QUALITY ENGINEERING FÜR DAS INTERNET DER DINGE (CPIOT) LEHRPLAN ZUM BASISKURS

ASQF/GTB CERTIFIED PROFESSIONAL FOR IOT
FOUNDATION LEVEL

Lehrplan Version 1.1

2019

Copyright und Nutzungsrechte

Autoren

Aida Boukhris (Friedrich-Alexander-Universität Erlangen-Nürnberg), Vera Gebhardt (tecmata GmbH), Alexander Gladisch (T-Systems MMS GmbH), Daniel Hummel (IT-P GmbH), Günter Jung (imbus AG), Ralf Mack (Atos Information Technology GmbH), Matthias Pruksch (sepp.med GmbH), Axel Rennoch (Fraunhofer-Institut für Offene Kommunikationssysteme), Alfred Richter (DB Systel GmbH), Nils Röttger (imbus AG), Ina Schieferdecker (Fraunhofer-Institut für Offene Kommunikationssysteme / TU Berlin), Jessica Schiffmann, Günter Schneider (Sulzer GmbH), Armin Metzger (GTB)

Reviewer

Jan Markus Giesen (Brunel Car Synergies GmbH), Thomas Haase (T-Systems MMS GmbH), Anne Kramer (sepp.med GmbH), Helmut Pichler (Nagarro GmbH), Frederik Teichert (HILSTER Testing Solutions GmbH)

Änderungsübersicht

Version	Datum	Autor	Bemerkung
1.0	21.12.2017	Autoren und Reviewer	Erste freigegebene Version
1.1	14.11.2018	Autoren und Reviewer	Zweite freigegebene Version
1.1	14.02.2019	Reviewanpassungen	

Inhaltsverzeichnis

Liste der Lernziele

IoT-QE LZ 1 (K1) Wissen, was Internet der Dinge bedeutet [10]
IoT-QE LZ 2 (K2) Konstruktives und analytisches Quality Engineering im Kontext IoT erklären können [15]
IoT-QE LZ 3 (K1) Wissen, dass Quality Engineering eine hohe Relevanz für das Internet der Dinge hat [5]
IoT-QE LZ 4 (K2) Die Besonderheiten des IoT und die damit verbundenen spezifischen Herausforderungen für das Quality Engineering erklären können [30]
IoT-QE LZ 5 (K3) Auswirkungen der datengetriebenen IoT Geschäftsmodelle beurteilen können [60]
IoT-QE LZ 6 (K2) Relevanz und Schwerpunkte von Qualitätsmerkmalen für IoT im Überblick erklären können [10]
IoT-QE LZ 7 (K2) Die Relevanz der Qualitätsmerkmale auch für den Betrieb erklären können [15]
IoT-QE LZ 8 (K1) Die Bedeutung von Standards und regulatorische Anforderungen kennen [5]
IoT-QE LZ 9 (K1) Funktionale Qualitätsmerkmale kennen [5]
IoT-QE LZ 10 (K2) Die Sicherheitsherausforderungen (sowohl Security als auch Safety) bei IoT Systemen erklären können. [15]
IoT-QE LZ 11 (K3) Eine Analyse der Auswirkungen der Qualitätsmerkmale IT-Sicherheit und funktionale Sicherheit auf Konstruktives QE vornehmen können [60]
IoT-QE LZ 12 (K2) Die Anforderungen an Interoperabilität für IoT Systeme erklären können [10]
IoT-QE LZ 13 (K1) Die für IoT Systeme wesentlichen Qualitätsmerkmale Robustheit und Resilienz kennen [10]
IoT-QE LZ 14 (K2) Die Anforderungen an Wartbarkeit und Übertragbarkeit für IoT Systeme erklären können [15]
IoT-QE LZ 15 (K2) Die besonderen Herausforderungen an das Qualitätsmerkmal Performanz (Zeitverhalten und Verbrauchsverhalten) für IoT Systeme erklären können [10]
IoT-QE LZ 16 (K2) Die Relevanz Ethischer Aspekte für IoT erklären können [10]
IoT-QE LZ 17 (K2) Das Spannungsfeld von IT-Sicherheit und funktionaler Sicherheit erklären können [15]
IoT-QE LZ 18 (K2) Das Spannungsfeld von Usability, Wartbarkeit und IT-Sicherheit erklären können [15]
IoT-QE LZ 19 (K2) Das Spannungsfeld von Resilienz, Robustheit und Performanz erklären können [15]
IoT-QE LZ 20 (K2) Das Spannungsfeld von Konnektivität, Interoperabilität und IT-Sicherheit erklären können [15]
IoT-QE LZ 21 (K3) Die Qualitätsmerkmale eines Systems bewerten und daraus Anforderungen an das IoT System ableiten können [50]
IoT-QE LZ 22 (K1) Wissen was eine Architektur für IoT geeignet macht [15]
IoT-QE LZ 23 (K1) Ausgewählte IoT Referenzarchitekturen kennen [10]
IoT-QE LZ 24 (K2) Die Elemente von IoT Architekturen mit Hilfe von AIOTI erklären können [15]
IoT-QE LZ 25 (K2) Die Schichten von IoT Architekturen am Beispiel AIOTI erklären können [15]
IoT-QE LZ 26 (K2) Die Funktionen der Schichten in IoT Architekturen am Beispiel AIOTI erklären können [15]
IoT-QE LZ 27 (K2) Den spezifischen Einfluss der Daten auf IoT Architekturen erklären können [15]
IoT-QE LZ 28 (K1) RAMI als spezifische IoT Architektur kennen [10]
IoT-QE LZ 29 (K2) Die Schichten von IoT Architekturen am Beispiel RAMI 4.0 erklären können [15]
IoT-QE LZ 30 (K1) oneM2M als spezifische IoT Architektur kennen [10]
IoT-QE LZ 31 (K2) Die Schichten von IoT Architekturen am Beispiel OneM2M erklären können [15]
IoT-QE LZ 32 (K3) Eine IoT Referenzarchitektur auf eine spezifische IoT Systemarchitektur abbilden können [30]

IoT-QE LZ 33 (K1) Best Practices in IoT kennen [10]
IoT-QE LZ 34 (K1) Die Vorteile agiler Methoden kennen [10]
IoT-QE LZ 35 (K1) Die Vorteile automatisierter Methoden kennen [10]
IoT-QE LZ 36 (K2) DevOps für IoT erklären können [15]
IoT-QE LZ 37 (K2) Die Bedeutung von Produkt- und Systemvarianten für IoT erklären können [15]
IoT-QE LZ 38 (K2) Die Bedeutung des Quality Engineering für die Betriebsphase bei IoT Systemen erklären können [15]
IoT-QE LZ 39 (K1) Die Notwendigkeit von Monitoring im Betrieb von IoT Systemen kennen [verteilt auf Kapitel]
IoT-QE LZ 40 (K2) Die Herausforderungen verteilter Tests für IoT Systeme erklären können [verteilt auf Kapitel]
IoT-QE LZ 41 (K2) Die besonderen Herausforderungen beim Testen von IoT Lösungen wie ihre Offenheit, Verteiltheit, Dynamik, Skalierung und Varianz erläutern können [10]
IoT-QE LZ 42 (K2) Für IoT spezifische Testvorgehen und Teststufen erläutern können [20]
IoT-QE LZ 43 (K3) Testziele für IoT definieren und deren Priorisierung durchführen können [30]
IoT-QE LZ 44 (K3) Risikobasierte Priorisierung von Testzielen durchführen können [30]
IoT-QE LZ 45 (K2) Die Besonderheiten beim IoT Testen benennen und Beispiele für IoT Tests auf verschiedenen Ebenen erläutern können [10]
IoT-QE LZ 46 (K2) Die Notwendigkeit der Testautomatisierung für den IoT Test erläutern können [15]
IoT-QE LZ 47 (K2) IoT Testarchitekturen und typische IoT Testobjekte erläutern können [15]
IoT-QE LZ 48 (K2) Wesentliche Aspekte der IoT Testarchitektur erläutern können [15]
IoT-QE LZ 49 (K2) Nutzbarkeit und Grenzen klassischer Testmethoden für IoT Systeme erläutern können [20]
IoT-QE LZ 50 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Sicherheit und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [20]
IoT-QE LZ 51 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Interoperabilität und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [15]
IoT-QE LZ 52 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Performanz und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [20]
IoT-QE LZ 53 (K2) Die Herausforderungen bei der Prüfung auf Konformität und Zertifizierung erklären können [20]
IoT-QE LZ 54 (K2) Die Bedeutung der im IoT Kontext verknüpften Lebenszyklen für das QE verstehen [15]
IoT-QE LZ 55 (K2) Die interdisziplinäre Natur des IoT-Lebenszyklus verstehen [15]
IoT-QE LZ 56 (K2) Bedeutung von Drittbeteiligten im IoT Kontext verstehen [15]

0 Einführung

Qualifizierungsschema

Der vorliegende Lehrplan bietet der Industrie Hilfe in Form von Methoden, Leitlinien zur Qualitätssicherung und Absicherung von IoT Lösungen in Form von Qualifizierungsschemata und einem Glossar als De-Fakto-Standard an. Er wurde von Experten im Rahmen einer Arbeitsgruppe in Kooperation zwischen ASQF e.V. und GTB e.V. auf Basis von Best Practices erstellt und wird von dieser Arbeitsgruppe kontinuierlich gepflegt.

Das Thema Internet der Dinge (Internet of Things, IoT) eröffnet Unternehmen und Menschen völlig neue Möglichkeiten, Abläufe zu vereinfachen, Erkenntnisse zu gewinnen und Dienste anzubieten. Gleichzeitig stellt es jedoch Gesellschaft und Industrie vor große Herausforderungen. IoT ist ein Gebiet, welches derzeit noch im Fluss ist. Erste Standards entwickeln sich, aber es existiert noch nicht DER Standard. Auch die Integration der IoT Aspekte in benachbarte Themengebiete wie Industrie 4.0, Big Data, etc. ist noch in einer frühen Phase. Der Lehrplan will daher einen Überblick in die Prinzipien und Herangehensweisen entlang richtungsweisender Standardisierungsinitiativen und den aktuellen Stand der Entwicklung geben.

Die Durchführung von Trainings ist nur von durch ASQF bzw. GTB akkreditierten Trainings-unternehmen erlaubt. Die Teilnehmer werden von einer vom ASQF bzw. GTB akkreditierten Zertifizierungsstelle geprüft. Bei erfolgreicher Prüfung wird den Teilnehmern die Qualifikation „Certified Professional for IoT, Foundation Level“ durch die Zertifizierungsstelle bescheinigt.

Business Outcomes

Der Geschäftsnutzen (Business Outcomes) durch die Teilnahme an einer Ausbildung basierend auf vorliegendem Lehrplan für den Teilnehmer und dessen Organisation stellt sich wie folgt dar:

IoT-QE_BO01_Sensibilisierung: Sensibilisierung: Verständnis der besonderen Herausforderungen des Quality Engineering im Kontext von IoT.

IoT-QE_BO02_Standards: Reibungslose Zusammenarbeit in und mit IoT Teams durch Kenntnis von Standards und dem gemeinsamen Glossar.

IoT-QE_BO03_Expertise: Anwenden und Meistern des Quality Engineering im IoT Kontext durch Übertragen „klassischer“ QE Expertisen als auch Erlangen spezifischer Expertisen des IoT QE.

IoT-QE_BO04_QE-in-der-Organisation: Verbesserung des Quality Engineering im IoT Kontext in der Organisation durch Support der IoT Teams und Transfer von QE Expertise in die Organisation.

IoT-QE_BO05_Persönliche-Entwicklung: Persönliche Weiterentwicklung des Trainingsteilnehmers durch Expertise in einem anspruchsvollen Zukunftsthema.

Lernziele und kognitive Stufen des Wissens

Aus den Business Outcomes ergeben sich die im Lehrplan behandelten Lernziele (Learning Objectives). Jeder Abschnitt des Lehrplans enthält Lernziele mit einer zugeordneten kognitiven Stufe. In Anhang C ist eine Beschreibung der Level zu finden.

In Basislevel-Kursen finden üblicherweise folgende Stufen Anwendung:

K1: kennen

K2: verstehen

K3: anwenden.

Der Level hat auch Einfluss auf die Lehrdauer und die Art der möglichen Prüfungsfragen.

1 Motivation [130]

Begriffe

Internet of Things, Internet der Dinge (IoT)	Infrastruktur von miteinander verbundenen Entitäten, Personen, Systemen und Informationsquellen zusammen mit Diensten, welche Informationen der physikalischen Welt und virtuellen Welt verarbeitet und darauf reagiert.
Digital Twin	Die digitale Abbildung eines physischen Objekts, eines Prozesses oder Systems in der virtuellen Welt. Diese Abbildung enthält sowohl den Aufbau als auch die Dynamik eines IoT Objekts und zwar über dessen gesamten Lebenszyklus.
Konstruktives Quality Engineering	Ganzheitliches Ergreifen vorbeugender Maßnahmen zur Vermeidung, etwas Falsches, etwas auf ungeeignete Weise oder etwas mit mangelnder Sorgfalt zu entwickeln

1.1 Quality Engineering für das Internet der Dinge: Was ist das? [30]

- Was ist das Internet der Dinge? (10 Min)

IoT-QE LZ 1 (K1) Wissen, was Internet der Dinge bedeutet [10]

Unter IoT wird eine Infrastruktur verstanden, die aus von miteinander verbundenen Entitäten, Personen, Systemen und Informationsquellen zusammen mit Diensten, welche Informationen der physikalischen Welt und virtuellen Welt verarbeitet und darauf reagiert.

Es handelt sich dabei weder um eine neue noch um eine fest definierte Technologie bzw. exakte Systemdefinition. Internet der Dinge bedeutet, dass physische Objekte zunehmend durch ihre digitalen Repräsentationen („Digital Twin“) in digitalen Umgebungen repräsentiert und so orchestriert oder auch integriert werden können. Dies verändert sowohl die technologische Basis als auch die Anwendungsszenarien von softwarebasierten Lösungen.

Das Thema Internet der Dinge (Internet of Things, IoT) stellt Gesellschaft und Industrie vor große Herausforderungen. Alles ist in einer sich ändernden Infrastruktur integriert. Es wirken technische Szenarien wie Architekturen und Komponenten sowie gesellschaftlichen Rahmenbedingungen (z.B. Gesetze zum Schutz privater Daten und ethische Fragen) aufeinander ein.

Um diese Herausforderungen zu meistern und sich auf neue Methoden, Abläufe und Bedingungen einstellen zu können, müssen die Unternehmen einen notwendigen Kulturwandel vollziehen. Beispiele hierfür ist die Entstehung einer Kultur sicherheitsbewussten Denkens (im Sinne der IT-Sicherheit). Auch ethische Richtlinien müssen massiv beachtet werden. Daraus folgt zwingend ein notwendiges Umdenken bzgl. der Prioritäten und Ausprägungen von Techniken, Qualitätskriterien und Prozessen.

