, Installations- oder Laufzeit, ggf. auch nachträglich) hinzugefügt werden können (Plug-Ins, Module, DLLs); ein Framework, Middleware oder Betriebssystem zur Verwaltung gemeinsamer Ressourcen, Kommunikation und Synchronisation. Beispiel: generische TCMS-Software mit Modulen/Treibern für verschiedene Fahrzeuggeräte
- Enge Integration: Modularisierte Software; jedes Modul realisiert einen klar definierten Funktionsumfang, aber es gibt vielfältige Abhängigkeiten zwischen den Modulen und Zugriff auf gemeinsame Daten. Beispiel: ETCS-Software
- Vollständige Integration: Ein monolithisches Programm, die Funktionen und Daten sind logisch nicht voneinander zu trennen. Beispiel: Türsteuer-Software

Generell kann man Trends angeben, wie sich der Grad der Integration aufgrund technischer Gegebenheiten entwickeln wird. Für die Hardware waren in der Vergangenheit in der Regel die Kostenaspekte treibend. Intuitiv drängt sich der Gedanke auf, dass die Kosten für die gleiche Performance seit Jahrzehnten praktisch konstant gefallen sind und der Preis für CPU Leistung und die Speicherkosten heutzutage vernachlässigbar sind. Bis zu einem gewissen Grad ist dies für Produkte außerhalb des Massenmarktes sicher zutreffend; heutzutage wird im Prinzip großzügig und in vielen Fällen sogar verschwenderisch mit Hardware-Ressourcen umgegangen. In der Vergangenheit führten der Preisverfall und die hohe Verfügbarkeit zur Dezentralisierung („Personal Computer“) bis hin zur Personalisierung („Smart Phone“).

Aber Hardware kostet trotzdem Geld; und zwar stehen nun eher die Kosten pro Installation und nicht mehr die leistungsorientierten Kosten im Fokus. Daher ist insbesondere im professionellen Umfeld ein Trend zu möglichst effizienter und vollständiger Nutzung, also auch vollständiger Integration zu beobachten („Cloud Computing“), um auch die Anzahl der Installationen gering zu halten. Hierbei ist als erster Schritt eine Standardisierung zu sehen, um Hardware zunächst durch verschiedene Anwendungen nutzbar und somit auch wiederverwendbar zu machen (engl. „reuse“). Einsparungen von Entwicklungskosten und durch Serieneffekte werden in diesem Szenario wirksam. Wesentlich deutlicher werden die Einsparungen allerdings, wenn die gleichzeitige Nutzung von Ressourcen (engl. „sharing“) möglich wird. Real können Hardware-Ressourcen oft von sehr vielen Nutzern bzw. Anwendungen quasi gleichzeitig genutzt werden. Die Technik hierfür ist bereits ganz am Anfang der Verbreitung von Computern in den 1960er Jahren entstanden – sogenannte virtuelle Maschinen. Die kurz "Virtualisierung" genannte Technik wurde ja schon mehrfach erwähnt, ihr kommt allerdings bei der Diskussion, ob und wie Hardware und/oder Software integriert werden kann, eine zentrale Rolle zu, weil hierdurch eine wirksame Trennung der technischen Bereiche möglich ist, und damit auch die Diskussion um Interdependenzen obsolet wird.

Gänzlich anders wie bei der Diskussion um Hardware-Integration stellt sich die Situation im Bereich der Software dar. Mit zunehmender Integrationsdichte steigt die Anzahl der Komponenten, die in einem System untergebracht werden können, und damit in quadratischem Maß die Anzahl möglicher Verbindungen dieser Komponenten. Die Komplexität der gegenseitigen Abhängigkeiten birgt vielfältige Möglichkeiten für Fehler. Da viele der Verbindungen der Einzelkomponenten nach außen hin nicht direkt sichtbar sind, sind diese Fehler systemintern und schwer zu analysieren. Daher ist die Validierung hochintegrierter, komplexer Systeme im Allgemeinen schwieriger als die einfacher, nichtintegrierter Systeme,

bei der alle Schnittstellen beobachtet werden können. Wie weiter unten gezeigt wird, ist dies insbesondere bei der Integration von sicherheitskritischen und nicht sicherheitskritischen Komponenten ein Problem.

Daher ist bei der Software in der Regel eine deutliche, nachvollziehbare Trennung in kleine, überschaubare Einheiten geboten. Allerdings betrifft dies (nur) die logische Trennung und impliziert nicht die Notwendigkeit einer physikalischen Trennung. Somit ist auch keine Strukturaussage bezüglich der Hardware gefallen. Vielmehr ist die Frage, wie eine diese logische Trennung durch eine quasi physikalische Trennung sicherzustellen und nachzuweisen ist. Kurz gesagt, bei der Software sind kleine funktionale Einheiten anzustreben, die in einem weiteren Schritt möglichst freizügig einer Ausführungsressource zugeordnet werden.

Eine wesentliche Aufgabe der in Kapitel 2 vorgestellten Plattformen besteht darin, diese trennende Schicht, besser gesagt, Hülle zu definieren und bereitzustellen. Das Fehlen einer solchen allgemeinen Plattform im Bahnbereich schränkt die Möglichkeiten daher erheblich ein. Die im Folgenden dargestellten Szenarien sollen eine Vorstellung davon vermitteln, wie Integration mit einem solchen Middleware-basierten System funktioniert und welch weitgehende Konfigurationsmöglichkeiten sich dadurch bieten.

4.1.1 Integration RBC – ESTW / DSTW

Die Streckenzentrale (RBC, Radio Block Center) ist eine zentrale Komponente im ETCS. Die Hauptaufgabe des RBC ist die Führung der Züge in einem bestimmten Streckenabschnitt. Dazu erhält es vom Fahrzeug Zugdaten und vom Stellwerk Streckendaten, und generiert daraus eine Movement Authority für den Zug. Parametrisiert (projektiert) wird das RBC mit der Topologie der Strecke (Streckenatlas).

Die Hauptaufgabe eines STW (Stellwerks) ist das sichere Einstellen von Fahrstraßen für Züge. Eine Fahrstraße ist sozusagen für den Zug der „Sichtbereich“ und für andere Züge die „Sichtbarkeit“. Dazu generiert das STW Schaltbefehle für Weichen- und Schrankenantriebe sowie Signale, und nimmt Meldungen der Gleisfreimeldeanlagen entgegen. Schaltbefehle an die Stelleinheiten können mechanisch, elektrisch, oder elektronisch (im ESTW) oder digital (DSTW) erfolgen. Weitere Schnittstellen existieren zum RBC und ggf. zum Nachbarstellwerk. Bei einem DSTW sind die Schnittstellen überwiegend Ethernet- und IP-basiert. Dadurch werden einerseits analoge Signalkabel überflüssig, andererseits können Stelleinheiten in beliebiger Entfernung zum STW angesprochen werden.

Üblicherweise sind RBC und STW Systeme verschiedener Hersteller, die räumlich getrennt sind und nur über wenige gemeinsame Daten verfügen. Nachrichten, z.B. vom Zug, werden meist auf getrennten Wegen übertragen. Diese Ausführungen zeigen, dass RBC und STW derzeit als lose gekoppelte Systeme betrachtet werden können. Die Verzahnung von Streckenzentrale und Stellwerk und insbesondere der Konzentrationsprozess bei den Stellwerken legt daher die Überlegung nahe, ihre Funktionen physisch in einem einzigen System zusammenzuführen. Nachfolgend werden Chancen und Risiken einer engeren Integration diskutiert.

Analog zur Einleitung dieses Kapitels sind dabei verschiedene Integrationsszenarien zu unterscheiden und zu bewerten. Wir werden im Folgenden die Integrationsaspekte unter den prägenden Aspekten von

- Anwendungssoftware,
- Hardware, und
- Middleware-Plattform

diskutieren.

Anwendungssoftware: Betrachtet man die Funktionalität „Sicherung einer Zugfahrt“, stellt man schnell fest, dass dies weit mehr erfordert als die klassische Aufgabe eines STW gemeinhin umfasst, nämlich ein freies Gleis zu gewährleisten, Weichen zu stellen, eventuell noch Flankenschutz und Durchrutschweg sicherzustellen, das Ganze als Fahrstraße zu sichern und das Signal auf Grün zu stellen. Natürlich ist dies alles notwendig, allerdings für die dynamische Komponente der Zugfahrt nicht hinreichend, denn hier muss mindestens noch die Geschwindigkeit betrachtet werden. Und spätestens bei der Teilfunktion „Überwachung der Geschwindigkeit“ stellt man fest, dass diese Funktion streckenseitig nur punktuell geleistet werden kann und daher aus gutem Grunde seit Beginn der Eisenbahn im Fahrzeug selbst, zuerst nur vom Lokführer, inzwischen auch durch Technik, vorgenommen wird.

Hier ist also das typische Muster eines Edge Computing zu sehen, wie es in der Telekom (Kapitel 2.3) - und Automatisierungsindustrie (Kapitel 2.4) in den Plattformen vorgesehen wurde. Das System der Sicherung einer Zugfahrt ist in dieser Form eindeutig ein verteiltes System. Planung, Allokation und Teilfunktionen der Überwachung sind an der Strecke (STW und RBC), die Überwachung der Geschwindigkeit erfolgt im Zug durch die OBU (an der Edge (Kante), nahe am physikalischen Prozess). In diesem Sinne ist eine wesentliche Funktion eines RBC die streckenseitige Stellvertreterfunktion des Zuges. In dieser Rolle fordert das RBC vom STW die Daten der Fahrstraße an und sorgt für die konsistente und synchronisierte Übertragung dieser Daten zur OBU. Die Fahrstraße liegt also logisch in insgesamt drei Ausführungen vor; einmal im STW, einmal im RBC und noch einmal in der OBU. Das RBC als Mittler, Transformierer und Verdichter hat also datentechnisch im Kern eine klassische Replikations- und Synchronisationsaufgabe.

Damit wird auch deutlich, dass sowohl die RBC- als auch die Stellwerks-Software auf gemeinsame Grundfunktionen zugreifen; daher liegt es zwar auf der Hand, sich über eine mittlere Integration Gedanken zu machen. Wie aber zuvor ausgeführt, sind Replikation und Synchronisation ganz normale Funktionen in vielen IT Systemen, die allerdings technisch gelöst und nicht etwa vermieden werden sollten. Das Szenario einer engeren Integration von RBC- und Stellwerks-Software erscheint uns problematisch. Je höher die Anzahl der Module und ihrer wechselseitigen Abhängigkeiten, desto größer sind auch die Möglichkeiten für Fehler. Für ein hochgradig modulares System ist es eine große Herausforderung, Integrationstests zu entwickeln, die alle möglichen Interaktionen berücksichtigen. Eine mögliche Vorgehensweise wäre es, bei der Validierung auf Modul-Beweise und Assume-Guarantee-Verfahren zur Verifikation der Interaktion zu setzen. Damit könnte der Testaufwand auf ein akzeptables Maß reduziert werden, allerding mit erhöhtem Aufwand für formale Verifikation. Entsprechende Verfahren sind aber in der Industrie bislang noch nicht etabliert.

Dass diese Überlegungen keines falls reine Theorie sind, belegt eine aktuelle Studie der Schweizerischen Bundesbahn (siehe **ESG** (2018)). Dabei soll eine als ETCS-Stellwerk bezeichnete Kombination von RBC und Stellwerk entwickelt werden. Die Applikationen sollen dabei in zentralisierten Datenzentren auf standardisierter, kommerzieller Hardware laufen. Die ESG empfiehlt dabei eine frühe Einbeziehung der Zulassungsbehörden, sowie die Verwendung bereits qualifizierter Server-Komponenten als Hardware-Basis.

Es sei in diesem Zusammenhang darauf hingewiesen, dass die Funktionalität eines RBC, eines DSTW und auch einer OBU komplett durch Software darstellbar sind. Und damit kann und muss die Verteilung von Teilfunktionen der Gesamtfunktionalität „Sicherung einer Zugfahrt“ nicht fest bestimmten Komponenten zugeordnet werden, sondern diese Zuordnung ist je nach sich ändernden Struktureigenschaften oder Funktionsbereichen neu zu diskutieren. Die höhere Hardwareintegration ist genauso eine neue Struktureigenschaft. Zur Verdeutlichung: z.B. die Berechnung der Geschwindigkeitsvorgaben aus Strecken- und Zugprofil stellt eine Berechnungsfunktion dar, die nur vom Vorhandensein der notwendigen Parameter abhängt, aber prinzipiell sowohl von einer Streckeneinrichtung (z.B. dem RBC), aber auch von dem Zug (z.B. OBU) selbst vorgenommen werden kann.

Aus der Definition des Begriffs *Integration* folgt, dass es bei der Aufgabe zur Betrachtung von Streckenzentrale und Stellwerk an dieser Stelle um eine viel grundlegendere Fragestellung geht, nämlich, schon auf der Ebene der Systemanforderungen nicht mehr künstlich zwischen beiden Einheiten zu unterscheiden. Die Existenz beider Systeme ergibt sich aus der Geschichte der Leit- und Sicherungstechnik. Bei der Definition von ETCS wurden für die neuen Aufgaben der Kommunikation und der Sicherungstechnik sowohl an der Strecke als auch im Zug einfach entsprechend neue Komponenten definiert, das RBC und die OBU. Im ETCS ist die Schnittstelle zwischen Streckenzentrale und Stellwerk, im Gegensatz zur Schnittstelle zwischen Streckenzentrale und Fahrzeug, meist nur national bzw. vom Betreiber standardisiert. Vor diesem Hintergrund wäre aus unserer Sicht also eher eine Gesamtbetrachtung und daraus abgeleitet auch eine Gesamtarchitektur mit der Identifikation geeigneter Teilfunktionen angezeigt, die wir im Folgenden jedoch nur punktuell andeuten können.