- Was bedeutet Quality Engineering für IoT? (20 Min)

IoT-QE LZ 2 (K2) Konstruktives und analytisches Quality Engineering im Kontext IoT erklären können [15]

Wirksame Qualitätsplanung und -sicherung verhindert oder erkennt frühzeitig Fehler im Entwicklungsprozess, in der Produktion, vor/nach Betrieb und trägt somit maßgeblich zur Akzeptanz des Produkts und letztlich zum wirtschaftlichen Erfolg von IoT Produkten und Dienstleistungen bei.

Generell gilt die Devise: „Vorbeugen ist besser als Heilen“. Konstruktive Qualitätsarbeit zielt darauf, Fehler von Anfang an zu vermeiden. Durch Orientierung an Best Practices und Standards in den Entwicklungs- und Produktionsabläufen (Prozesse) sowie der Arbeitsumgebung (Ausstattung) werden Erfahrungen des Marktes nutzbar gemacht. Die Planung von Qualitätsprüfungen und

Berücksichtigung von Testbarkeit während der Entwicklung und Fertigung, sowie die Planung des späteren Betriebs hinsichtlich der Aufrechterhaltung der geforderten Service Levels gehört ebenfalls zur konstruktiven Qualitätsarbeit.

Das konstruktive Quality Engineering für IoT Systeme baut auf dem grundlegenden Verständnis der Architektur und der erforderlichen Qualitätsmerkmale von IoT Systemen auf und unterstützt die analytische Qualitätssicherung durch eine optimale Planung von Qualitätsmaßnahmen.

Analytische Qualitätsarbeit dient der frühzeitigen Aufdeckung von Fehlern. Dabei kommen statische Verfahren (z.B. die Nutzung von Review-Techniken zur frühen Fehleraufdeckung oder Modellprüfungen) und die dynamische Qualitätssicherung durch Tests (funktionale Tests, Lasttests, Akzeptanztests, Gebrauchstauglichkeitstests, Sicherheitstests, Penetrationstests, etc.) zum Einsatz.

Das analytische Quality Engineering für IoT Systeme erfordert intensive Planungsphasen und kreative Methodenauswahl zur Sicherstellung der frühstmöglichen Fehleraufdeckung. Die Planung muss stetig auf Eignung und Anpassungsbedarf geprüft werden und Anpassungen sind zwingend durchzuführen und zu überwachen.

IoT-QE LZ 3 (K1) Wissen, dass Quality Engineering eine hohe Relevanz für das Internet der Dinge hat [5]

Die Konstruktion eines IoT Systems erfordert von Anfang an ein sehr hohes Bewusstsein für Quality Engineering und somit eine vorausschauende Vorgehensweise, welche die Erfüllung von Qualitätskriterien sicherstellen soll. Quality Engineering für IoT bedeutet insbesondere, Qualitätsaspekte im gesamten Lebenszyklus eines Produktes umfassend zu berücksichtigen. Dieser Lebenszyklus reicht von der Konzeption über die Typentwicklung, den Test bis hin zur Serienfertigung und Qualitätskontrolle. In der Betriebsphase spielen Aspekte der Funktions- oder Dienstüberwachung eine Rolle. Selbst die Außerbetriebnahme kann ein Thema für das Quality Engineering werden, wenn die Zugänglichkeit, der Datenschutz oder Umweltfragen eine Rolle spielen.

1.2 Besonderheiten des QE4IoT [90]

- Was verändert das Internet der Dinge für das Quality Engineering? (30 Min)

IoT-QE LZ 4 (K2) Die Besonderheiten des IoT und die damit verbundenen spezifischen Herausforderungen für das Quality Engineering erklären können [30]

IoT-Systeme sind gekennzeichnet durch:

- die Kombination sehr heterogener Hard- und Software und die große Zahl von miteinander interagierenden Komponenten,
- lokale Vernetzung im Intranet und globale Vernetzung über das Internet,
- die Vielfalt von unterschiedlichen Technologien und Protokollen auf Anwendungsebene und bei Verbindungen,
- mobile Endgeräte und Sensorik/Aktuatorik mit unterschiedlichsten Hardware-Ressourcen und Generationen von Betriebssystemen,
- die Erfassung, Kommunikation und Verarbeitung von volatilen, heterogenen und großen Datenmengen,
- die Dynamik der Strukturen und Komponenten (als „lebendes“ und offenes System) und
- die nötigen horizontalen und vertikalen Integrationen unter Nutzung von verschiedenen Plattformen für das Edge- und Cloud-Computing.

Hieraus entstehen besondere Herausforderungen für das Quality Engineering in Bezug auf:

- Interdisziplinäres Arbeiten – unterschiedliche technische Domänen erfordern eine Zusammenarbeit von Spezialisten mit unterschiedlichen Vorgehensweisen und Kenntnissen. Diese sprechen oft nicht die gleiche (technische) Sprache.
- Große Gerätevielfalt und Vielfalt an Varianten – die Konformität zu Standards oder Schnittstellenvereinbarungen und die Auswahl der sinnvollen Varianten für den Business Case muss sichergestellt werden.
- Komplexe Betriebsszenarien – Testsituationen können die spätere Betriebssituation oft nur annähernd abdecken.
- Fehlende Zugänglichkeit der Geräte – Endgeräte sind häufig nicht einer Analyse oder Fehlerbeseitigung zugänglich.
- Anfälligkeit für Angriffe – vernetzte, über das Internet erreichbare Systeme sind prinzipiell anfällig für Angriffe und müssen entsprechend über den gesamten Produktlebenszyklus geschützt werden. Da der Austausch und die Verarbeitung von Daten im Vordergrund stehen, hat Datensicherheit eine zentrale Bedeutung, sowohl in Bezug auf jede einzelne Komponente, als auch in Bezug auf das Zusammenspiel auf den verschiedenen Ebenen und das Ende-zu-Ende-Verhalten.
- Stark konkurrierende Qualitätsanforderungen – Qualitätsanforderungen stehen oftmals im Widerspruch zueinander, so dass beispielsweise eine hohe Sicherheit zu Lasten der Performanz oder des einfachen Zugangs geht.

- IoT Business ist Datenbusiness – Aspekte für das QE (60 Min)

IoT-QE LZ 5 (K3) Auswirkungen der datengetriebenen IoT Geschäftsmodelle beurteilen können [60]

IoT Systeme ermöglichen neuartige und durch das Internet weitreichendere datengetriebene Geschäftsmodelle. Die essentielle Wertschöpfung entsteht durch die Gewinnung von Handlungsoptionen auf Basis selbst erhobener Daten bzw. ergänzender Daten von Drittanbietern. Beispiele dafür sind:

- Information und Kontrolle in Echtzeit (Dashboard, Tracing),
- Auswertung der Historie mit Anwendung auf Vorhersagemodelle (Predictive Maintenance, Machine Learning) oder
- hocheffiziente Verwaltung von Ressourcen (On Demand, Shareconomy) auf Basis von Nutzungsdaten.

Aus den vorhandenen Daten werden Informationen extrahiert, Erkenntnisse gewonnen und Handlungsoptionen ermittelt. Diese werden umgesetzt, wodurch neue Daten zur Verfügung stehen, die wiederum zu neuen Erkenntnissen führen – siehe auch Abbildung 1. Die Services, Prozesse und Software zur Sammlung und Verarbeitung der Daten sowie zur Aufbereitung der Ergebnisse bedürfen dazu einer kontinuierlichen Optimierung.

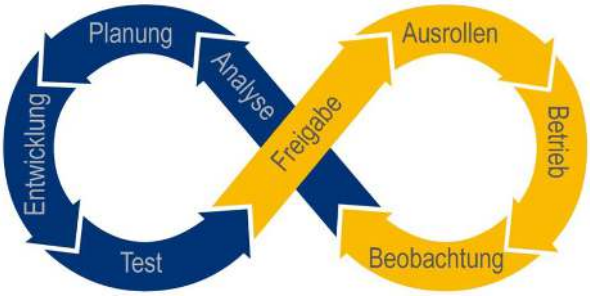

Abbildung 1: Datengetriebener DevOps-Prozess

Durch die Nutzung des Internets als Kommunikationsplattform können physische Komponenten in ein Netzwerk von Sensoren, Aktoren und zentralen sowie verteilten Systemen eingebunden werden. So stehen Daten aus der realen Welt zur Nutzung in Geschäftsprozessen zunehmend zeitnaher, umfangreicher, vollständiger, umfassender, aber auch heterogener zur Verfügung.

Die verfügbaren Komponenten sowie die Art und Qualität der durch sie bereit gestellten Daten, wird eine hohe Dynamik aufweisen und nicht in jedem Fall vollständig planbar sein. Die Analyse dieser Daten kann zu Erkenntnissen führen, die vorher nicht absehbar waren. Aspekte der Datenaggregation, der Filterung und der Schutzziele personenbezogener, geschäfts- und sicherheitskritischer Daten müssen bei der Planung und dem Betrieb von IoT Anwendungen beachtet werden.

Daten erhalten somit eine herausragende Bedeutung im IoT-Business. Eine kontinuierliche Anpassung der Produkte und Lösungen, über den gesamten Lebenszyklus inklusive des Betriebs, wird erforderlich.

Übung zur Datenorientierung in IoT

Die Kursteilnehmer bilden Gruppen zu je etwa 3-4 Mitgliedern. Jede dieser Gruppen priorisiert die wichtigsten Qualitätsmerkmale für ihren Aufgabenbereich und definiert einen Zielkorridor. Der erreichte Stand wird im Plenum vorgestellt und kurz diskutiert (30 min). Ziel der Übung ist es, ein gemeinsames Grundverständnis und eine gemeinsame Sprache in Bezug auf Qualitätsmerkmale zu erarbeiten, bevor das Thema Qualitätsmerkmale in Kapitel 2 vertieft wird.

1.3 Beispiel „Smart Home“ [10]

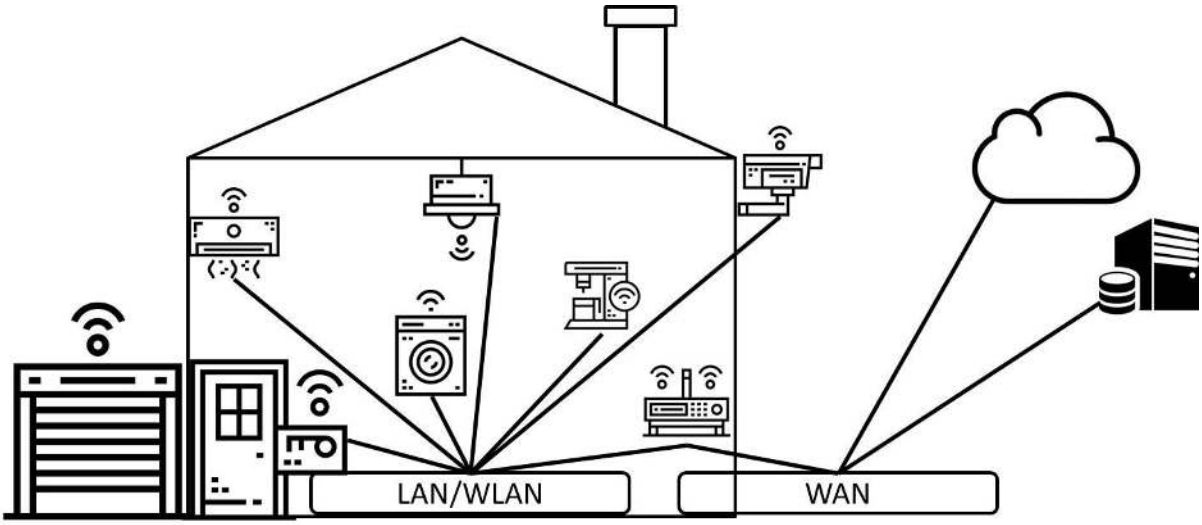

Abbildung 2: Beispiel Smart Home

Zu den für IoT typischen Anwendungsbereichen gehört seit langem die intelligente Vernetzung von in einem Wohnhaus (Smart-Home) befindlichen Haushaltsgeräten, Unterhaltungselektronik und weiteren Sensoren in Leuchtmitteln, Thermostaten, Ladegeräten oder Klimaanlagen. Die unterschiedlichen Geräte und Instrumente kommunizieren sowohl Zustandsinformationen als auch Steuerungssignalen an einen Server über ein lokales Netzwerk, z.B. einem im Haus befindlichen Router. Dieser Server kann innerhalb oder außerhalb des Wohnhauses aufgestellt sein und von den Bewohnern über ihre Endgeräte kontrolliert werden. Abbildung 2: Smart Home

Smart-Home Lösungen können für die Übertragung der in den Geräten/Sensoren gewonnenen Daten, sowie der vom Benutzer ausgelösten Steuerungen eigene Nachrichtenformate und Transportprotokolle nutzen. Es kann unterschieden zwischen den Geräten der Anwendungen bzw. der Nutzer, den Netzwerkelementen und den IoT Entitäten für die Verarbeitung bzw. Aufbereitung der Daten werden. Die Verarbeitung der IoT Daten kann in Abhängigkeit von der konkreten Situation und den Qualitätsanforderungen ebenfalls im Wohnhaus (Fog Computing), in der näheren Umgebung (Edge Computing) oder in der Cloud (Cloud-Computing) erfolgen.

2 Qualitätsmerkmale und Standards [285]

Begriffe

Qualitätsmerkmal	[ISTQB 17]: (1) Fähigkeit oder Eigenschaft, welche die Qualität einer Einheit beeinflusst. [ISO/IEC/IEEE 24765:2010(E)] (2) Ein Satz von Eigenschaften eines Softwareprodukts, anhand dessen seine Qualität beschrieben und beurteilt wird. Ein Softwarequalitätsmerkmal kann über mehrere Stufen in Teilmerkmale verfeinert werden. [ISO 9126] Qualitätsmerkmale sind Funktionalität, Zuverlässigkeit, Gebrauchstauglichkeit, Effizienz, Änderbarkeit und Übertragbarkeit. [ISO 9126]
Funktionale Sicherheit (Safety)	Schutz Dritter durch den sicheren Betrieb der Systeme. [ISTQB 16]
IT-Sicherheit (Security)	Eigenschaften der Software, die sich auf die Fähigkeit beziehen, nicht autorisierte Zugriffe auf Programme oder Daten zu verhindern, unabhängig davon, ob diese versehentlich oder vorsätzlich erfolgen. [ISO 9126] Die Fähigkeit des Softwareprodukts zum Schutz von Informationen und Daten, so dass Unbefugte oder Systeme sie nicht lesen oder ändern können und nur autorisierte Personen bzw. Systeme Zugang haben. [ISO/IEC 12207:1995]
Robustheit	Der Grad, zu welchem Ausmaß eine Komponente oder ein System bei ungültigen Eingaben und extremen Umgebungsbedingungen korrekt funktioniert. [ISO/IEC/IEEE 24765:2010(E)] Siehe auch Fehlertoleranz. [ISTQB 16]
Resilienz	Resilienz ist die Fähigkeit einer Organisation, störende Risiken zu bewältigen [ISO Guide 73]. Bei technischen Systemen, ist es die Fähigkeit bei Störungen bzw. Teil-Ausfällen nicht vollständig zu versagen, sondern wesentliche Systemdienstleistungen aufrechtzuerhalten
Performanz	Zeitverhalten und Verbrauchsverhalten (z.B. Energieeffizienz), Skalierbarkeit. [ISTQB 16]
Interoperabilität	Die Fähigkeit eines Softwareprodukts, mit einer oder mehreren spezifizierten Komponenten zusammenzuwirken [ISTQB 16]
Gebrauchstauglichgkeit (engl. Usability)	Qualitätsmerkmale, die gekennzeichnet sind durch Wirksamkeit, Effizienz und Zufriedenheit. [ISO/IEC 25010:2011]
Wartbarkeit	Diagnostizierbarkeit, Aktualisierbarkeit, Restartfähigkeit – auch aus der Ferne. [ISTQB 16]
Produkt-Zertifikat	Unabhängiger fachkundiger Nachweis der Erfüllung von Anforderungen durch ein Produkt.