Aus den oben dargestellten Überlegungen lassen sich folgende Konsequenzen ziehen. Die Allokation und Integration von (logischen) Softwarebausteinen sollte nicht an die Integration von (physischen) Hardwarekomponenten gekoppelt werden. Für eine weitergehende logische Integration, über die in Zukunft im Rahmen des EULynx Projekts (siehe Abbildung 5: euLyNX Projekt (Kurzbeschreibung)) einheitlich definierten Schnittstellen hinaus, besteht keine Notwendigkeit. Ganz im Gegenteil sollten die Teilfunktionen klarer identifiziert werden, um so das Verständnis und vor allem auch die Integration neuer Funktionalitäten zu fördern.

Gegen eine vollständige, logische Integration von RBC- und Stellwerks-Software spricht vor allem der überproportional wachsende Aufwand für die Validierung. Monolithische Programme sind erheblich schwerer zu verifizieren als modularisierte Programme. Auf Grund der hohen Komplexität der beiden Komponenten erscheint es uns nicht zukunftssicher, diese zu einem Programm zu verschmelzen. Eine Lösung in Richtung Middleware bzw. Virtualisierung verspricht mehr Modularität, mehr Transparenz, weniger Gesamtkomplexität, mehr Flexibilität, bessere Langzeitstabilität – um nur einiges zu nennen.

Hardware: Die bisherige räumliche Trennung der beiden Systeme ist hauptsächlich bedingt durch historische Einschränkungen bei den maximalen Längen von analogen Datenleitungen. In IP-basierten Netzen entfallen diese Beschränkungen, Daten und Befehle können über beliebige Strecken übertragen werden. (Ein weiterer historischer Grund für die räumliche Nähe eines Stellwerks zu den einzustellenden Weichen war die visuelle Kontrolle der Schaltfunktion; auch diese Einschränkung ist heute als obsolet zu betrachten.) Daher ist zu erwarten, dass in absehbarer Zeit zumindest eine mittlere Integration flächendeckend erreicht werden kann und aus Kostengründen von den Betreibern auch angestrebt wird; sämtliche Signalisierungs- und Schaltfunktionalität wird in wenigen Schaltzentralen oder Betriebsleitstellen konzentriert werden.

Auch eine enge oder vollständige Integration ist denkbar. Für eine vollständige Integration wäre eine einheitliche, integrierte Hardware-Plattform erforderlich, die sowohl die Stellwerks- als auch die RBC-Funktionalität beherbergt. Das ist von der Hardware heute möglich, aber die „normale“, komponentenorientierte Systemstruktur im Bahnbereich lässt eine Hardwareintegration nur zusammen mit einer Softwareintegration zu. Dieser Zusammenhang ist der eigentliche entscheidende Faktor. Zur Verdeutlichung: wenn auf einer sehr leistungsfähigen Hardware nur ein sehr großes STW laufen kann, handelt man sich ein immenses Problem bei der Integration der Anwendungssoftware bzw. der Anwendung selbst ein. Wünschenswert ist stattdessen die Hardwareintegration unabhängig von der Softwareintegration; d.h. auf einer sehr leistungsfähigen Hardware können mehrere STW laufen und die Größe der STW hängt einzig von den Bedürfnissen der Betreiber ab.

Middleware-Plattform: Das Thema wurde oben schon eingehend angesprochen. Middleware-Plattformen trennen Hardware von Software, indem sie eine virtuelle Hardware bereitstellen. Plattformen wie IMA im Avionics-Bereich und AUTOSAR im Automotive-Bereich machen exakt diese Trennung möglich. Ähnlich wie ein IMA-Modul im Avionics-Bereich könnte ein solcher „universeller sicherer

Bahncomputer“ oder sogar ein „sicheres Rechenzentrum für sicherheitskritische Bahnapplikationen“ je nach Skalierung verschiedene sicherheitskritische Software-Programme gleichzeitig ausführen. Zur Verdeutlichung: auf einer solchen Plattform können mehrere Software-STW und Software-RBC ausgeführt werden. Natürlich müsste das System genug Redundanz aufweisen, um die höchste Sicherheitsanforderungsstufe zu erfüllen. Beispielsweise müsste auch die Rechenkapazität dabei so ausgelegt sein, dass maximale Reaktionszeiten garantiert werden können. Beim Stand der Technik erscheint es uns aber ohne weiteres machbar, solch ein Rechensystem zu entwickeln.

Es sei hier auch angemerkt, dass solch ein Betriebsszenario auch ohne bahnspezifische Plattformarchitektur realisierbar ist, indem man z.B. auf typischerweise in der Cloud (bzw. den Rechenzentren in der Cloud) verwendete Techniken wie z.B. Virtualisierung zurückgreift. Eine bahnspezifische Middleware-Plattform hätte den Vorteil, nur einen wesentlichen kleineren Funktionsumfang anbieten zu müssen, was einer Qualifizierung sicher sehr entgegenkommen würde.

Die dadurch entstehenden Freiräume bei der Verteilung der Software sind immens. Man denke nur, mit wie wenigen Rechenzentren „Cloud“-basierte Systeme auskommen. Von daher sollte die Leistung eines Rechenzentrums ohne Berücksichtigung weiterer Aspekte wie Sicherheit oder Redundanz für ein landesweites Stellwerk ausreichend sein. Im Wartungsbereich sollten sich - und dies gilt im Übrigen auch unabhängig von einer Zentralisierung – Administrationsarbeiten, z.B. auch Updates aus der Ferne vornehmen lassen. Das Roll Out von ETCS L3 würde sich in solch einem Szenario als reines Software-Update gestalten.

Zusammenfassend sind folgende Argumente bei der Frage nach einer Integration von STW und RBC zu bedenken:

- Die Zentralisierung von Hardwarestandorten ist möglich und aus Kostengründen wünschenswert.
- Funktionen werden hauptsächlich noch durch Software repräsentiert.
- Nutzung von Virtualisierung und/oder einer Middleware-Plattform entkoppeln die Integrationsaspekte von physischer Hardware und logischen Softwarebausteinen.
- Durch diese Unabhängigkeit entstehen praktisch beliebige Freiräume für die Zuordnung und Verteilung von Software über die gesamte Lebensdauer.

4.1.2 Integration OBU – TCMS

In diesem Abschnitt betrachten wir Vor- und Nachteile der Integration von Systemen bzw. Funktionalitäten unterschiedlicher Sicherheitsanforderungsstufen (SIL). Als Beispiel wählen wir die Fahrzeugeinrichtung (On-Board-Unit, OBU) im ETCS und das Zugsteuersystem (Train Control and Monitoring System, TCMS).

Die OBU ist eine Komponente des ETCS, deren Aufgabe es ist, die vom RBC empfangenen Daten auszuwerten, dem Triebfahrzeugführer anzuzeigen und den Zug gegebenenfalls vor einem Gefahrenpunkt zum Halt zu bringen. Dies ist klarerweise eine SIL4 Funktionalität.

Das TCMS steuert, überwacht und verwaltet verschiedene Ausrüstungen an Bord des Zuges, z. B. Türen, Antrieb, Klimaanlage usw. Es kann auch mit sicherheitsrelevanten Funktionen der Fahrzeugsteuerung, z.B. Bremsen oder Schlupfüberwachung beim Bremsen, ausgestattet sein; oft enthält das TCMS jedoch nur oder überwiegend SIL2 Funktionalitäten. Sicherheitsrelevante Funktionen mit SIL4 wie z.B. die Durchführung des Bremsvorgangs im gesamten Zug werden durch getrennte Systeme ausgeführt.

Es gibt etliche Komponenten, die OBU und TCMS gemeinsam verwenden könnten, z.B. die Führerstandsanzeige, und es ist vor allem damit zu rechnen, dass zukünftig weitere Funktionen, wie z.B. die

Zugintegritätsprüfung für ETCS L3, sicherheitsrelevant werden. Aus wirtschaftlichen Gründen würde es sich also anbieten, diese zu integrieren. Für Funktionen wie Break-by-wire müssen TCMS und OBU notwendigerweise zusammenarbeiten. Daher könnte eine Integration der entsprechenden Funktionen aus Komplexitätsgründen sinnvoll sein.

Die europäische Norm (EN) 50129 verlangt in A.4.2.1, dass ein Subsystem, d.h. die Kombination von Geräten, welches verschiedene Sicherheitsfunktionen implementiert, die Anforderungen der jeweils höchsten Sicherheitsanforderungsstufe erfüllen muss. Ähnlich verlangt die EN 50128 in §9.4.9: „Wo die Software aus Teilen unterschiedlicher Software-Sicherheitsanforderungsstufen besteht, müssen alle Softwareteile so betrachtet werden, als würden sie der höchsten Software-Anforderungsstufe angehören“. Die Software-SIL richtet sich nach der System-SIL: EN 50128 besagt in §5.2.3, „ohne besondere Vorsorgemaßnahmen muss die Software-Sicherheitsanforderungsstufe mindestens so groß sein wie die System-Sicherheitsanforderungsstufe“. Daher wäre für ein TCMS, welche OBU-Funktionalität realisiert, sowohl für Hardware als auch für Software SIL4 nachzuweisen. Dies ist auf Grund des hohen Aufwands nicht wirtschaftlich.

Jedoch lässt die Norm Ausnahmen zu. Laut EN 50129 kann die Anforderungsstufe verringert werden, wenn die Unabhängigkeit der Einzelfunktionen nachgewiesen werden kann. EN 50128 erlaubt die Verringerung, wenn Mechanismen existieren, die verhindern, dass der Fehler eines Softwaremoduls das System in einen unsicheren Zustand bringt. Also hängt die Praktikabilität einer Integration ganz entscheidend von der Hard- und Software-Architektur der Steuerung ab. Sowohl für Hard- als auch Software kann der Nachweis der Unabhängigkeit von Einzelfunktionen schwierig sein. Mechanismen zur Eindämmung von Software-Fehlern sind z.B. Modularität, Behandlung von Ausnahmebedingungen, Redundanz und diversitäre Programmierung.

Ob und wie eine gemeinsame Verwendung sicherheitsgerichteter und nicht sicherheitsgerichteter Komponenten in einem integrierten Rahmen möglich ist, war Gegenstand des EU Horizon 2020 Projekts „Safe4Rail“ (https://safe4rail.eu/, 1.10.2016-30.9.2018) im Rahmen der Shift2Rail Joint Undertaking, siehe auch Abbildung 7: SAFE4RAIL Projekt (Kurzbeschreibung). Ein Ziel des Projekts war es, ein Konzept für einen Anwendungsrahmen mit gemischten Kritikalitäten zur modularen Integration von verteilten TCMS Anwendungen zu erstellen, welches Sicherheitsanforderungsstufen bis hin zu SIL 4 unterstützt. Als Anwendungsbeispiel wurde ein Brake-by-wire System in einem künftigen oder hypothetischen TCMS betrachtet. Die funktionale Architektur wird dabei in Servicebremse, Notbremse, Parkbremse, Bremssystem-Management mit Schlupfkontrolle, und automatischen Bremsentest heruntergebrochen. Von diesen Unterfunktionen ist nur die Notbremsfunktion, die den Zug innerhalb einer garantierten Strecke zum Halten bringt, gemeinsam für ETCS und OBU. Die Architektur garantiert „Service and Emergency brake are managed as autonomous functions, each of them with their proper requirements and dedicated information. This guarantees the functions independency” (**Safe4RAIL** (2017)). Auf diese Weise wird eine Integration der Notbremsfunktionalität von OBU und TCMS ermöglicht.

Diese trennende Funktion wurde nun schon mehrfach angesprochen; hier wird dadurch noch einmal deutlich, welches Potential solch eine Funktion als Teil einer Bahn-Middleware-Plattform freisetzen kann. Und auch die vom Projekt gewählten Lösungswege, nämlich „Instanziierung als AUTOSAR Profil“ und „Virtualisierung durch Hypervisor“ (PikeOS & Integrity) sind daher sehr gut nachvollziehbar und machen deutlich, dass eine Adaption schon bekannter Techniken an dieser Stelle einen sinnvollen und auch praktikablen Weg darstellt.

Man kann sogar noch weiter gehen und sagen, dass nicht nur die Systemarchitektur hier übernommen wurde, sondern auch wesentliche Ansätze des Entwicklungsprozesses. So wurden im Rahmen des Projektes Safe4Rail auch die Simulations- und Testmethoden bearbeitet und exemplarisch umgesetzt, wodurch sicher deutliche Kosten- und Zeiteinsparungen zu erwarten sind. Die Ausführung von Anwen-

dungssoftware verschiedener Hersteller auf einem System ist seit IMA und AUTOSAR sowohl im Luftfahrt- als auch im Automobilbereich gängige Praxis.

Zusammenfassend kann man also sagen, dass auch bei diesem Beispiel die Entwicklungen aus anderen Branchen, also

- Funktionen werden nur durch Software repräsentiert, und
- Nutzung einer einheitlichen Middleware-Plattform,

im Bahnbereich nicht nur anwendbar sind, sondern dass auch tatsächlich mit einer Anwendung zu rechnen ist. Die Parallelen zum Beispiel ETCS & RBC kommen deutlich zu Tage. Wie Safe4Rail basiert auch der bei smartrail4.0 skizzierte Ansatz zur Realisierung eines sicheren Rechenzentrums interessanterweise auf den Technologien für Fahrzeuge aus dem Avionics- und dem Automobilbereich (siehe **ESG** (2018)). Für die Bahnindustrie würde eine einzige, einheitliche Plattform, also eine Plattform sowohl für Rolling Stock als auch für LST (DSTW, RBC) eine wesentliche Verbesserung der internen Architektur und in der Folge auch aller wesentlichen Entwicklungs- und Qualifizierungsprozesse darstellen. Dass diese Entwicklungseffizienz deutliche Kosten- und Zeiteinsparungen mit sich bringen würde, liegt auf der Hand.

SAFE4RAIL Projekt
SAFE architecture for Robust distributed Application Integration in roLling stock
Shift2Rail 2016-2018 - www.safe4rail.eu
SAFE4RAIL bildet die Basis für eine grundlegend vereinfachte Embedded Computing und vernetzte TCMS-Plattform zur modularen Integration und Zertifizierung aller sicherheits-, zeit- und unternehmenskritischen Zugfunktionen, einschließlich verteilter harter Echtzeitsteuerungen, Sicherheitssignale und Funktionen bis SIL4.
Die generische Embedded-Plattform-Architektur von SAFE4RAIL ermöglicht die Integration und Virtualisierung von kritischen und unkritischen Funktionen auf rekonfigurierbaren Computer- und Netzwerkressourcen. Die Projektsimulations- und Testumgebung basiert auf den Konzepten der Hardwareabstraktion und Domänentrennung, die einen schnellen Einsatz und Test von Anwendungen ermöglichen, z.B. durch Unterstützung von frühen Funktionsintegrationstests lange vor der Fahrzeugintegration.