2.1 Einführung [30]

Die Kenntnis über Qualitätsmerkmale, deren Bedeutung und Priorisierung ist die Basis und bildet die Grundlage für das Quality Engineering – der erste Schritt ist zu wissen, welche Qualitätsaspekte zu adressieren sind. Entsprechende Normen haben sich im industriellen Einsatz etabliert. Das vorliegende Kapitel gibt einen Überblick über Qualitätsmerkmale mit Relevanz und Priorität für IoT Systeme, -Produkte und –Komponenten und deren Bedeutung im IoT Kontext.

Dieser Lehrplan beschreibt Merkmale, die in verschiedenen Normen, im ISTQB-Glossar oder ähnlichem definiert werden. Der Lehrplan wird einzelne Merkmale aus einer Norm nicht auf Merkmale einer anderen Norm mappen oder die Unterschiede zwischen den einzelnen Merkmalen oder deren Unterkategorien diskutieren.

Stattdessen werden die Merkmale in Bezug auf IoT Architekturen, auf Prozesse und auf das Quality Engineering für IoT in den folgenden Kapiteln ausgeführt.

- Überblick über die IoT relevanten Qualitätsmerkmale (10 Min)

IoT-QE LZ 6 (K2) Relevanz und Schwerpunkte von Qualitätsmerkmalen für IoT im Überblick erklären können [10]

Qualitätsmerkmale sind Eigenschaften, die die Qualität eines Systems, eines Produktes oder einer Dienstleistung prägen. Allgemeine Beispiele sind: Gebrauchstauglichkeit, Zuverlässigkeit, Normgerechtigkeit, Ästhetik, Haltbarkeit, Sicherheit und der Bedienungskomfort von Produkten oder auch Kompetenz bei Dienstleistungen.

Das Quality Engineering für IoT verbindet unterschiedlichste Disziplinen wie z.B. System Engineering, Software Engineering für Applikations-, Unternehmens oder Embedded-Software und Qualitätsmanagement. Jede dieser Disziplinen kann Besonderheiten, eigene Normen und eine eigene Definition von Qualitätsmerkmalen mitbringen. Qualitätsmerkmale sind teilweise in generellen Normen (z.B. [ISO/IEC 25010], [ISO/IEC/IEEE 24765:2010(E)], [ISO 9126]) oder auch domänenspezifischen Normen definiert. Eine gemeinsame Sprache und ein gemeinsames Grundverständnis über Qualitätsmerkmale ist Voraussetzung für Effektivität und Qualität in der Zusammenarbeit in einem (interdisziplinären) IoT Projekt.

Systematik von Qualitätsmerkmalen: Die in der Softwareentwicklung verbreitete Norm ISO/IEC 25010 [ISO/IEC 25010] fasst Qualitätsmerkmale für Software zu den folgenden Gruppen zusammen: Funktionalität, Zuverlässigkeit, Benutzbarkeit, Sicherheit, Effizienz, Wartbarkeit, Portabilität und Kompatibilität.

Einige dieser Qualitätsmerkmale bzw. deren Untereigenschaften (z.B. Interoperabilität als Eigenschaft von Kompatibilität) erfordern im IoT Kontext besonderes Augenmerk. Diese werden in weiteren Verlauf des Kurses detaillierter betrachtet.

- Der Betrieb eines IoT Systems: Herausforderung bereits bei der Systemkonzeption (15 Min)

IoT-QE LZ 7 (K2) Die Relevanz der Qualitätsmerkmale auch für den Betrieb erklären können [15]

Bereits beim Design von IoT Systemen muss auf die betrieblichen Belange des Systems geachtet werden. So müssen Funktionalitäten zum Monitoring der Betriebssituation, Monitoring der Leistungsfähigkeit, des Wiederanlaufs und der Wartung berücksichtigt werden (Build to Run).

Die Konzeption der Betriebsphase muss unter anderem die folgenden Servicequalitäten im Blick haben:

Verfügbarkeit – die Voraussetzungen zur Einhaltung der zugesagten Verfügbarkeit müssen geschaffen werden. Bei Änderungen soll die Funktion des Dienstes während der Änderung nicht oder nur im festgelegten Rahmen unterbrochen werden. Die Änderungen dürfen keine negativen Seiteneffekte auf andere Funktionalitäten haben.

Performanz: Zeitverhalten und Verbrauchsverhalten (z.B. Energieeffizienz), Skalierbarkeit [ISTQB 16], Angemessenheit der verfügbaren Kapazität – je nach Betriebssituation können Kapazitätserweiterungen notwendig sein, die vor Auftreten von Engpässen bereitgestellt werden müssen (siehe auch Abschnitt -).

Zuverlässigkeit (Robustheit, Resilienz): Betrieb in harschen Umgebungen, Aufrechterhaltung von (Teil-) Funktionalität bei Störungen z.B. durch Wetter, Vandalismus, gezielten Stör- und Manipulationsversuche, Fehlbedienung, fehlerhafte Eingaben, etc. (siehe auch Abschnitt -).

Aufzeichnung der wichtigsten Betriebsparameter und Nachweisführung der Einhaltung vereinbarter Qualitätsmerkmale: Die Aufzeichnung ausgewählter Kennzahlen zu Performance und Ressourcenverbrauch muss geplant werden.

- Standards (5 Min)

IoT-QE LZ 8 (K1) Die Bedeutung von Standards und regulatorische Anforderungen kennen [5]

Die Sicherstellung der Einhaltung von Standards ist eine der grundlegenden Aufgaben des Quality Engineerings:

- Identifikation relevanter Normen, Standards und Zertifizierungen und ihrer Weiterentwicklungen.
- Umsetzung in konkrete Spezifikationen für das Produkt, das System oder die Dienstleistung.
- Überprüfung der jeweiligen Anforderungen.

Prüfzeichen und Zertifikate dienen der Dokumentation der Konformität und stärken das Vertrauen der Kunden oder Nutzer in die Produkte. Gesetzliche und regulatorische Forderungen (z.B. Elektrosicherheit, Leitlinien von Behörden) müssen beachtet werden.

Unternehmensspezifische Standards sind zu berücksichtigen und Best Practices sollten bekannt sein und genutzt werden. Typische Beispiele für Standards sind ISO/IEC 2700x Information Technology – Security techniques, IEC 61508 Funktionale Sicherheit sicherheitsbezogener elektronischer Systeme oder der BSI Leitfaden Informationssicherheit.

2.2 Qualitätsmerkmale mit besonderer Bedeutung für IoT [135]

Die IoT Qualitätsmerkmale werden entlang der ISO/IEC 25010 [ISO/IEC 2510] diskutiert.

- Funktionalität (5 Min)

IoT-QE LZ 9 (K1) Funktionale Qualitätsmerkmale kennen [5]

Funktionale Qualitätsmerkmale nach ISO/IEC 25010 beziehen sich auf die Erfüllung der Geschäftsanforderungen an ein System, ein Produkt oder einen Dienst.

- **Vollständigkeit (Functional Completeness)** – Umfang der Erfüllung von explizit geforderten oder implizit erwarteten Aufgaben des Systems.
- **Korrektheit (Functional Correctness)** – Genauigkeit mit der das Produkt oder System die gewünschten Ergebnisse erzielt.
- **Angemessenheit (Functional Appropriateness)** – Ist die Ausführung der Funktionalität angemessen realisiert – nicht zu kompliziert oder aufwändig.

Insbesondere die Funktionalität von intelligenten selbstlernenden Systemen erfordert besondere Testverfahren, da sich das Systemverhalten kontinuierlich mit dem Training des Systems ändert.

- IoT Sicherheit (75 Min)

IoT-QE LZ 10 (K2) Die Sicherheitsherausforderungen (sowohl Security als auch Safety) bei IoT Systemen erklären können. [15]

IoT-QE LZ 11 (K3) Eine Analyse der Auswirkungen der Qualitätsmerkmale IT-Sicherheit und funktionale Sicherheit auf Konstruktives QE vornehmen können [60]

Die IoT Sicherheit eines IoT-Systems oder -Produkts umfasst sowohl die Betrachtung von Gefährdungen des Systems selbst (IT-Sicherheit / Security) als auch die Berücksichtigung von Gefahren, die vom System ausgehen (funktionale Sicherheit / Safety).

- **IT-Sicherheit / Security**, ([ISO/IEC 12207:1995]) bezeichnet die Fähigkeit des Softwareprodukts zum Schutz von Informationen und Daten, so dass Unbefugte oder Systeme sie nicht lesen oder ändern können und nur autorisierte Personen bzw. Systeme Zugang haben.
- **Funktionale Sicherheit / Safety** ([ISTQB 16]) bezeichnet den Schutz Dritter durch den sicheren Betrieb der Systeme.

Durch die Anbindung an das Internet steigt die Gefährdung durch manipulative Angriffe und Eindringversuche wie auch von ungewollten physischen Einwirkungen (Zerstörung, Entwendung, Manipulation) deutlich an.

Die Entwicklung sicherer Hard- und Software entsprechend aktueller Marktstandards (z.B. ISO 27034 [ISO 27034]) wirkt auf die Sicherheit des Gerätes und der Schutz der verarbeiteten Informationen.

Die folgenden Qualitätsmerkmale nach ISO/IEC 25010 sind diesem Umfeld zuzuordnen:

- **Vertraulichkeit (Confidentiality)** – Schutz der Informationen vor dem Zugriff unberechtigter Systeme oder Personen beispielsweise durch verschlüsselte Speicherung und Kommunikation.
- **Integrität (Integrity)** – Schutz der Informationen vor unberechtigter Veränderung beispielsweise durch Verschlüsselung oder Prüfsummen.
- **Verfügbarkeit (Availability)** – siehe Kapitel -.
- **Nachweisbarkeit (Non-repudiation)** – Fähigkeit nachzuweisen, dass Aktionen oder Ereignisse tatsächlich stattgefunden haben und Schutz der Nachweise vor Verfälschung.
- **Verantwortlichkeit (Accountability)** – Fähigkeit, Aktionen einer Entität zu genau dieser Entität zurückzuverfolgen.
- **Authentizität (Authenticity)** – Sicherstellen, dass Informationen nur zwischen legitimen Systemkomponenten ausgetauscht werden. Zugehörig: **Identifizierbarkeit,** d.h. die im System befindlichen Komponenten können jederzeit eindeutig adressiert werden.

Vertraulichkeit, Integrität und Verfügbarkeit gelten als primäre Schutzziele, die anderen Qualitätsmerkmale als ergänzende Schutzziele.

Hinzu kommen gesetzliche Vorgaben zum **Datenschutz**, die unbedingt zu beachten sind. Der Schutz der Privatsphäre stellt eine äußerst wichtige Qualitätsanforderung dar, die über europäisches Recht und das bundesdeutsche Datenschutzgesetz (BDSG)gesetzlich reguliert ist. Die Nutzung **personenbezogener Daten** muss vertraglich festgehalten und durch den Kunden freigegeben werden. Eine vertraglich unzulässige Verknüpfung von Daten mit personenbezogenen Daten muss unterbunden werden. Hierbei ist darauf zu achten, dass auch die Dinge personenbezogene Daten speichern können, die dann entsprechend zu schützen sind.

Ethische Aspekte müssen berücksichtigt wie der Schutz der persönlichen Autonomie, der Privatsphäre oder des Vertrauens werden.

Anforderungen zur Gewährleistung der funktionalen Sicherheit (Safety) von IoT Systemen oder -Produkten sind in allgemeiner Form in der Norm IEC 61508 sowie in verschiedenen branchenspezifischen Normen (z.B. ISO 26262 Road vehicles Functional Safety, ISO 50128 Bahnanwendungen, ISO 13849 Sicherheit von Maschinen, etc.) aufgeführt.

Von wesentlicher Bedeutung für die IT-Sicherheit der Architekturen ist die Kenntnis und Risikobewertung der für IoT typischen Angriffsvektoren auf den unterschiedlichen Architekturebenen und -elementen als Basis für die Implementierung entsprechender Schutzmechanismen:

Architekturebene / -element	Angriffsvektoren
Things **(physikalische Geräte)**	- Physikalische Geräteschnittstellen - Gerätespeicher und Speichererweiterung (z.B. SD-Karten) - Firmware der Geräte - Physikalische Manipulation oder Diebstahl der Geräte
Netzwerk-Schicht **(Konnektivität)**	- Webinterfaces der Geräte - Netzwerkschnittstellen der Geräte
IoT-Schicht **(Computation-, Aggregation- und Storage-Dienste)**	- Cloud Web Interfaces - Backend APIs - Update Mechanism (over the air updates) - Sonstige Kommunikation zwischen IoT-Schicht und Netzwerk-Schicht
Applikations-Schicht **(Analytics, Visualisierung und Steuerung)**	- Mobile Applikationen - Webapplikationen
User	- Social Engineering

Tabelle 1: Angriffsvektoren

Diese Angriffsvektoren sind im analytischen Quality Engineering entsprechend der Prioritäten aus der Risikoanalyse mit Testschwerpunkten zu versehen (Kap. -).

Übung zur IT-Sicherheit von IoT-Systemen

Die Kursteilnehmer bilden Gruppen zu je etwa 3-4 Mitgliedern und erarbeiten konkrete Qualitätsanforderungen (30 min). Der erreichte Stand wird im Plenum vorgestellt und diskutiert (15 min).

- Kompatibilität (10 Min)

IoT-QE LZ 12 (K2) Die Anforderungen an Interoperabilität für IoT Systeme erklären können [10]

Kompatibilität – die Möglichkeit, IoT-Systeme, -Produkte oder ihre Komponenten mit den Systemen, Produkten oder Komponenten anderer Hersteller zusammen oder aufeinander abgestimmt zu benutzen.

IoT Systeme, -Produkte oder -Komponenten müssen unter Umständen auf allen Ebenen der IoT Architektur mit Komponenten und Plattformen unterschiedlicher und wechselnder Hersteller koexistieren und ggfs. zusammenarbeiten. Zwei Unterkategorien der Kompatibilität spielen deshalb für IoT Systeme eine wesentliche Rolle:

- **Interoperabilität** – Sicherstellen der Kommunikation durch kompatible Datenformate und Protokolle und der Konnektivität zwischen Geräten unterschiedlicher Hersteller [ISTQB 16].