Abbildung 7: SAFE4RAIL Projekt (Kurzbeschreibung)

smartrail4.0 Projekt (2017-2020)
https://smartrail40.ch/
smartrail4.0 ist ein Innovationsprogramm der Schweizer Bahnbranche. Mit dem Programm smartrail 4.0 wollen die Schweizer Bahnen die Digitalisierung und das Potenzial neuer Technologien nutzen, um die Kapazität und die Sicherheit weiter zu erhöhen, die Bahninfrastruktur effizienter auszulasten, Kosten zu sparen und damit die Wettbewerbsfähigkeit der Bahn längerfristig (2020-2038) zu erhalten. Auf die SBB bezogene Ziele sind die Reduktion der Außenanlagen um 70%, eine dauerhafte Ergebnisverbesserung um CHF 450 Mio. p.a., eine Erhöhung der netzweiten Trassenkapazität um 15–30%, eine um 50% erhöhte Verfügbarkeit der Sicherungsanlagen, die Senkung der Kollisionswahrscheinlichkeit beim Rangieren und an Baustellen um 90%, und hohe Datenfunk-Kapazität für Kunden mit einem Durchsatz >20 MBit/sec. Teilprojekte sind: Prozesse und Anforderungen; Traffic Management System; ETCS-Stellwerk; Lokalisierung, Connectivity und Security; und Automatic Train Operation.

Abbildung 8: smartrail4.0 Projekt (Kurzbeschreibung)

5 Fazit und Ausblick

Das „System Bahn“ ist ohne Software nicht mehr vorstellbar. Mit anderen Worten: Für die Bahnindustrie, die Eisenbahnverkehrsunternehmen und nicht zuletzt die Zulassungsbehörden unterscheidet sich das System Bahn immer weniger von „reinen“ IT-Produkten. Daraus ziehen wir für die der Studie zu Grunde liegenden Fragestellungen die folgenden Schlussfolgerungen.

Produktstrukturen: Software wird zunehmend die leistungsbestimmenden Eigenschaften übernehmen. Die Architektur, die Entwicklung und letztlich der Betrieb von Softwaresystemen sind damit entscheidend für den weiteren Erfolg des schienengebundenen Verkehrs. Natürlich werden auch künftig die Anforderungen an die Hardware der IT Systeme in der Bahn sehr hoch sein, aber durch die Integration verschiedener Geräte und Funktionalitäten, durch standardisierte Hardwarekomponenten und leistungsfähige Kommunikationsstrukturen und -netze ist eine Konzentration auf relativ wenige Rechnerstandorte möglich, die sich gegenüber der Anwendungssoftware wie eine Cloud darstellen.
In kleinerem Maßstab sind diese Systeme auch in den Zügen nutzbar. Solch eine homogene Architektur vereinfacht Entwicklung, Produktion und Wartung der Systeme und trägt zur Wirtschaftlichkeit bei gleichzeitiger Innovationsfähigkeit des Systems Bahn entscheidend bei.
Herausforderungen aus softwaretechnischer Sicht sind dabei das globale Variantenmanagement, Virtualisierung von Hardware, Verwendung von COTS-Komponenten, Portabilität von Software, sowie neue Fahr- und Leitfunktionen für autonomes Fahren und automatische Verkehrsregelung.

Standards: Flexibilität und Effizienz auf der Ebene der Anwendungssoftware sind nur möglich, wenn darunter eine stabile und einheitliche Plattform existiert. Solch eine Bahn-Middleware-Plattform sollte weltweit und über möglichst viele Bereiche der Branche, also auch LST und TCMS, gedacht werden. Die Luftfahrt-, die Automobil- und die Telekomindustrie mit den Plattformen IMA, AUTOSAR und 5G haben diesen Weg bereits beschritten und die Automatisierungsindustrie steckt mit der Industrie-4.0 Initiative gerade mittendrin. Firmenstandards, nationale Initiativen und sogar europäische Anstrengungen erscheinen im Zeichen weltweiter Märkte nicht hinreichend.
Im Zuge der europäischen Vereinheitlichung der Zulassungsverfahren ist zudem über eine Konsolidierung der verschiedenen Prozess-Normen zur Software-Entwicklung nachzudenken.

Methoden und Prozesse: Die Digitalisierung ergreift immer mehr Industrien und Wirtschaftsbereiche. Dies geschieht nicht nur an der Kundenschnittstelle, sondern auch in Entwicklung und Produktion. Die IT bestimmt damit auch immer mehr die Belange der Domänen und damit werden traditionelle domänenspezifische Arbeitsweisen obsolet und gleichen sich immer mehr an. Telekommunikation, Automotive, Automatisierung und auch der Bahnbereich gleichen sich im IT Bereich immer mehr aneinander an, und der Transfer von Vorgehensmodellen und Betriebsweisen, aber auch der von Menschen, nimmt zu. Dies bietet viel Potential für Synergien. Deutlich wird dies beispielsweise bei neuen Methoden wie der modellbasierten oder agilen Software-Entwicklung. Ein weiteres Beispiel ist in den sicherheitskritischen Bereichen die Vernetzung und dadurch zunehmende Sicherheitsprobleme (IT Security), die heutzutage an vielen Stellen noch nicht hinreichend Beachtung finden. Hier können domänenübergreifende Vorgehensweisen und der Transfer von Erfahrungswissen anderer Domänen dazu beitragen, Probleme zu vermeiden, bevor sie akut werden.

Ausbildung: Je mehr das System Bahn die Charakteristiken eines IT-Produkts annimmt, desto mehr gewinnt die IT Kompetenz als eigenständige Säule an Bedeutung. Die Frage nach „mehr IT Kompetenz für den Bahnspezialisten“ weicht zudem der Frage, „wieviel Domänenkompetenz für den IT Spezialisten“ gebraucht wird. Die Hersteller und Betreiber im Bahnbereich reihen sich ein in den Konkurrenzkampf um die besten IT-Spezialisten. Auch hier liegt ein Teil der Lösung in der technischen Architektur der Bahn IT. Klare Strukturen ermöglichen klare Zuordnungen. Hardware, Rechenzentren und Middle-

ware sind das Geschäft verschiedener IT Spezialisten; Anwendungssoftware für die Bahn hingegen erfordert ein hohes, wahrscheinlich sogar überwiegendes Wissen aus der Domäne Bahn.

6 Abbildungsverzeichnis

7 Quellenverzeichnis

Alle Links waren aktuell am Datum des Reports (geprüft am 12.9.2018).

3GPP (2018): 3GPP Progress on FRMCS in Rel-16, 2018, http://www.3gpp.org/news-events/partners-news/1964-frmcs_r16

Bechter, Markus (2015): AUTOSAR Adaptive Platform, 8th AUTOSAR Open Conference 2015-10-29, Tokyo, Japan, https://www.autosar.org/fileadmin/AOC/AOC_2015/Presentations/AUTOSAR_8AOC_Adaptive_Platform_Bechter.pdf

BMBF (2017): Projekt HASELNUSS. https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/haselnuss

Cawley, Oisín; Wang, Xiaofeng; Richardson, Ita (2010): Lean/Agile Software Development Methodologies in Regulated Environments – State of the Art. International Conference on Lean Enterprise Software and Systems, Springer (2010), pp 31-36. https://link.springer.com/chapter/10.1007%2F978-3-642-16416-3_4

CrESt (2017): CrESt - Modellbasierte Entwicklung kollaborativer eingebetteter Systeme. Project home page, https://crest.in.tum.de/

Cybersecurity Ventures (2018): Cybersecurity Jobs Report 2018-2021. https://cybersecurityventures.com/jobs/

Düpmeier, Frederik (2017): Entwurf einer neuen, regelbasierten Sicherungslogik unter Annahme der vollständigen Ortung aller Schienenfahrzeuge. In: Scientific Railway Signalling Symposium: Die Steuerung des Eisenbahnbetriebs der Zukunft, TU Darmstadt (2017). http://tuprints.ulb.tu-darmstadt.de/7403/7/SRSS_2017_Tagungsband_final2_korrigiert.pdf

ESG (Elektronik-System-Gesellschaft GmbH) (2018): On Design, Introduction and Operation Of Safety-critical Applications in a Data Center In the Railway System of Schweizerische Bundesbahnen SBB. https://smartrail40.ch/service/download.asp?path=download\downloads\Safety-critical%20Applications%20in%20Data%20Center%20in%20the%20Railway%20System%20SBB.pdf

ETSI (2018): ETSI Workshop, Developing the Future Radio for Rail Transport, 4-5 July 2018, Sophia Antipolis, France, https://www.etsi.org/news-events/events/1292-developing-the-future-radio-for-rail-transport

EUTPC (2016): European Truck Platooning Challenge. Project home page, https://www.eutruckplatooning.com/default.aspx

Fraunhofer FOKUS (2016): Netzinfrastrukturen für die Gigabitgesellschaft, Berlin 2016, https://www.bmvi.de/SharedDocs/DE/Anlage/Digitales/gigabit-studie.html

Fuchsen, Rudolf (2018): How to address Certification for Multi-Core Based IMA Plattforms. Professional article, Sysgo AG (2018). https://www.sysgo.com/services/knowledge-center/professional-articles/how-to-address-certification-for-multi-core-based-ima-plattforms/

Gary, Kevin; Enquobahrie, Andinet; Ibanez, Luis; Cheng, Patrick; Yaniv, Ziv; Cleary, Kevin; Kokoori, Shylaja; Muffih, Benjamin; Heidenreich, John (2011): Agile methods for open source safety- critical software, In: Software – Practice and Experience 41.9, Special Issue on Focus on Agile Software Development. https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.1075

Genc, Cengiz (2014): Die Bedeutung interdisziplinärer Qualifikationen in Eisenbahnprojekten. Signal + Draht 106 (2014). https://www.assystem-germany.com/fileadmin/user_upload/Fachartikel/FA_Interdisziplinare_Qualifikation_SignalDraht_2014.pdf

InterOperability Laboratory (2018): IOL INTACT® Protocol Testing Software (Product Page). https://www.iol.unh.edu/solutions/test-tools/intact

IPv6 TAHI Project (2017): IPv6 Ready Logo Phase-2 (project page). http://www.ipv6ready.org.cn/home/views/default/resource/logo/phase2-core/index.htm University of New Hampshire-InterOperability Lab Tahi Project. Ipv6 test specifications - core protocols. Technical report, IPv6 Forum, 2016 https://www.ipv6ready.org/?page=documents&tag=ipv6-core-protocols

IPv6 Forum (2018) (http://www.ipv6forum.com) IPv6 Ready Logo Program Approved List (2018) https://www.ipv6ready.org/db/index.php/public/?o=4

Jonsson, Henrik; Larsson, Stig; Punnekkat, Sasikumar (2012): Agile Practices in Regulated Railway Software Development. 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops. https://ieeexplore.ieee.org/abstract/document/6405469/

Leister, Hans (2017): ETCS und digitale Technologie für Stellwerke. Eisenbahn-Revue International 8-9/2017. https://mofair.de/wp-content/uploads/2017/08/Sonderdruck_ETCS_NeuPro_ERI_2017_8-9.pdf

Litzel, Nico (2018): Was ist OPC UA? https://www.bigdata-insider.de/was-ist-opc-ua-a-698144/

Marsch, Patrick (2018): Deutsche Bahn‘s Digital Rail Vision and its Implications on the Future Radio for Rail Transport, ETSI Workshop on Developing the Future Radio for Rail Transport, July 5th, 2018, https://docbox.etsi.org/Workshop/2018/201807_RT_WORKSHOP/S05_LONG_TERM_VISION/MARSCH_DEUTSCHE_BAHN-PUBLIC.pdf

NeGSt (2013): Neue Generation Signaltechnik. https://projekte.fir.de/negst/sites/projekte.fir.de.negst/files/ag5_05_negst_2300_projektierung_cots-produkte_20131216.pdf

openETCS (2012): Open Proofs Methodology for the European Train Control Onboard System https://itea3.org/project/openetcs.html

Safe4RAIL (2017): Requirements definition for Brake by Wire and safety concept. Safe4RAIL project Deliverable D4.2, Nov. 2017, https://safe4rail.eu/downloads/deliverables/Safe4RAIL-D4.2-Requirements-definition-PU-M13.pdf

SARTRE (2012): Safe Road Trains for the Environment. Project home page, http://web.archive.org/web/20160426135450/http://www.sartre-project.eu:80/en/Sidor/default.aspx

Schlingloff, Holger (2015): Towards a Curriculum for Model-Based Engineering of Embedded Systems. In: MBEES 2014 - 10. Workshop Modellbasierte Entwicklung eingebetteter Systeme. Dagstuhl, 5.-7. März 2014. https://www2.informatik.hu-berlin.de/~hs/Publikationen/2014_MBEES_Schlingloff_Towards-a-Curriculum-for-Model-Based-Engineering-of-Embedded-Systems.pdf

Schlingloff, Holger (2017): Specification and Verification of Collaborative Transport Robots.In: EITEC 2018 - 4th International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems. Cyber-Physical Systems Week, Porto, Apr. 2018. http://eitec.informatik.tu-muenchen.de/

Stecklina, Katja; Passeck, Clemens, (2015). Grundlagen sicherheitsgerichteter Software-Entwicklung. In: Cunningham, D. W., Hofstedt, P., Meer, K. & Schmitt, I. (Hrsg.), INFORMATIK 2015. Bonn: Gesellschaft für Informatik e.V.. (S. 1731-1734). https://dl.gi.de/bitstream/handle/20.500.12116/2159/1731.pdf?sequence=1&isAllowed=y