  Systeme/Komponenten können miteinander Informationen austauschen und nutzen. Dazu sind neben der Konnektivität der Komponenten kompatible Datenformate und Protokolle und eine einheitliche Interpretation der Daten erforderlich.

- **Koexistenz** – Systeme/Komponenten können gemeinsame Infrastrukturen nutzen, ohne sich gegenseitig in ihrer Funktionalität einzuschränken.

- Robustheit und Resilienz (10 Min)

IoT-QE LZ 13 (K1) Die für IoT Systeme wesentlichen Qualitätsmerkmale Robustheit und Resilienz kennen [10]

Robustheit bezeichnet den Umfang, in dem ein System oder eine Komponente außergewöhnlichen Belastungen standhält. Beispielhafte außergewöhnlichen Belastungen: Hitze, Kälte, Erschütterungen aber auch starkes Datenaufkommen, eingeschränkte Kommunikations-Anbindung oder eine schwankende Energieversorgung [ISTQB 16].

Resilienz ist die Fähigkeit zur Störungsbeseitigung sowie Benutzerunterstützung im Störungsfall – bei System- oder Dienstausfall oder Beeinträchtigungen muss ggf. geeignetes Personal oder Automatismen (Services) zur Verfügung stehen, um den Betrieb schnellstmöglich wiederherzustellen und/oder dem Nutzer Unterstützung zu gewähren.

IoT-Produkte oder -Komponenten sind oft harschen Bedingungen ausgesetzt. Als Dinge in der realen Welt müssen sie vor Umwelteinflüssen geschützt werden und oft besonderen physischen Belastungen standhalten. Eine realisierte Selbstverwaltung der abgesetzten Geräte unterstützt die Resilienz der Produkte.

Damit sind die Qualitätsmerkmale Robustheit und Resilienz für die Gesamtsysteme inklusive ihrer möglichen Anteile in der Cloud von großer Bedeutung.

- Wartbarkeit und Übertragbarkeit (15 Min)

IoT-QE LZ 14 (K2) Die Anforderungen an Wartbarkeit und Übertragbarkeit für IoT Systeme erklären können [15]

Wartbarkeit ist definiert als Diagnostizierbarkeit, Aktualisierbarkeit, Restartfähigkeit – auch aus der Ferne [ISTQB 16].

Auch im IoT stehen Hersteller von langlebigen IoT Geräten (z.B. Fahrzeuge, Produktionsmaschinen oder hochwertige Hausgeräte) vor der Herausforderung, dass er die Sicherheit, Interoperabilität und Wartung seiner IoT Geräte in Verbindung mit einer sich ändernden IoT Prozesskette über viele Jahre

hinweg unterstützen muss (z.B. Schließung von Sicherheitslücken, Unterstützung neuer Kommunikationsformate).

Typische Wartungsarten sind dabei korrigierend, verbessernd, adaptiv und vorausschauend (*predictive* und *preventive*). Einerseits ermöglicht die Anbindung an das Internet erst eine automatisierte Wartung inklusive Vorhersagen und Prävention, andererseits sind viele IoT Geräte nicht dauerhaft mit dem Internet verbunden und nur eingeschränkt zugänglich.

Die ISO/IEC 25010 beleuchtet zum Thema Wartung vor allem die Aspekte der Software-Entwicklung:

- **Modularität** (Modularity) – Grad, zu dem ein System oder Computerprogramm aus einzelnen Komponenten besteht, sodass eine Änderung einer Komponente nur minimalen Einfluss auf andere Komponenten hat.
- **Wiederverwendbarkeit** (Reusability) – Grad an Aufwand und Wirksamkeit, zu dem ein Asset für mehr als ein System oder zur Entwicklung weiterer Assets genutzt werden kann.
- **Analysierbarkeit** (Analyzability) – Aufwand und Wirksamkeit, mit dem ein Produkt / System / Komponente hinsichtlich eines aufgetretenen Fehlverhaltens oder einer Störung diagnostiziert werden kann. Umfasst das Erkennen, ob und welche Teile von einer Störung betroffen oder für eine Störung verantwortlich sind und welches der Grund für die Störung ist.
- **Modifizierbarkeit** (Changeability) – Aufwand und Wirksamkeit, mit dem ein System verändert werden kann, ohne die Funktionalität zu beeinträchtigen oder Fehler zu injizieren.
- **Prüfbarkeit** (Testabillity) – Aufwand und die Wirksamkeit, mit dem Testkriterien fixiert werden können und der für die Durchführung von Tests erforderlich ist.

ISO/IEC 25010 gliedert das Qualitätsmerkmal **Übertragbarkeit** wie folgt auf:

- **Anpassbarkeit** – Grad, zu dem ein Produkt oder System effektiv und effizient auf andere oder entstehende Hardware-, Software- oder Nutzungsumgebungen angepasst werden kann (Individualisierbarkeit, falls die Anpassungen von einem Endnutzer vorgenommen werden).
- **Installierbarkeit** – Grad der Effektivität und Effizienz, mit der ein Produkt oder System in einer bestimmten Umgebung erfolgreich installiert und/oder deinstalliert werden kann.
- **Austauschbarkeit** – Grad, zu dem ein Produkt ein anderes bestimmtes Produkt mit demselben Zweck in derselben Umgebung ersetzen kann.

- Performanz (10 Min)

IoT-QE LZ 15 (K2) Die besonderen Herausforderungen an das Qualitätsmerkmal Performanz (Zeitverhalten und Verbrauchsverhalten) für IoT Systeme erklären können [10]

Gutes Laufzeitverhalten bei gleichzeitig niedrigem Ressourcenverbrauch stellt für viele IoT Produkte und -Komponenten eine große Herausforderung dar, da die Geräte oft sehr klein sind und günstig produziert werden müssen und keine externe Stromversorgung haben. Moderne Übertragungstechniken und -protokolle (z.B. LTE-M, LoRa, SigFox, etc.) sind auf kleine Datenraten und große Reichweiten bei kleinem Stromverbrauch optimiert, so dass Bandbreitengrenzen und ggfs. hohe Latenzzeiten im Design entsprechend berücksichtigt werden müssen.

Beim Qualitätsmerkmal Performanz unterscheidet ISO/IEC 25010 die folgenden für IoT Systeme, -Produkte und -Komponenten besonders relevanten Unterkategorien:

- **Laufzeitverhalten** (Time Behaviour) – Fähigkeit, Anforderungen hinsichtlich der Antwort- und Bearbeitungszeiten sowie die Durchsatzraten bei der Ausführung seiner Funktionen zu erfüllen.
- **Ressourcenverbrauch** (Resource utilization) – Art und Umfang der Verbräuche von Ressourcen (Strom, Speicherplatz, etc.), um die geforderten Funktionen zu erfüllen.
- **Kapazität** (Capacity) – Grad, zu dem die Höchstgrenzen eines Produkt- oder Systemparameters an die Anforderungen erfüllt sind.

- Ethische Aspekte bei IoT (10 Min)

IoT-QE LZ 16 (K2) Die Relevanz Ethischer Aspekte für IoT erklären können [10]

Die zunehmende Vernetzung der Dinge und der steigende Einsatz von Automatisierung, Datenanalysen und maschinellem Lernen führen zu einem gefühlten Kontrollverlust bei Nutzern und Betreibern von IoT Systemen. Der Schutz der Privatsphäre und andere, von der Ethik berührte Entscheidungen werden teilweise Maschinen überlassen. Dieser Herausforderung muss bei der Konstruktion und Validierung von IoT Systemen Rechnung getragen werden. Ethik beschäftigt sich mit den Regeln und der Bewertung menschlichen Handelns. Bei der Bewertung ethischer Aspekte muss der jeweilige Einsatzkontext des IoT Systems und der gesellschaftliche Hintergrund (z.B. Moral, Kultur, Gesetze) der Nutzer und Betreiber berücksichtigt werden.

Dabei muss beachtet werden, dass ethische Entscheidungen aufgrund des unterschiedlichen moralischen und kulturellen Hintergrunds in den verschiedenen Regionen unterschiedlich ausfallen können. Wichtige Aspekte für die Bewertung von ethischen Fragestellungen sind u.a. Legalität, Gerechtigkeit, Respekt, Entscheidungsfreiheit, Umweltschutz und Nachhaltigkeit.

So können ethische Aspekte alle Qualitätsmerkmale eines IoT Systems beeinflussen.

Für IoT Projekte spielen ethische Aspekte eine zunehmend wichtige Rolle. Der Quality Engineer muss mögliche ethische Implikationen erkennen können, um diese bei der Konstruktion und Absicherung von IoT Systemen angemessen zu berücksichtigen. Dabei liegt die Verantwortung nicht beim Quality Engineer. Der Input zu ethischen Fragestellungen muss beim Qualitätsmanagement oder/und der Projektleitung eingefordert werden.

Beispiele:

1. Schutz der Privatsphäre: Welche gesellschaftlichen Normen (oder Gesetze) müssen beachtet werden, um das Selbstbestimmungsrecht des Menschen nicht zu verletzen (z.B. durch Möglichkeiten der Überwachung medizinischer Diagnosen)?

   Illegal wäre es, wenn medizinische Werte ohne das Wissen des Nutzers vom messenden Gerät nicht nur an die Applikation, sondern auch an die Krankenkasse des Nutzers übertragen werden.

2. Entscheidungsfreiheit: Welche Informationen dürfen einem Nutzer einer Software nicht vorenthalten werden, damit er immer noch frei entscheiden kann?

   Im Hinblick auf Sicherheit und Nachhaltigkeit ist es fragwürdig, wenn das Navigationssystem dem Nutzer nur schnelle Routen vorschlägt und keine Option auf langsamere oder risikoärmere zulässt.

2.3 Qualitätsmerkmale und Ihre Spannungsfelder in IoT Systemen [60]

- Das Spannungsfeld von IT-Sicherheit und funktionaler Sicherheit (15 Min)

IoT-QE LZ 17 (K2) Das Spannungsfeld von IT-Sicherheit und funktionaler Sicherheit erklären können [15]

Im IoT-Umfeld kommt es – stärker noch als im klassischen Umfeld – zu einem Spannungsfeld zwischen den Anforderungen an die IT-Sicherheit der Dinge und der vergleichsweise weit entwickelten IT-Sicherheit für klassische IT-Architekturen mit ihren oben erwähnten Schutzzielen. Klassisch erfordert funktionale Sicherheit stabile - meist zertifizierte - Softwareversionen, wohingegen IT-Sicherheit auch regelmäßige und kurzfristig notwendige Updates verlangt. Dabei gewinnen wegen der Datenbezogenheit bei IoT neben den primären auch die ergänzenden Schutzziele zunehmend an Bedeutung: Liegt der Fokus bei den – meist in Datenzentren – gut gesicherten IT-Systemen auf der Vertraulichkeit und Integrität, so liegt das Augenmerk bei den in aller Welt verstreuten Dingen auf Authentizität und Zurechenbarkeit. Dabei stellt die Verteilung funktionaler Sicherheit über IT und

Dinge hinweg automatisch Anforderungen an die Vernetzung und Interaktion bzgl. Verfügbarkeit, Vertraulichkeit und Verbindlichkeit.

- Das Spannungsfeld von Gebrauchstauglichkeit, Wartbarkeit und IT-Sicherheit (15 Min)

IoT-QE LZ 18 (K2) Das Spannungsfeld von Gebrauchstauglichkeit, Wartbarkeit und IT-Sicherheit erklären können [15]

Dinge und Services sollen möglichst einfach und sicher installiert, angepasst, in Stand gehalten und außer Betrieb genommen werden können (siehe auch Kapitel 6. Lifecycle):

- Einfach, d.h. möglichst automatisch und ohne aufwändige manuelle Eingriffe von Benutzern oder Servicebetreibern.
- Sicher, d.h. ohne Verletzung des Datenschutzes oder anderer zu schützender Werte.

Ein Benutzer möchte eine möglichst nahtlose Integration in seine installierte Umgebung mit seinen bereits genutzten Services und Dingen inklusive einer intuitiven Möglichkeit, diese auf seine individuellen Wünsche anzupassen. Demgegenüber muss ein Servicebetreiber einen vertragsgemäßen, sicheren Betrieb gewährleisten, der im Notfall auf manuelle Fernwartung zurückgreift und möglichst ohne den Einsatz eines Servicetechnikers vor Ort auskommt, was oftmals im Widerspruch zur einfachen Bedienbarkeit steht.

- Das Spannungsfeld von Resilienz, Robustheit und Performanz (15 Min)

IoT-QE LZ 19 (K2) Das Spannungsfeld von Resilienz, Robustheit und Performanz erklären können [15]

Die grundsätzliche Struktur von IoT Systemen als verteilte Systeme ist für die Resilienz von Vorteil, während sie aufgrund der Vielzahl an Schnittstellen eine Herausforderung für die Robustheit und Performanz darstellen kann. So kann z.B. die Störung / der Ausfall eines zentralen Gateways ein IoT System unverfügbar machen.

Hier kann eine redundante Auslegung sowohl die Robustheit als auch Performanz verbessern, wobei mehr Systemressourcen benötigt werden.

- Das Spannungsfeld Konnektivität, Interoperabilität und IT-Sicherheit (15 Min)

IoT-QE LZ 20 (K2) Das Spannungsfeld von Konnektivität, Interoperabilität und IT-Sicherheit erklären können [15]

IoT Systeme und –Komponenten sind vernetzt, wofür interoperable Schnittstellen benötigt werden, um zuverlässig die Konnektivität zu gewährleisten. Andererseits bieten interoperable Schnittstellen oftmals ähnliche, wenn nicht gleiche Schwachstellen bzgl. IT-Sicherheit auf. Zudem können durch ein schadhaftes IoT Gerät über die Vernetzung IoT relativ einfach weitere Geräte und Komponenten eines IoT Systems gefährdet werden.

2.4 Zusammenhang zwischen Qualitätsmerkmalen und Anforderungen [60]

IoT-QE LZ 21 (K3) Die Qualitätsmerkmale eines Systems bewerten und daraus Anforderungen an das IoT System ableiten können [50]

Die Anforderungen an ein IoT. System sowie und deren Priorisierung ergeben sich direkt aus den relevanten Qualitätsmerkmalen.

Übung zur Herleitung von Anforderungen

Rollenspiel an Hand eines realistischen Beispiels/Szenarios eines IoT Produktes/Systems:

- Die Kursteilnehmer bilden Gruppen zu je etwa 3-4 Mitgliedern. Jede dieser Gruppen identifiziert und priorisiert die wichtigsten Qualitätsmerkmale.
- Aus diesen werden die zwei wichtigsten ausgewählt und für diese werden jeweils mehrere Anforderungen an das System aus Sicht des QE4IoT beschrieben. (25 min)
- Anschließend wird die Auswahl im Plenum vorgestellt und diskutiert. (25 min)

2.5 Beispiel „E-Health“ [10]

(Angelehnt an [oneM2M 16], chapter 7.3 *Secure remote patient care and monitoring*)

Anwendungen im Umfeld digitaler Medizin (E-Health) ermöglichen zunehmend eine Fernüberwachung, Ferndiagnosen und eine Fernbetreuung von Patienten und ihrer gesundheitlichen Parameter. Sie erübrigen damit zumindest teilweise die Notwendigkeit eines Praxisbesuches oder der Vor-Ort-Betreuung der Patienten durch Arzt oder Pflegepersonal und führen damit zu erheblichen Kosteneinsparungen und vermeiden Aufwände und Unannehmlichkeiten. Zusätzlich ermöglichen sie ein ganzheitliches Management chronischer Erkrankungen und erlauben es den Patienten, länger selbstständig im vertrauten Umfeld zu leben.

Die Messung unterschiedlicher gesundheitlicher Parameter erfolgt dabei mit Hilfe von medizinischen und anderen Sensoren im oder am Körper oder im Umfeld der Patienten. Mit Hilfe geeigneter Anwendungen können diese Informationen fernausgelesen und analysiert werden. Alarme, die durch diese Sensoren ausgelöst werden, können automatisch Meldungen an Hilfeleister generieren, sobald lebensbedrohliche Situationen erkannt oder Schwellwerte überschritten werden. Nachrichten können aber auch an Pflegekräfte oder Familienangehörige übermittelt werden, wenn weniger gravierende Anomalien erkannt werden. In der umgekehrten Richtung können über ein solches System durch Aktoren beim Patienten aber auch Handlungen ausgelöst (beispielsweise Dosierungen verändert oder Hilfsleistungen fernbedient) werden.

Anmerkung: In vielen Rechtssystemen – so auch in Deutschland – ist der Schutz von personenbezogenen Daten und insbesondere der von gesundheitsbezogenen Daten streng reguliert und Datenschutzverletzungen in diesem Umfeld werden stark bestraft.

E-Health-Systeme können private, zu schützende Daten auf ganz unterschiedlichen sensiblen Ebenen enthalten. Eine große Sorgfalt ist erforderlich, damit der Zugriff auf diese verschiedenen Daten nur für die jeweils autorisierte Nutzergruppe (Patient, Arzt, Pflegedienst, Familie) möglich ist.

Beteiligte Gruppen:

- Patienten, die Sensoren zur Messung ihrer medizinischen Werte benutzen.
- Betreiber von E-Health Anwendungen, die diese Sensoren bereitstellen und den Betrieb zur Überwachung der Messwerte durchführen und Dienste im Zusammenhang mit der Verarbeitung von Nachrichten an Pflegedienste, etc. erbringen.
- Medizinisches und pflegerisches Personal (Krankendienste, Pflegedienste, Ärzte, etc.) und andere administrative Dienstleister (Abrechnungsstellen, Versicherungen), die kontrollierten Zugriff auf ausgewählte Gesundheitsdaten erhalten müssen.
- Technische Dienstleister wie Netzprovider, Softwareanbieter, etc.

Auslöser für einen Datenzugriff:

- Neue Messdaten durch ein medizinisches IoT Gerät liegen vor.

- Eine Auswertung von empfangenen medizinischen Daten liegt vor und eine Reaktion (Alarm, Benachrichtigung, etc.) muss erfolgen.
- Eine Anfrage nach sensiblen medizinischen Daten zu einem Vorgang liegt von einem Berechtigten vor.
- Ein neuer Beteiligter (z.B. ein neuer Arzt) ist für ein medizinisches Szenario zuzulassen.

3 Konstruktives QE – IoT Architektur [165]

Begriffe

Edge Computing	Dezentrale Datenverarbeitung mittels Teilauswertung von Sensorik Daten am Rand des Netzwerks, der sogenannten Edge, als Vorbereitung für den Upload in die Cloud.
Fog Computing	Dezentrale Datenverarbeitung mittels Teilauswertung von Daten in einem lokalen Netzwerk als Vorbereitung für den Upload in die Cloud.
Referenzmodell	Ein Referenzmodell ist ein abstraktes Framework für das Verständnis der wesentlichen Beziehungen zwischen den Entitäten einer Umgebung und dem Entwickeln von einheitlichen Standards oder Spezifikationen zur Unterstützung dieser Umgebung.

3.1 Was eine Architektur für IoT geeignet macht [15]

IoT-QE LZ 22 (K1) Wissen was eine Architektur für IoT geeignet macht [15]

Im Kontext IoT spielen Referenzarchitekturen eine herausragende Rolle, da typischerweise sehr komplexe Gesamtsysteme mit vielen Teilkomponenten betrachtet werden. Referenzarchitekturen müssen grundlegende Definitionen liefern und Gemeinsamkeiten für alle Systeme, die auf ihr beruhen, spezifizieren. Zudem müssen die besonderen Anforderungen im IoT Kontext berücksichtigt werden um zur Umsetzung von spezifischen IoT Projekten geeignet zu sein [Weyrich 16], [Heidrich 16]:

- Konnektivität: die Kommunikation zwischen verschiedenen Partnern (Geräten, Anwendungen, Diensten) muss durch die Referenzarchitektur sichergestellt werden.
- Interoperabilität: es kommen unterschiedliche Technologien zum Einsatz, gleichzeitig muss die Interoperabilität zwischen den einzelnen Komponenten des Gesamtsystems durch die Referenzarchitektur sichergestellt werden.
- Skalierbarkeit: Referenzarchitekturen müssen die Möglichkeit bieten, kleine IoT Lösungen als auch große IoT Lösungen, die ggf. eine Vielzahl von Geräten, Anwendungen und Diensten umfassen, abzubilden
- Datenerhebung und Datenanalyse: die Erhebung, Analyse und Weitergabe von Daten ist eine der grundlegenden Funktionen in IoT, die durch eine Referenzarchitektur ermöglicht werden müssen. IoT-Business ist Datenbusiness.
- IT-Sicherheit und Datenschutz: in allen Bereichen des IoT benötigt, muss durch eine Referenzarchitektur berücksichtigt werden. Hier gilt die Prämisse: - Security-by-Design
- Im Kontext industrielles IoT (IIoT) ggf. Echtzeitfähigkeit: d.h. verschiedene Klassen zeitlicher Anforderungen müssen durch die Referenzarchitektur abgebildet werden können

3.2 IoT Referenzarchitekturen [150]

Adäquate und auf die Domäne optimierte Architekturen sind die technische Basis für die Qualität eines Systems. Die Kenntnis der Spezifika von und Anforderungen an Architekturen im IoT Kontext sind daher ein wichtiger Bestandteil des Quality Engineerings für IoT.

- Überblick über bestehende IoT Referenzarchitekturen (10 Min)

IoT-QE LZ 23 (K1) Ausgewählte IoT Referenzarchitekturen kennen [10]

Referenzarchitekturen sind Modellmuster / Referenzmodelle für eine Klasse von Architekturen. Referenzarchitekturen definieren die Architektur eines Systems aus mehreren unterschiedlichen Blickwinkeln. Ein wesentlicher Blickwinkel definiert die Elemente einer Architektur. Zudem

beschreibt eine Referenzarchitektur die Interaktionen (Datenkommunikation, Synchronisation von Aktionen) unabhängig von der zugrundeliegenden Plattform. Sie bietet als generisches Modell Regeln und Leitlinien bei der Entwicklung einer spezifischen Architektur eines Systems und dient [ISO/IEC CD 30141]:

- der Beschreibung der Eigenschaften eines IoT Systems,
- der Definition der Domänen des IoT Systems,
- der Beschreibung des IoT Systems und seiner Elemente,
- der Beschreibung der Interoperabilität der Entitäten eines IoT Systems.

Eine übergreifende und umfassende Standardisierung von Referenzarchitekturen für IoT existiert derzeit nicht. Eine Vielzahl domänenunabhängiger und domänenspezifischer Referenzarchitekturen befindet sich derzeit in Entwicklung.

Die Referenzarchitektur AIOTI HLA (High Level Architecture) der "Alliance for Internet of Things Innovation" (AIOTI) [AIOTI 16] ist eine der - Stand 2017 - prominenten domänenunabhängigen IoT Referenzarchitekturen und dient in vorliegendem Lehrplan als Leitlinie für das IoT Architekturthema. Das Referenzmodell der Industrie 4.0 ist RAMI 4.0, eine in Deutschland prominente Initiative.

Das Referenzmodell OneM2M ist ein drittes prominentes Modell, welches darauf abzielt, eine Spezifikation für einheitliche M2M-ServiceSchicht zu etablieren umso den Austausch und das Teilen von Daten unter allen möglichen IoT-Applikationen zu ermöglichen.

Neben den Referenzarchitekturen existieren diverse kommerzielle und Open Source Lösungen für IoT IT-Plattformen in der Cloud. Hardware Plattformen und Tool Kits erleichtern auch kleineren Projekten den Einstieg in IoT Entwicklungen.

- AIOTI HLA (60 Min)

IoT-QE LZ 24 (K2) Die Elemente von IoT Architekturen mit Hilfe von AIOTI erklären können [15]

Ein wesentlicher Aspekt ist der statische Blickwinkel auf die Elemente einer IoT Architektur. Das **Domain Model** der AIOTI Referenzarchitektur definiert die Elemente einer IoT Architektur wie folgt:

- User: Benutzer, menschlich oder anderweitig.
- Thing: physisches Objekt.
- IoT Service.
- Virtual Entity: virtuelle Instanz des physischen Objekts.
- IoT Device: Schnittstelle zu den physischen Möglichkeiten des physischen Objekts.

Ein User interagiert mit einem physischen Objekt (Thing), wobei ein IoT Service als Vermittler dieser Interaktion dient. Dieser IoT Service ist verbunden mit einer Virtual Entity (Virtuelle Instanz), die das physische Objekt virtuell abbildet und dessen Charakteristika in der virtuellen Welt repräsentiert. Die Interaktion des IoT Service mit dem physischen Objekt wird durch ein IoT Gerät (Device) ermöglicht, das auch die physischen Fähigkeiten des Dings erschließt.

IoT-QE LZ 25 (K2) Die Schichten von IoT Architekturen am Beispiel AIOTI erklären können [15]

Ein weiterer wesentlicher Aspekt ist der dynamische Blickwinkel, bei AIOTI abgebildet durch das AIOTI**-Funktionsmodell.** Es beschreibt Funktionen und Schnittstellen zwischen den Elementen eines IoT Systems und besteht aus drei Schichten:

- **Die Applikationsschicht (Application Layer)** beinhaltet Kommunikations- und Schnittstellenmethoden für die Kommunikation zwischen Prozessen.
- **Die IoT Schicht (IoT Layer)** enthält die spezifische IoT Funktionalität (z.B. Datenmanagement) und stellt diese über Application Programming Interfaces der Applikationsschicht zur Verfügung. Die IoT Schicht verwendet die Dienste der Netzwerkschicht.

- **Die Netzwerkschicht (Network layer)** gruppiert Dienste auf Daten- und Kontrollebene. Die Netzwerkschicht stellt Transportmechanismen für Nutzerdaten (Kommunikation nah und fern sowie zwischen Entitäten der IoT-Schicht) und Steuerungsdienste zur Verfügung.

IoT-QE LZ 26 (K2) Die Funktionen der Schichten in IoT Architekturen am Beispiel AIOTI erklären können [15]

Die Funktionen innerhalb der Schichten werden durch Entitäten beschrieben:

- Die **App-Entität** realisiert die die IoT Applikationslogik dezentral in Geräten, Gateways oder Servern. Beispiele: Trackingsysteme für Fahrzeugflotten, Remote Blutzuckerüberwachung, etc.
- Die **IoT Entity** stellt IoT Funktionen und die daraus generierten Daten den App-Entitäten oder anderen IoT Entitäten zur Verfügung. Eine IoT Entität verwendet die darunterliegende Netzwerkschicht um Daten zu senden und zu empfangen und für den Zugriff auf die Kontrollebene des Netzwerkes.
- Die **Netzwerke** der Netzwerk-Schicht integrieren typischerweise heterogene Netzwerk-Technologien (PAN, LAN, WAN, etc.) und Netzwerk-Domänen, welche über das Internetprotokoll verbunden sind.

Je nach eingesetzten Kommunikationstechnologien kann die Netzwerkschicht unterschiedliche Dienstgüten (Quality of Service, QoS) anbieten. Letztlich werden die Anforderungen hieran von der Applikationsschicht bestimmt.

IoT-QE LZ 27 (K2) Den spezifischen Einfluss der Daten auf IoT Architekturen erklären können [15]

Das unter Umständen hohe Datenvolumen sowie die teilweise begrenzten Bandbreiten und Speichervolumen der Entitäten in der IoT Schicht, sowie deren nicht immer kontinuierlich gewährleistete Online-Verfügbarkeit machen **Edge Computing** zu einem wichtigen architekturalen Ansatz. Edge Computing bezeichnet die dezentrale Datenverarbeitung am Rand des Netzwerks, der sogenannten Edge. Daten aus einem IoT System werden auf der Edge (z.B. im Gateway) aufbereitet, aggregiert und gespeichert und über die Netzwerk-Schicht den Applikationen direkt oder wiederum über eine Cloud zur Verfügung gestellt.

Einem analogen Ansatz folgt **Fog Computing**. Fog-Systeme sind Cluster von IoT Systemen, quasi kleine Rechenzentren, die im lokalen Netzwerk eine Teilauswertung der Daten vornehmen, um sie für den Upload in die Cloud vorzubereiten. Bei Edge geht es noch mehr um direkte Geräte-Sensorik.

- RAMI 4.0 (25 Min)

IoT-QE LZ 28 (K1) RAMI als spezifische IoT Architektur kennen [10]

Das „Referenzarchitekturmodell Industrie 4.0“, kurz RAMI 4.0, wurde im Jahr 2016 als DIN SPEC 91345:2016-04 veröffentlicht. Das Referenzarchitekturmodell zielt darauf ab, mit den beteiligten Branchen (z.B. IKT, Automatisierung und Maschinenbau) ein einheitliches Verständnis für das Thema Industrie 4.0 herzustellen.

IoT-QE LZ 29 (K2) Die Schichten von IoT Architekturen am Beispiel RAMI 4.0 erklären können [15]

RAMI 4.0 beschreibt die wesentlichen Elemente eines Assets (aus der physischen Welt oder der Informationswelt) mittels eines dreidimensionalen Schichtenmodells. Auf diese Weise sollen komplexe Zusammenhänge aufgegliedert werden. Durch Kombination der drei Achsen des Schichtenmodells wird der jeweils relevante Aspekt zu jedem Zeitpunkt im Lebenslauf eines Assets dargestellbar. Die drei Achsen sind: [DIN SPEC 91345]:

- Architektur (Layers) beschreibt die Architektur mit ihren Funktionen und funktionsspezifischen Daten in Form von 6 Schichten: Asset, Integration, Communication, Information, Functional, Business;
- Produktlebenszyklus (Life Cycle & Value Stream) stellt den Lebenslauf eines Assets dar (Entstehung bis Entsorgung) sowie den Wertschöpfungsprozess in Anlehnung an IEC 62890;

- Hierarchie-Achse (Hierarchy) orientiert sich am Referenzarchitekturmodell für eine Fabrik in Anlehnung an die Normen DIN EN 62264-1 und DIN EN 61512-1, erweitert um Industrie 4.0 Aspekte.