Turk, Dan; France, Robert; Rumpe, Bernhard (2002): Limitations of Agile Software Processes. Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002, May 26-30, Alghero, Italy, pg. 43-46, 2002. https://arxiv.org/abs/1409.6600

TU Darmstadt (2017): Die Steuerung des Eisenbahnbetriebs der Zukunft - Tagungsband des 1. Scientific Railway Signalling Symposiums, Darmstadt April 2017, http://tuprints.ulb.tu-darmstadt.de/7403/7/SRSS_2017_Tagungsband_final2_korrigiert.pdf

TU München (2018): Masterstudiengang Automotive Software Engineering (Software Engineering für Software im Automobil). https://www.in.tum.de/fuer-studieninteressierte/master-studiengaenge/automotive-software-engineering.html

8 Glossar

BEGRIFF	ERKLÄRUNG
5G	Kurzbezeichnung für den Mobilfunkstandard der fünften Generation
AGILITÄT	Agile Softwareentwicklung ist durch eine iterative und inkrementelle Vorgehensweise sowie durch sich selbst organisierende Arbeitsgruppen charakterisiert.
ALL-IP	Bezeichnung für den Prozess klassische Telefonnetze durch Netzwerke, die auf dem Internet Protocol (IP) beruhen, zu ersetzen.
AUTOSAR	AUTomotive Open System Architecture (AUTOSAR) ist eine weltweite Entwicklungspartnerschaft von Automobilherstellern, Zulieferern und anderen Unternehmen aus der Elektronik-, Halbleiter- und Softwareindustrie. Sie verfolgt den Zweck, eine offene und standardisierte Softwarearchitektur für elektronische Steuergeräte (ECUs) zu entwickeln und zu etablieren.
BLE	Bluetooth Low Energy (BLE) ist ein Funkstandard zur energieeffizienten Vernetzung von Geräten in einer Umgebung von bis ca. 10 Metern.
BSI	Bundesamt für Sicherheit in der Informationstechnik (BSI)
CAN	Controller Area Network (CAN) ist ein serielles Feldbussystem. Ein Feldbus ist ein Bussystem, das in einer Anlage Feldgeräte wie Messfühler (Sensoren) und Stellglieder (Aktoren) zwecks Kommunikation mit einem Automatisierungsgerät verbindet.
CD (AUTOSAR)	Complex Driver (CD), Software Komponente deren Definition außerhalb von AUTROSAR liegt (herstellerabhängig).
CENELEC	Comité Européen de Normalisation Électrotechnique (CENELEC) ist eine europäische Normierungsorganisation.
CLOUD COMPUTING	Cloud Computing beschreibt die Bereitstellung von IT-Infrastruktur wie beispielsweise Speicherplatz, Rechenleistung oder Anwendungssoftware als Dienstleistung über das Internet.
CONTINUOUS INTEGRATION	Continuos Integration bezeichnet den Prozess des fortlaufenden Zusammenfügens von Komponenten zu einer Anwendung.
COTS	Commercial off-the-shelf (COTS) bezeichnet Standardsoftware bzw. -hardware, die in großer Stückzahl produziert wird.
DMI	Driver Machine Interface (DMI) ist eine Komponente im ETCS. Das DMI ist die Schnittstelle zwischen dem Lokführer und dem ERTMS/ETCS-System (Führerstandsanzeige).
DSL	Domain specific language (DSL) ist eine formale Sprache, die zur Interaktion zwischen Menschen und Computern für ein bestimmtes Problemfeld (die sogenannte Domäne) entworfen und implementiert wird.
DSTW	Digitales Stellwerk

ECU	Electronic Control Unit (ECU) sind elektronische Module, die überwiegend an Orten eingebaut werden, an denen etwas gesteuert oder geregelt werden muss (Steuergeräte).
EDGE COMPUTING	Beim Edge Computing werden Computer-Anwendungen, Daten und Dienste von zentralen Knoten (Rechenzentren) weg zu den äußeren Rändern eines Netzwerks verlagert (im Gegensatz zur zentralen Verarbeitung beim Cloud Computing).
ERP	Enterprise-Resource-Planning (ERP) bezeichnet die unternehmerische Aufgabe, Ressourcen wie Kapital, Personal, Betriebsmittel, Material und Informations- und Kommunikationstechnik im Sinne des Unternehmenszwecks rechtzeitig und bedarfsgerecht zu planen und zu steuern.
ESTW	Elektronisches Stellwerk
ETCS	European Train Control System (ETCS) ist ein europäisches Zugbeeinflussungssystem und grundlegender Bestandteil des europäischen Eisenbahnverkehrsleitsystems (ERTMS).
ETHERNET	Eine Spezifikation der Software und Hardware für kabelgebundene Kommunikationsnetze.
EVC	European Vital Computer (EVC), spezielles Steuergerät an Bord eines Zuges (definiert im Rahmen von ETCS).
FLEXRAY	FlexRay ist ein serielles, deterministisches und fehlertolerantes Feldbussystem für den Einsatz im Automobil, welches höhere Datenraten als herkömmliche CAN-Bus-Systeme ermöglicht.
FPGA	Field Programmable Gate Array (FPGA) ist ein integrierter Schaltkreis, der es erlaubt logische Schaltungen zu laden.
FRMCS	Future Railway Mobile Communication System (FRMCS) ist eine Projektbezeichnung für bahnspezifische Anpassungen des LTE Standards.
GSM-R	Global System for Mobile Communications – Railway (GSM-R) ist eine für den Bahnbereich entwickelte Erweiterung des Mobilfunkstandards GSM.
HOCHSPRACHE	Hochsprache (höhere Programmiersprache) ist eine Programmiersprache zur Abfassung eines Computerprogramms, die in Abstraktion und Komplexität von der Ebene der Maschinensprachen deutlich entfernt ist, so dass mehr und komplexere logische Zusammenhänge mit weniger Text ausgedrückt werden können.
HYPERVISOR	Hypervisor bezeichnet eine abstrahierende Schicht zwischen tatsächlich vorhandener Hardware (bzw. dem darauf laufenden Betriebssystem) und weiteren zu installierenden Betriebssystemen.
IMA	Integrated Modular Avionics (IMA), eine flugtaugliche, modulare Elektronikeinheit aus standardisierten Komponenten und Schnittstellen in Hard- und Software zur Kommunikation zwischen den verschiedenen Systemen in einem Luftfahrzeug.