Zu beachten ist, dass die Beschreibung im RAMI4.0 eine rein logische Beschreibung ist und sich eine reale Umsetzung davon unterscheiden kann. Security ist für RAMI4.0 ein elementarer Bestandteil und muss bei der Beschreibung jedes Abschnitts der drei Achsen immer mit betrachtet und beschrieben werden [DIN SPEC 91345].

- OneM2M (25 Min)

IoT-QE LZ 30 (K1) oneM2M als spezifische IoT Architektur kennen [10]

OneM2M ist ein erstmals 2015 von der gleichnamigen Organisation veröffentlichter Standard, der diverse Spezifikationen umfasst und das Ziel hat, eine einzige, horizontale Plattformarchitektur für den Austausch und das Teilen von Daten unter allen möglichen Applikationen und Technologien zu verwirklichen.

IoT-QE LZ 31 (K2) Die Schichten von IoT Architekturen am Beispiel OneM2M erklären können[15]

OneM2M nutzt ein 3 Schichten Modell, welches Applikationsschicht (Application Layer), eine Middleware-Schicht (Common Services Layer) und eine Netzwerkschicht umfasst. Die Funktionale Architektur von OneM2M definiert verschiedene Entitäten, basiert auf dem 3 Schichten Modell und umfasst die folgenden Funktionen [oneM2M 18]:

- Application Entity (AE): Entitäten in der Applikationsschicht, welche eine Applikationslogik implementieren.
- Common Service Entity (CSE) repräsentiert die Instanziierung einer Sammlung von „Common Service Functions“, welche die üblichen IoT-Funktionen für die Applikationen bereitstellen (z.B. Daten- und Gerätemanagement, Lokalisierungsservices)
- Network Service Entity (NSE) stellen die Services des darunterliegenden Netzwerkes für die CSEs bereit (z.B. Gerätemanagement, Lokalisierungsservices, oder Gerätetriggering). Hinweis: die darunterliegenden Netzwerke stellen den reinen Datentransport-Service für die Entitäten in oneM2M bereit – diese Services sind nicht Bestandteil eines NSE.

oneM2M verfügt neben der Architekturspezifikation über eine breite Palette an weiteren Spezifikationen [oneM2M drafts], wobei das Hauptaugenmerk auf dem Common Service Layer liegt, aber beispielsweise auch Spezifikationen zu Tests und Sicherheitsmaßnahmen umfasst. Für Common Service Layer und Applikation-Layer normativ spezifiziert ist die Verwendung der Protokoll CoAP, MQTT, LWM2M, Websockets und HTTP. Außerdem werden Metainformationen wie Datenstrukturen und APIs definiert, so dass Drittanwendungen an oneM2M-Systeme anbinden können. Dies ermöglicht den Austausch von IoT-Daten zwischen verschiedenen Systemen und Anwendungen.

- Abbildung von IoT Systemen auf Referenzmodelle (30 Min)

IoT-QE LZ 32 (K3) Eine IoT Referenzarchitektur auf eine spezifische IoT Systemarchitektur abbilden können [30]

Im Sinne des konstruktiven Quality Engineering sollte sich die spezifische Architektur eines IoT-Systems an einer geeigneten Referenzarchitektur orientieren.

Übung zu IoT-Referenzarchitekturen

Die Kursteilnehmer bilden Gruppen zu je etwa 3-4 Mitgliedern und lösen folgende Aufgabenstellungen:

- Die vorgestellten Referenzmodelle besitzen Gemeinsamkeiten. Welche sind dies? Können AIOTI HLA und oneM2M aufeinander abgebildet werden? Wenn ja, wie?
- Das Smart-Home Beispiel (siehe auch Kap. 1.3) soll mit Hilfe unterschiedlicher IoT Referenzarchitekturen dargestellt werden, AIOTI HLA und oneM2M.

Anschließend werden die Ergebnisse von den Gruppen vorgestellt und im Plenum diskutiert.

4 Konstruktives QE – Prozesse und Methoden [85]

Begriffe

DevOps	DevOps (Zusammensetzung aus Development - Operations) ist ein Vorgehen bei Entwicklung und Administration von Anwendungen. Durch gemeinsame Anreize, Prozesse und Werkzeuge soll eine effektivere und effizientere Zusammenarbeit der Bereiche Dev, Ops und Qualitätssicherung ermöglicht werden.
Agile Softwareentwicklung	Agile Softwareentwicklung zielt darauf ab, den Entwicklungsprozess schlanker und flexibler zu halten. Selbstorganisierte Teams, enge Abstimmung mit dem Kunden sowie die Lieferung von Funktionalitäten in kurzen Abständen sind charakteristisch.
Update-Fähigkeit	Möglichkeit zur Erweiterung und/oder Verbesserung einer Version eines Softwareproduktes.

4.1 Prozesse und Best Practices für die IoT Entwicklung [10]

IoT-QE LZ 33 (K1) Best Practices in IoT kennen [10]

Das Ziel einer lernenden, sich selbst kontinuierlich verbessernden Organisation ist integraler Bestandteil von IoT Geschäftsmodellen. Datengetrieben werden Handlungsoptionen abgeleitet, die auch zu Änderungen in Geschäftsabläufen und Organisation führen können. Die Festlegung und Einhaltung von Prozessen nützt der Organisation durch:

- Schaffung von Transparenz (Fehlervermeidung durch Verstehen),
- Schaffung von Verantwortlichkeiten (Identifikation der Beteiligten mit ihrer Tätigkeit),
- bereichsübergreifende Kommunikation und Koordination,
- prozessorientiertes Denken und Handeln und
- Bildung einer Basis zur weiteren Optimierung und Automatisierung.

Best Practices bieten einen ähnlichen, wenn auch nicht vergleichbaren Nutzen, da sie unverbindlich und meist nur in Teilbereichen der Organisation angewendet werden. Im Unterschied zum Prozess und als Ergänzung dazu ist ein Best Practice:

- eine unverbindliche Empfehlung, wie in einem bestimmten Fall vorzugehen ist,
- flexibler als ein Standard und
- bei geänderten Anforderungen oder Bedingungen einfacher durch eine erfolgsversprechende Vorgehensweise ersetzbar.

Aufgrund der für IoT Produkte und -Lösungen geltenden besonderen Anforderungen, z.B. an Time-to-Market, Variantenvielfalt, Komplexität, datengetriebene Wertschöpfung ist auf die große Bedeutung von interdisziplinären Teams aus den Bereichen Development, Operations und Qualitätssicherung (kurz DevOps) hinzuweisen [Humble 10]. Zu den Best Practices im IoT gehören:

- Das Team trägt gemeinsam die Gesamt-Verantwortung für die jeweilige Entwicklung und verfügt über entsprechende Handlungsmöglichkeiten (Kompetenzen).
- Key Performance Indicator werden im Team und über Teamgrenzen hinweg ausgehandelt.

- Alle Umsetzungsschritte (Prototyp, Implementierung, Automatisierung etc.) orientieren sich am Nutzen (bzgl. Wertschöpfung).

4.2 Ansätze zur kontinuierlichen Entwicklung [35]

- Vorteile agiler Methoden (10 Min)

IoT-QE LZ 34 (K1) Die Vorteile agiler Methoden kennen [10]

Sequentielle Entwicklungsmodelle wie das V-Modell, die vorgelagerte Spezifikationen und nachgelagerte Verifikation & Validierung mit dem Kunden nutzen, geraten auch und insbesondere bei IoT-Systemen unter Druck durch:

- hohe Aufwände für die Spezifikationserstellung bei IoT-Produkten, die häufig sehr komplex sein können,
- zusätzliche technische Anforderungen, die noch während der Entwicklung entstehen,
- neue Anforderungen, die auf einer schnellen Anpassung an Kundenbedürfnisse beruhen.

Hier wirken agile Entwicklungsansätze durch ihre Prinzipien auf Effizienzsteigerungen hin. Sie basieren auf kontinuierlichen Verbesserungsprozessen, die in Iterationen beispielsweise nach dem Schema Plan-Do-Check-Act durchgeführt werden können:

- Plan – definiere Objekte und Prozesse zur Zielerreichung
- Do – setzte den Plan um und sammle dabei Daten zur Beurteilung
- Check – überprüfe auf Abweichungen zu Plan und Zielerreichung, sowie Optimierungspotential
- Act (Adjust) – übernehme erfolgreiches und verbessere

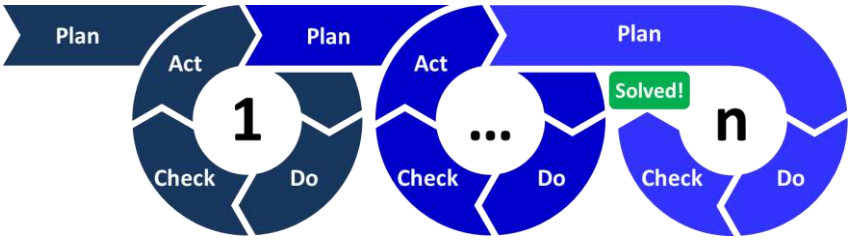

Abbildung 3: Agile Entwicklung mit dem Plan-Do-Check-Act-Schema

- Vorteile automatisierter Methoden (10 Min)

IoT-QE LZ 35 (K1) Die Vorteile automatisierter Methoden kennen [10]

Die Software, Dienste und Prozesse eines IoT Systems zur Sammlung und Verarbeitung von Daten inklusive der Aufbereitung der Ergebnisse werden oftmals kontinuierlich weiterentwickelt.

Zudem ist der Faktor Zeit für IoT Systeme oftmals besonders relevant, beispielsweise für das zeitnahe Ausrollen von Security Updates. Oft wird auch eine schnelle Reaktion auf Marktentwicklungen gefordert.

Ein händisches Eingreifen beinhaltet hierbei ein erhöhtes Fehlerrisiko und führt aufgrund der Länge der beteiligten Prozessketten unweigerlich zu Flaschenhälsen. Die Erhöhung von Rechenleistung zur weitest gehenden Automatisierung ist dagegen vergleichsweise kostengünstig, wobei die Aufwände für Entwicklung und Pflege der Automatisierung ebenso beachtet werden müssen. Werden bei der Automatisierung zudem Parallelisierungsoptionen genutzt, können relativ einfach Latenzen verringert und die Robustheit der Automatisierung erhöht werden.

Zudem können verschiedene Infrastrukturen beispielsweise für die Entwicklung, den Vertrieb, den Betrieb, das Monitoring, die Wartung oder das Produkt- und Kundenmanagement genutzt werden.

Auch diese sollten möglichst weitgehend automatisiert nutzbar sein. Dabei sollten diese Automatisierungsansätze gleichermaßen dem konstruktiven und analytischen Quality Engineering unterliegen.

- DevOps für IoT (15 Min)

IoT-QE LZ 36 (K2) DevOps für IoT erklären können [15]

DevOps beschreibt einen Vorgehens-Ansatz aus den Bereichen der Softwareentwicklung und Systemadministration. DevOps ist ein Kofferwort aus den Begriffen Development (englisch für Entwicklung) und IT Operations (englisch für IT-Betrieb). DevOps soll durch gemeinsame Anreize, Prozesse und Werkzeuge eine effektivere und effizientere Zusammenarbeit der Bereiche Development, Operations und Qualitätssicherung ermöglichen. Mit DevOps sollen die Qualität der softwarebasierten Systeme, die Geschwindigkeit ihrer Entwicklung und Auslieferung und die Zuverlässigkeit ihres Betriebs durch ein Miteinander der beteiligten Teams verbessert werden.

In der Praxis bedeutet dies beispielsweise:

- für die Entwickler eine vermehrte Beschäftigung mit der Installation von virtuellen Maschinen und mit Aspekten der IT-Sicherheit oder mit der Planung und Durchführung von Auslieferungen.
- für die Administratoren die Beschäftigung mit Automatisierung in Kombination mit „Infrastructure as Code“, d.h. einer bestimmte IT-Infrastruktur, die Operations-Teams anstatt manueller Verfahren automatisch per Code verwalten und bereitstellen können, sowie der Umgang mit Versionsverwaltung und automatisierten Tests.
- für Entwickler und IT-Betrieb sich auf neue, bereichsübergreifende Key Performance Indicators (KPIs) und damit gemeinsame Anreiz-Metriken zu einigen und einzustellen.

Die an DevOps beteiligten Gruppen verfolgen auftragsgemäß Ziele, die teilweise in Konflikt zueinanderstehen:

- Entwickler möchten schnell Änderungen umsetzen,
- Tester möchten das Risiko von Abweichungen verringern und
- Administratoren möchten einen stabilen Betrieb gewährleisten.

Durch Etablierung einer zur Konsensbildung geeigneten DevOps-Kultur wird die Organisation in die Lage versetzt, schnell und effizient auf die Änderung von Rahmenbedingungen oder Geschäftszielen zu reagieren. Im Kontext von IoT bedeutet dies, das Silodenken aufzubrechen.

4.3 Weitergehende QE-Aktivitäten nach dem Rollout [30]

- Varianten in IoT Systemen (15 Min)

IoT-QE LZ 37 (K2) Die Bedeutung von Produkt- und Systemvarianten für IoT erklären können [15]

In der Systementwicklung werden verschiedene, insbesondere funktionale Ausprägungen eines Produkts als Varianten bezeichnet. Basis, aus der sie hervorgehen und unterscheiden sich durch variantenspezifische Eigenschaften, z.B. durch kundenspezifische Anpassungen. Varianten können unabhängig von einer zeitlichen Betrachtung entstehen.

Mit Versionen werden demgegenüber verschiedene Entwicklungsstände von Varianten bezeichnet, die ausgehend von einer definierten Basis in einer zeitlichen Entwicklung stehen. Die Versionsverwaltung erfasst die durchgeführten Änderungen. Zur Nachvollziehbarkeit werden alle Versionen in einem Archiv mit Zeitstempel und Benutzerkennung gesichert und können später wiederhergestellt werden.

Ein IoT System besteht aus einer Vielzahl von Komponenten (Hardware und Software), die unterschiedlichen Lebenszyklen und Entwicklungsgeschwindigkeiten unterliegen (siehe auch Kapitel 6). Die IoT Komplexität wird durch den Einsatz von Varianten und Versionen zusätzlich vervielfacht. Aufgrund der kontinuierlichen Entwicklung von IoT Systemen kommt es u.a. zur:

- Anpassungen an relevante Kundengruppen,
- Pilotierung neu entwickelter Komponenten oder zum
- Ersatz defekter oder veralteter Dinge.

Diese Weiterentwicklungen führen zu einem parallelen Betrieb von verschiedenen Verarbeitungsketten im gesamten IoT System. Das kann höhere Entwicklungs- und Wartungskosten und ein erhöhtes Risiko für die gesamte Ende-zu-Ende-Kette bedeuten. Ein Beispiel hierfür ist, dass Sensordaten unterschiedlicher Genauigkeit einer unterschiedlichen Verarbeitung bedürfen und Entscheidungen mit Machine Learning potentiell unterschiedlich beeinflussen können, womit sich das IoT Gesamtsystem verändern kann. Daher wird die nachhaltige Kontrolle der Komponentenschnittstellen inklusive einer präzisen Qualifizierung dringend empfohlen. Außerdem ist auf ein sorgfältiges Varianten- und Versionsmanagement und auf deren Dynamik im Betrieb zu achten.

- Betrieb von IoT Systemen (15 Min)

IoT-QE LZ 38 (K2) Die Bedeutung des Quality Engineering für die Betriebsphase bei IoT Systemen erklären können [15]

Der Wert der im Betrieb gewonnenen Daten liefert auch Vorgaben für das Quality Engineering in Bezug auf Inbetriebnahme, Instandhaltung, Änderung und Außerbetriebsetzung eines IoT Systems. Diese Arbeiten werden an einem Gesamt-System oder Teilen davon, typischerweise am „lebenden“ Objekt durchgeführt. Zur Vermeidung von Systemausfällen wird eine risikobasierte Planung empfohlen, die - je nach Kritikalität - das veränderte System auf unerwünschte Auswirkungen überprüft. Zugänge hierzu bieten beispielsweise:

- die Pilotierung auf einem Teilsystem oder
- die betriebsspezifischen Regressionstests.

Neben der obligatorischen Aufrechterhaltung des Betriebs sind insbesondere die den Änderungen nachfolgenden Auswirkungen auf die Datenverarbeitung, die Beeinträchtigungen beim Kunden und die Zielerreichung der gewünschten Änderung zu überprüfen. Insbesondere ist auf die Absicherung der Skalierung, Last/Performanz und Sicherheit zu achten. Bei einem akzeptablen Überprüfungsergebnis und Restrisiko kann mit einer umfassenden Inbetriebnahme einer Änderung begonnen werden. Dabei stehen typischerweise geforderte Qualitätsanforderungen und die Häufigkeit der Inbetriebnahme neuer Versionen oder Varianten in Konkurrenz. Insbesondere das Geschäftsmodell für ein IoT System oder Teilen davon kann eine hohe Frequenz der Systemaktualisierung erfordern, die im Widerspruch zu den geforderten Qualitätseigenschaften steht. Das Hintenanstellen von Qualitätseigenschaften kann eine valide Strategie darstellen, wenn erkannte Qualitätseinbrüche mit hoher Priorität ausgeglichen werden und der entstehende Schaden in einem kontrollierbaren Korridor verbleibt.

4.4 Beispiel “Ladevorgang eines Elektroautos” [10]

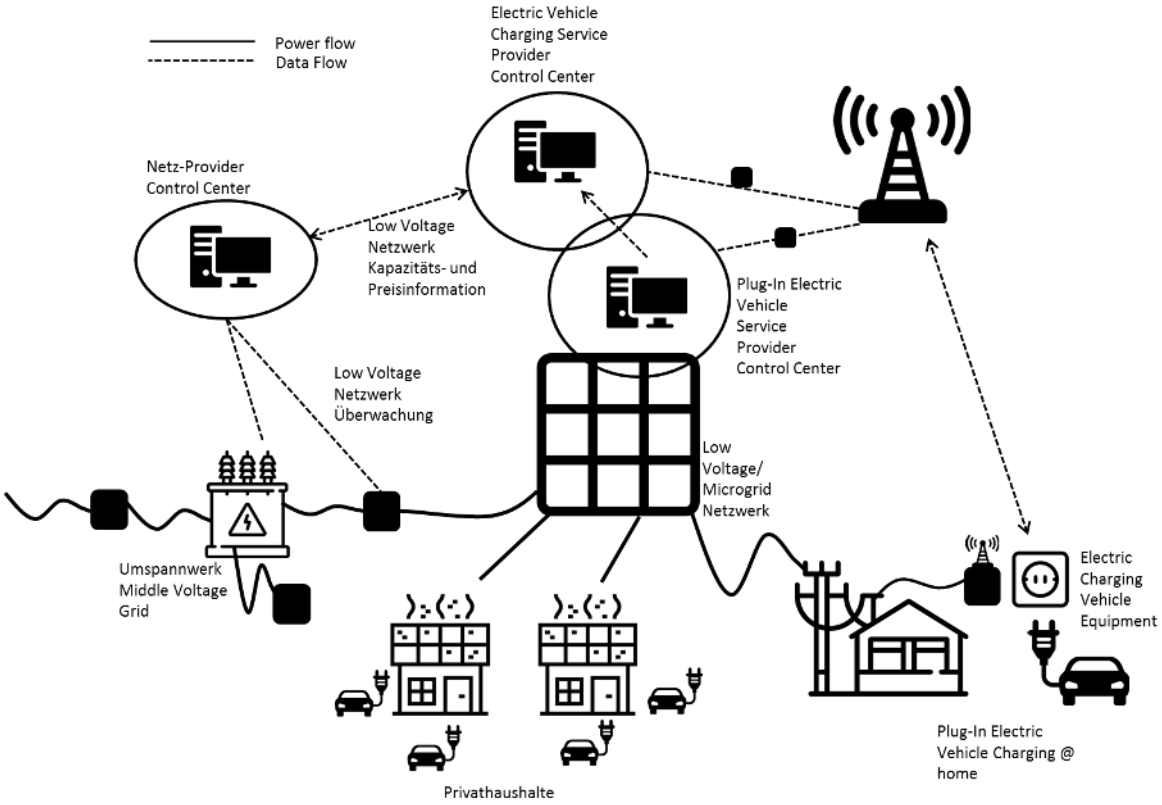

Abbildung 4: Ladevorgang eines Elektroautos (Icons made by Freepik, samshizone & Retinaicons from www.flaticon.com)

(Use Case „Plug-In Electrical Charging Vehicles and power feed in home scenario“ aus oneM2M Use Case Sammlung [oneM2M 16])

In diesem Anwendungsfall (siehe auch Abbildung) interagieren diverse Geräte und Systeme aus unterschiedlichen Industriedomänen miteinander und ihre Lebenszyklen sind typischerweise nicht synchronisiert.

Die Einhaltung der Qualitätsmerkmale im Echtbetrieb eines beteiligten Geräts oder Systems kann unmittelbar durch auftretende Qualitätsmängel eines anderen Systems beeinträchtigt werden. Diese Qualitätsmängel können beim Ersteinsatz aber auch erst nach längerer Nutzungszeit auftreten.

Deshalb ist es entscheidend, auch die nach dem Rollout liegenden Phasen (Operate und Update) des jeweiligen Lebenszyklus mit QE Aktivitäten zu begleiten. Etwa durch Monitoring ausgewählter Qualitätsmerkmale nach spezifischen Metriken. Bei Überschreitung von Schwellwerten können dann proaktiv korrigierende Maßnahmen eingeleitet werden.

5 Analytisches QE (inkl. Test)[240]

Begriffe

Fuzz Testing	Testtechnik, die mit Hilfe automatisch generierter und an ein Zielsystem versendeter anomaler ungültiger Nachrichtenfolgen, gebrochener Datenstrukturen oder ungültiger Daten Eingaben finden kann, die Störungen oder eine Verschlechterung von Dienstleistungen verursacht. ETSI TR 101 583 „Ein Testverfahren zur Entdeckung von Sicherheitsschwachstellen durch die massenhafte Eingabe von zufälligen Daten (Fuzz genannt) in die Komponente oder das System.“ [ISTQB 16]
Konformität	Die Fähigkeit eines Softwareprodukts, anwendungsspezifische Normen oder Vereinbarungen oder gesetzliche Bestimmungen und ähnliche Vorschriften zu erfüllen. [ISO 9126]
Things under Test	Erweiterung des Begriffs System under Test [ISTQB 17] auf ein SUT, welches cyperphysische Komponenten integriert.
Datenqualität	Bewertung von Datenbeständen hinsichtlich ihrer Korrektheit, Relevanz und Verlässlichkeit, sowie ihrer Konsistenz und Verfügbarkeit auf verschiedenen Systemen.

IoT-QE LZ 39 (K1) Die Notwendigkeit von Monitoring im Betrieb von IoT Systemen kennen [verteilt auf Kapitel]

IoT-QE LZ 40 (K2) Die Herausforderungen verteilter Tests für IoT Systeme erklären können [verteilt auf Kapitel]

5.1 Einleitung [10]

IoT-QE LZ 41 (K2) Die besonderen Herausforderungen beim Testen von IoT Lösungen wie ihre Offenheit, Verteiltheit, Dynamik, Skalierung und Varianz erläutern können [10]

IoT Lösungen zeichnen sich allgemein durch Offenheit, Verteiltheit, Dynamik, Skalierung und eine lange Betriebslaufzeit aus. Aus diesen Gründen werden daher neue Ansätze der analytischen Qualitätssicherung erforderlich.

Beispielsweise muss die entwicklungsbegleitende Qualitätssicherung in der Laufzeit verlängert werden. Entlang des DevOps-Ansatzes sind Testen, Runtime-Monitoring und Zertifizierung zu verknüpfen und somit neu zu bedenken. Für IoT ist es auch spezifisch, dass sich nach den üblichen Abnahme- bzw. Systemtests eine weitere Teststufe „Betrieb“ anschließt, die mögliche spätere Veränderungen berücksichtigt, z.B. Schnittstellenerweiterungen, Wechsel von Systemteilen oder neu diagnostizierte Schwachstellen.

Eine besondere Schwierigkeit ergibt sich aus der Fragestellung der Haftung bei Schadensfolgen nach Zweckentfremdung oder Sicherheitsvorfällen, insbesondere, wenn im Rahmen einer Zertifizierung zu einem bestimmten Zeitpunkt auf eine Selbsterklärung des Herstellers oder Betreibers zurückgegriffen wird.

5.2 Für IoT spezifische Testvorgehen und Teststufen [20]

IoT-QE LZ 42 (K2) Für IoT spezifische Testvorgehen und Teststufen erläutern können [20]

Gerade für IoT Produkte ist effiziente und effektive Testbarkeit eine wichtige Voraussetzung für die Qualität des Produkts. Daher ist die Testanalyse und Testplanung in jeder Phase des Lebenszyklus (also auch die Wartungsphasen) durchzuführen. In den Wartungsphasen sind neben dem traditionellen Monitoring auch Predictive Maintenance Aspekte abzusichern. Eine frühzeitige Definition des

Testsystems erleichtert das Erstellen von Testschnittstellen im System. Alle Testaktivitäten werden dabei in jeder Phase des Lebenszyklus überprüft und auf die Erfordernisse der jeweiligen Phase angepasst. Zunehmend wichtiger werden insbesondere die Tests des Verhaltens in der Produktivphase, d.h. im Betrieb, z.B. durch den Betreiber nach Updates oder Wartungsintervallen.

Teststufe	Beispiel	Anmerkung
Abnahme-/System-Test oder Zertifizierung nach allgemeinen Prüf- und Integrationsanforderungen	Informationssicherheit Konformität zu unterstützten Protokollen Konformität zu standardisierten Abläufen	Abhängigkeiten von Nutzungsprofilen sind zu beachten (z.B. private vs. Industrielle Anwendung, militärische Anwendung). Die Konformität bezieht sich i.d.R. auf Normen und normenartige Dokumente.
Integrationstest zur Einbettung des Testobjekts in seine (Test-) Umgebung	Kompatibilität Interoperabilität	Kann stark von den spezifischen Einsatzszenarien des Testobjekts abhängen. Systemumgebung kann hohe evtl. auch nicht vollständig vorhersehbares Verhalten aufweisen (z.B. zukünftige neue Services). Umgebung kann durch Simulation erzeugt werden.
Betriebs-/Diagnose-Test in der Einsatzumgebung, ggfs. auch in der Produktivphase (z.B. passive Tests zum Monitoring des Verhaltens im Betrieb).	Vorhandensein erforderlicher Dienste (z.B. Produktivtests des Herstellers), betriebserhaltende Testszenarien	Da die Systemumgebung nicht vollständig vorhersehbares Verhalten aufweisen kann und sich ändern kann, sind auch Tests bzw. Analysen im Betrieb erforderlich. Auslöser sind z.B. neue Schwachstellen oder Updates, die im Labor nicht darstellbar sind. Keine „kontinuierlichen“ Test ohne Zustimmung des Nutzers. Es müssen ggf. damit verbundene zusätzliche Sicherheitsrisiken (IT-Sicherheit und Safety) betrachtet werden.

Tabelle 2: Teststufen

Der fundamentale Testprozess [ISTQB 17] wird im Kontext von IoT um die Phasen für die Laufzeit durch Monitoring & Watch Dogs erweitert, d.h. das System wächst zunehmend mit dem Testsystem zusammen:

- Planung und Steuerung: als Teil des Life Cycle Management
- Analyse und Entwurf: Risikoanalyse ergänzen
- Realisierung und Durchführung: Automatisierung und Laufzeit
- Bewertung und Bericht: fortlaufend
- Abschluss: erst mit Terminierung einer IoT Lösung

5.3 Testziele, Priorisierung und Risikoanalyse [75]

IoT-QE LZ 43 (K3) Testziele für IoT definieren und deren Priorisierung durchführen können [30]

Zu den Testzielen im Kontext von IoT gelten zunächst die Testziele entlang der Qualitätskriterien nach ISO/IEC 25010 (siehe Kapitel 3.1). Zusätzlich finden besondere Beachtung Interoperabilität, IT-Sicherheit und Performanz. Ebenfalls ist der Zusammenhang von funktionaler Sicherheit und Informationssicherheit zu berücksichtigen.

Die Priorität von Testzielen orientiert sich an der Priorität der damit zu überprüfenden Qualitätsmerkmale. Testziele und deren Priorisierung sind kontinuierlich über den gesamten Lebenszyklus eines IoT Systems zu bewerten und gegebenenfalls anzupassen bzw. zu ergänzen.

Neben den Funktionalen Qualitätsmerkmalen gewinnen folgende Qualitätsmerkmale für IoT Systeme eine erhöhte Bedeutung für die Priorisierung der Testziele im Vergleich zu „klassischen“ Systemen:

Grund	Qualitätsmerkmal
spezifische (verteilte) Architekturen	Interoperabilität Performanz und Leistungsfähigkeit Anpassbarkeit Robustheit und Resilienz
verknüpfte Lebenszyklen und Interdisziplinarität	Kompatibilität Wartbarkeit Übertragbarkeit
verknüpfte und umfassende Geschäftsprozesse, welche mit IoT-Systemen abgebildet werden	Funktionale Sicherheit (Safety) IT-Sicherheit (Security) Vertraulichkeit (Privacy) Gebrauchstauglichkeit Ethische Aspekte

Tabelle 3: Verstärkt zu testende Qualitätsmerkmale und ihre Motivation

Es ist hilfreich, die IoT Prüfanforderungen und Testziele in die Gruppen Prozess, System/Komponente und Kommunikationsprotokoll zu untergliedern.