INDUSTRIE 4.0	Industrie 4.0 ist die Bezeichnung für ein Zukunftsprojekt zur umfassenden Digitalisierung der industriellen Produktion.
IOT	Das Internet der Dinge (IoT) bezeichnet die Verknüpfung eindeutig identifizierbarer physischer Objekte (things) mit einer virtuellen Repräsentation in einer Internet-ähnlichen Struktur.
IP, IPV6	Internet Protocol (IP) Version 6 (IPV6), die aktuelle Version des Internet Protokolls.
IPC	Industrie-PC (IPC) ist ein Computer, der für Aufgaben im industriellen Bereich eingesetzt wird. Im engeren Sinn geht es dabei um Rechner, die einem IBM-kompatiblen Personal Computer ähneln und insbesondere mit Software für solche Geräte betrieben werden können.
LRU	Line-Replaceable-Unit (LRU) ist ein Bauteil eines Luftfahrzeugs, das im Rahmen einer Wartung oder Instandsetzung vor Ort ausgewechselt werden kann.
LST	Leit- und Sicherungstechnik (LST) bei der Sicherung von Zugfahrten
LTE	Long Term Evolution (LTE) ist ein Mobilfunkstandard der vierten Generation.
M2M	Machine-to-Machine (M2M) steht für den automatisierten Informationsaustausch zwischen Endgeräten wie Maschinen, Automaten, Fahrzeugen oder Containern untereinander oder mit einer zentralen Leitstelle, unter Nutzung des Internets und den verschiedenen Zugangsnetzen, wie dem Mobilfunknetz.
MIDDLEWARE	Eine Middleware ist eine Schicht in der Architektur von Informatiksystemen zwischen Betriebssystem und Anwendungen. Die Middleware unterstützt die Kommunikation zwischen Prozessen und erlaubt somit eine schnelle Applikationsentwicklung.
OBSOLESZENZMANAGEMENT	Management der Tatsache, dass Produkte und Komponenten auf Grund der Herstellungsweise, Abnutzung oder technischen Weiterentwicklung veralten, unbrauchbar oder überflüssig werden. Zum Obsoleszenzmanagement gehört die Bevorratung und Beschaffung von Ersatzteilen sowie die Planung und Entwicklung von Alternativen.
OBU	On-Board Unit (OBU) bezeichnet eine elektronische, meistens computerbasierte Fahrzeugeinrichtung, die wichtige interne Steuerfunktionen im Fahrzeug übernimmt.
OMG	Object Management Group (OMG) ist ein internationales Konsortium, das Standards für die objektorientierte Programmierung entwickelt.
OPC-UA	Open Platform Communications - Unified Architecture (OPC-UA) ist ein industrielles Kommunikationsprotokoll, um Maschinendaten (Regelgrößen, Messwerte, Parameter usw.) nicht nur zu transportieren, sondern auch maschinenlesbar semantisch zu beschreiben.
OSI	Open Systems Interconnection (OSI) Model ist ein ISO Standard bzw. Referenzmodell für Netzwerkprotokolle als Schichtenarchitektur.

PKI	Public-Key-Infrastruktur (PKI) bezeichnet in der Kryptologie ein System, das digitale Zertifikate ausstellen, verteilen und prüfen kann. Die innerhalb einer PKI ausgestellten Zertifikate werden zur Absicherung rechnergestützter Kommunikation verwendet.
POSIX	Portable Operating System Interface (POSIX) ist eine standardisierte Programmi- erschnittstelle zwischen Anwendungssoftware und Betriebssystem.
PZB/LZB	Punktförmige und linienförmige Zugbeeinflussung (PZB/LZB), Systeme zur Überwachung und Beeinflussung schienengebundener Fahrzeuge.
RBC	Radio Block Center (RBC) ist eine Komponente des ETCS, welche die Züge in einem bestimmten Bereich führt und überwacht.
SCRUM	Scrum ist ein Vorgehensmodell zur agilen Softwareentwicklung. In Scrum werden die Kunden-Anforderungen in einem „Product Backlog“ erfasst und inkrementell in sogenannten „Sprint“-Intervallen umgesetzt.
SDN	Software-defined Networking (SDN) ist ein Ansatz bei dem bestimmte Funktionsebenen eines Netzwerks als virtuelle Dienste dargestellt werden.
SOA	Serviceorientierte Architektur (SOA)) ist ein Architekturmuster der Informationstechnik aus dem Bereich der verteilten Systeme, um Dienste von IT-Systemen zu strukturieren und zu nutzen.
SPS	Speicherprogrammierbare Steuerung (SPS) ist ein Gerät, das zur Steuerung oder Regelung einer Maschine oder Anlage eingesetzt und auf digitaler Basis programmiert wird.
SYSML	Systems Modeling Language (SysML) ist eine grafische Modellierungssprache zur Entwicklung komplexer Systeme, die auf UML basiert.
TCMS	Train Control and Management System (TCMS) ist ein verteiltes computerbasiertes Kontrollsystem für Züge.
TPM	Trusted Platform Module (TPM) ist ein Chip, der einen Computer oder ähnliche Geräte um grundlegende Sicherheitsfunktionen erweitert.
TSN	Time-Sensitive Networking (TSN) bezeichnet eine Reihe von Standards, die Mechanismen zur Übertragung von Daten mit sehr geringer Übertragungslatenz und hoher Verfügbarkeit über Ethernet-Netzwerke definieren.
UML	Unified Modeling Language (UML) ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software. Der aktuelle Stand UML 2.4.1 ist als ISO/IEC 19505 standardisiert.
VIRTUALISIERUNG	Virtualisierung bezeichnet die Nachbildung eines Hard- oder Software-Objekts durch ein sich ähnliche verhaltendes Objekt mit Hilfe einer Abstraktionsschicht.

X509	X.509 ist ein Standard zum Erstellen digitaler Zertifikate.
ZIGBEE	ZigBee ist eine Spezifikation für drahtlose Netzwerke mit geringem Datenaufkommen, wie beispielsweise Hausautomation, Sensornetzwerke, Lichttechnik. Der Schwerpunkt von ZigBee liegt in Netzwerken mit kurzer Reichweite (bis 100 Meter).

9 Anhang: Fragebogen

Umfeld

(1) Was erwarten sie für die nächsten 2/5/10 Jahren an Veränderungen? Warum?

Produktstrukturen (Architektur)

(1) Welche Herausforderungen gilt es bei der Entwicklung und Integration von künftigen Systemstrukturen zu bewältigen?
(2) Welche Hardware-Entwicklungen und Trends sind für Sie von Bedeutung?
(3) Welche Software-Technologien werden künftig gebraucht?

Standards

(1) Welche Sicherheitskritikalitäten (Safety & Security) treten auf, wie wird deren Behandlung in den Standards und Normen geregelt? Gibt es Ansätze um die Themen Safety und Security gemeinsam anzugehen?
(2) Wie sind die Zulassungsprozesse, welche Probleme treten dabei auf?
(3) Wofür fehlen übergreifende Regelungen (Standards, Normen, Gesetze, Verordnungen)?

Methoden und Prozesse (Entwicklung)

(1) Wie gehen Sie mit der zunehmenden Komplexität (bzgl. Funktionen, Varianten usw.) um?
(2) Welche Software-Engineering-Methoden und –Werkzeuge haben sich in Ihrem Bereich bewährt?
(3) Wo sehen Sie den meisten Handlungsbedarf / das größte Verbesserungspotential innerhalb der Systementwicklung?
(4) Welche Software-Engineering-Methoden und aktuellen Trends betrachten Sie als besonders aussichtsreich, bzw. welche schätzen Sie als nicht zielführend ein?

Ausbildung

(1) Wo sind Ihrer Meinung nach Defizite in unserem jetzigen Ausbildungssystem (für Ihren Tätigkeitsbereich)?
(2) Wie lange dauert es, dass aus einem Absolventen ein domänenerfahrener Softwareentwickler wird?
(3) Gibt es firmeninterne Weiterbildungsprogramme für Softwareentwickler? Was wird dort vermittelt?