IoT-QE LZ 44 (K3) Risikobasierte Priorisierung von Testzielen durchführen können [30]

Bei sicherheitskritischen IoT Systemen ist eine besondere Betrachtung der Kritikalität von IoT Systemen durch eine Risikoanalyse für das ganze IoT System erforderlich. Die Ergebnisse dieser Risikoanalyse dienen der Herleitung und Priorisierung von Testzielen. Als Referenz der Einbeziehung einer Risikoanalyse in den Testentwicklungsprozeß dient ETSI EG 203 251 (Risk-based Security Assessment and Testing Methodologies, Kap. 7.2 ff).

Übung zu Testzielen für IoT-Systeme

Die Kursteilnehmer bilden Gruppen zu je etwa 3-4 Mitgliedern und lösen folgende Aufgabenstellungen an Hand eines konkreten Beispiels (z.B. Smart Home):

- Welche Testziele gibt es? Wie lassen sich diese gruppieren?
- Welche Prioritäten ergeben sich aus der Risikoanalyse?

Anschließend werden die Ergebnisse von den Gruppen vorgestellt und im Plenum diskutiert.

5.4 Testbarkeit und Testautomatisierung [15]

- Besonderheiten des IoT Testens (10 Min)

IoT-QE LZ 45 (K2) Die Besonderheiten beim IoT Testen benennen und Beispiele für IoT Tests auf verschiedenen Ebenen erläutern können [10]

Besonderheiten des IoT Testens in Ergänzung zu „klassischem“ Software- und Protokoll-Testen sind in der folgenden Tabelle aufgeführt [Schieferdecker 16].

Perspektive	**Besonderheiten**	**Testvarianten neben klassischem Software- und Protokoll-Testen**
Applikationen (Analytics, Visualisierung und Steuerung)	Hoher Stellenwert der Sicherheit und Gebrauchstauglichkeit	Gebrauchstauglichkeitstest, GUI- und (mobile) App Testing Performance und Scalability Testing Security Testing Crowd Testing
IoT Schicht (Plattformen und Schnittstellen, Computation-, Aggregation- und Storage-Dienste)	Hoher Stellenwert der Sicherheit, Konformität/Interoperabilität und Datenqualität	Real-Time Testing GUI Testing (für Management-Software) Security Testing
Physikalische Perspektive **(Geräte, Gerätekonnektivität)**	Hoher Stellenwert der Sicherheit, Konformität Konnektivität und Verfügbarkeit; Hoher Stellenwert der Robustheit, physikalische Sicherheit, Resourcenverwendung	Performance und Scalability Testing Services Testing (Connectivity) Security Testing Embedded Systems Testing Robustness-Testing (Physical) Security Testing Performance-Testing (Resourcen)

Tabelle 4: Besonderheiten des IoT Testens

- Testautomatisierung (5 Min)

IoT-QE LZ 46 (K2) Die Notwendigkeit der Testautomatisierung für den IoT Test erläutern können [15]

Um einen effektiven Test zu gewährleisten ist ein hoher Grad an Testautomatisierung über alle Testphasen zu erlangen, da

- Sicherung der Qualität im Lebenszyklus mit einem hohen Grad an Testregression verbunden ist,
- der Faktor time to market eine wesentliche und kontinuierliche Bedeutung hat,
- die Komplexität und Dynamik des Systemkontexts für das IoT Produkt hoch ist,
- manuelle Vorgänge ein höheres Fehlerrisiko aufweisen als automatisierte Vorgänge.

5.5 Testprozess und Testarchitektur [15]

- IoT Testarchitekturen (15 Min)

IoT-QE LZ 47 (K2) IoT Testarchitekturen und typische IoT Testobjekte erläutern können [15]

Da IoT Systeme verteilte Systeme sind, kommen verteilte Testarchitekturen und entsprechende Prozessstrategien zum Einsatz, unter anderem Verbesserung der Effizienz durch Virtualisierung über das gesamte Testsystem. Für IoT Testsysteme sind folgende Testarchitekturen typisch (für Beispiele siehe z.B. [Jäkel 17]):

- Geräte basierte IoT Testarchitektur (z.B. für das Testen von Retroboxen oder Gateways), entsprechend der AIOTI IoT Schicht.
- Dienst basierte IoT Testarchitektur (z.B. für das Daten-orientierte Testen von Dashboards in der Cloud), entsprechend der AIOTI Applikationsschicht.
- Infrastruktur basierte IoT Testarchitektur, (z.B. für das Testen von oneM2M Funktionselemente), entsprechend der AIOTI Netzwerkschicht.

SUT und Testsystem können für verschiedene Testanforderungen ihre Rollen vertauschen. D.h. es kann sinnvoll sein, eine Komponente in einem Fall als SUT und im anderen Fall als Testsystem einzusetzen, welches das SUT stimuliert.

Es ist dafür zu sorgen, dass die Testumgebung lückenlos in die Prozess- und Toolumgebung integriert ist, typischerweise über den gesamten Lebenszyklus und in eine DevOps Prozessumgebung.

IoT-QE LZ 48 (K2) Wesentliche Aspekte der IoT Testarchitektur erläutern können [15]

Wesentliche Aspekte der Testautomatisierungsarchitektur sind [ISTQB 16]:

- Das Verständnis der Technologien des System under Test (SUT) sowie von dessen Integration in das Testautomatisierungssystem (TAS). Spezifisch für IoT:
    - IoT Testschnittstellen typischerweise auf Protokoll Level und Service Level.
    - für eine zukunftssichere Implementierung bedarf es eine sorgfältige Analyse der Testschnittstellen.
    - typische Interaktion zwischen SUT und TAS sind ereignisgetrieben und Peer-to-peer.
    - Systemgrenzen des SUT sind entscheidend für die Effizienz und Effektivität des TAS, diese können für verschiedene Testanforderungen variieren.
- Das Verständnis der Testumgebung. Spezifisch für IoT:
    - Simulation von Testumgebungen hat für IoT eine wesentlich größere Bedeutung als im „klassischen“ Test.
    - Mögliche variable Systemgrenzen zwischen SUT und TAS müssen in der Testarchitektur berücksichtigt werden.

- die Wartbarkeit der Testumgebung spielt eine wichtige Rolle.
- die reale Einsatzumgebung des IoT Produkts ist zu berücksichtigen.
- die Integration der Testumgebung in DevOps Tools ist zu berücksichtigen.

- Zeit, Aufwand und Komplexität der Implementierung (Planung und Controlling).
- Benutzerfreundlichkeit der Implementierung (Design ausgerichtet auf die Benutzerprofile der Tester als Anwender). Spezifisch für IoT:
  - da IoT Entwicklungsprojekte hochgradig interdisziplinär sind, müssen Anforderungen an die Benutzerfreundlichkeit aller beteiligten Rollen Beachtung finden.

5.6 Testmethoden [95]

- Wichtige IoT Testmethoden (20 Min)

IoT-QE LZ 49 (K2) Nutzbarkeit und Grenzen klassischer Testmethoden für IoT Systeme erläutern können [20]

Testmethoden bzw. Testautomatisierungsmethoden und Testwerkzeuge für IoT sind nicht grundsätzlich neu, sondern eine spezielle Auswahl etablierter Ansätze unter Berücksichtigung der IoT spezifischen Eigenschaften des SUT und Besonderheiten aus der Anforderungsanalyse. Für die praktische Anwendung spielen vor allem Methoden für Interoperabilität, Sicherheit und Leistungsfähigkeit eine herausragende Rolle. Dabei wird ein hoher Grad an Testautomatisierung sinnvoll ergänzt durch manuelle Tests insbesondere für Explorative Tests (z.B. für mobile Geräte in unterschiedlichen Umgebungen).

Modellgetriebene Analysestrategien / Modellbasiertes Testen stellen optimale Analysemethoden (auch für die Definition der Systemgrenzen des SUT) zur Verfügung und sind eine wichtige Best Practice für Tests von IoT Systemen. Online MBT [ISTQB 17] ist eine Antwort auf den sich weiterentwickelnden Testentwurf entlang der Dynamik in IoT.

Die analytische QE wird durch eine Kette von Methoden und Werkzeugen begleitet. Eine Vielzahl von Werkzeugen steht zur Verfügung, teilweise auch als Open Source Werkzeug. Des Weiteren existieren für spezifische Domänen und deren Protokolle standardisierte Testbeschreibungen (z.B. ETSI in der Telekommunikation, Automotive und Autosar), typischerweise in formalen Beschreibungen wie TTCN-3.

Während der Ausführung von Betriebs-/Laufzeittests kann es erforderlich sein, dass zu einem späten Zeitpunkt die zugehörigen Testerwartungen (test purposes) verfeinert werden.

- Sicherheitstest (20 Min)

IoT-QE LZ 50 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Sicherheit und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [20]

Sicherheitstests haben für IoT eine herausragende Bedeutung. Die Aspekte der IT-Sicherheit werden hier u.a. wegen fehlender ökonomischer Anreize und fehlender Expertise in disziplinübergreifenden Projekten nicht im erforderlichen Umfang berücksichtigt. Zudem werden verschiedene Komponenten von IoT-Lösungen häufig von verschiedenen Projektteams entwickelt, so dass eine Gesamtbetrachtung der Sicherheit häufig fehlt.

Daher eigenen sich IoT Systeme hervorragend zum Aufbau von Botnetzen. Ein Botnetz ist eine Gruppe automatisierter Schadprogramme. Diese laufen auf vernetzten Rechnern, deren Netzwerkanbindung sowie lokale Ressourcen und Daten ihnen, ohne Einverständnis des Eigentümers, zur Verfügung stehen. Über Botnetze lassen sich wiederum zahlreiche Angriffe, z.B. Denial of Service Angriffe, durchführen. Bei Sicherheitstest im IoT-Umfeld müssen alle Architekturebenen in ausreichendem Umfang berücksichtigt werden. Zudem ist ein gesamtheitlicher Ansatz für den

Sicherheitstest notwendig. IOT Sicherheit ist nicht nur Geräte-Sicherheit! Die Absicherung einer einzelnen Teilkomponente ist nicht ausreichend zur Gewährleistung der Sicherheit des Gesamtsystems.

Im Folgenden werden die wichtigsten Testschwerpunkte aus verschiedenen Perspektiven dargestellt:

Perspektive	Angriffsvektoren	Methoden
Applikationen (Analytics, Visualisierung und Steuerung)	- Mobile Applikationen - Webapplikationen - Daten- und Kontrollströme	- Test auf Webschwachstellen - Test auf sensitive Daten bei mobilen Geräten - Datenstromanalyse / Proxy / Man in the Middle-Angriffe - Denial of Service - Suche nach logischen Schwachstellen im Gesamtkonzept
IoT-Schicht (Plattformen und Schnittstellen, Computation, Aggregation- und Storage-Dienste)	- Cloud-Dienste - Webinterfaces (Konfigurationsinterfaces) der Geräte - Daten- und Kontrollströme - Zugriffs- und Rechtemanagement - Update Mechanismen (over the air updates) - Lokalisierungs Service	- Test auf Webschwachstellen - Datenstromanalyse / Proxy / Man in the Middle-Angriffe - Suche nach Protokollschwachstellen oder Fehlkonfigurationen, z.B. unverschlüsselte Kommunikationsverbindungen - Denial of Service - Suche nach logischen Schwachstellen im Gesamtkonzept - Spoofing von Endgeräten
Physikalische Perspektive **(Geräte, Gerätekonnektivität)**	- Backend APIs - Daten- und Kontrollströme - Verschlüsselung - Sonstige Kommunikation zwischen IoT-Schicht und Netzwerk-Schicht - Gerätespeicher und Speichererweiterung (z.B. SD-Karten) - Firmware der Geräte - Physikalische Geräteschnittstellen - Netzwerkschnittstellen der Geräte	- Datenstromanalyse / Proxy / Man in the Middle-Angriffe - Suche nach logischen Schwachstellen im Gesamtkonzept - Test auf Webschwachstellen - Ausnutzung von bekannten Protokollschwachstellen - Denial of Service - Suche nach sensitiven Daten (Passwörter, Schlüssel, …) und Manipulation von Daten - Firmwareanalyse - Seitenkanalangriffe

	- Physikalische Manipulation oder Diebstahl der Geräte	- Suche nach logischen Schwachstellen im Gesamtkonzept

Tabelle 5: Testschwerpunkte für Sicherheitstests

- Interoperabilitätstest (15 Min)

IoT-QE LZ 51 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Interoperabilität und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [15]

Ein Typ des funktionalen Tests ist der Interoperabilitätstest. Er bewertet die Fähigkeit des Softwareprodukts mit ein oder mehr spezifizierten Komponenten oder Systemen zu interagieren. Bei dieser Art der Testdurchführung werden zwischen zwei gezielt ausgesuchten (Client/Server) Systemen Daten ausgetauscht. Bei diesen Daten handelt es sich um korrekte, entsprechend dem Protokoll definierte Eingaben, aber auch um bewusst fehlerhafte Eingaben, die die Stabilität (Reaktion) des Systems testen sollen.

Hinweis: In der Regel werden die Interoperabilitätstests nach erfolgreichem Abschluss von Konformitätstests (z.B. durch TTCN-3 Technologie) durchgeführt.

Wichtige Methoden des Interoperabilitätstests sind:

Testobjekt	**Methoden**	**Was wird getestet?**
Technische Interoperabilität	Basis-Tests von Konnektivität und Kommunikationsprotokollen	Koppelung von Hardware/Software Komponenten um eine Basis-Kommunikation zu gewährleisten
Syntaktische Interoperabilität	Gezielte Überprüfung von Nachrichtenformaten bzw. der Syntax abstrakter Datenformate, Anwendung von Encoder/Decoder	Einhaltung der Syntax von z.B. HTML, XML oder ASN.1 Datenstrukturen
Semantische Interoperabilität	Durchführung von Fallbeispielen und Nutzerszenarien, ggf. unter Einbeziehung von standardisierten Use Case Katalogen	Es wird geprüft ob die Implementierungen der interoperierenden Komponenten/Systeme einer gemeinsamer Interpretation folgen
	(Standardisierter) Katalog von Testzielen, die in tabellarischer Form Angaben zu Konfiguration, sequenziellen Abläufen von Triggern und Beobachtungen der beteiligten Komponenten oder Systemen vorgeben	Standardisierte Testziele. Plugtests

Tabelle 6: Wichtige Methoden der Interoperabilitätstests

Plugtests™ sind Konvents, auf denen Hersteller von elektronischem Equipment oder Software die Interoperabilität ihrer Produkte im Zusammenspiel mit Produkten anderer Hersteller testen. Sie sind nichtöffentlich, dauern häufig drei bis fünf Tage und werden von einer neutralen Institution (z.B. ETSI) vorbereitet und betreut. Getestet wird ein (standardisierter) Katalog von Testzielen, die in tabellarischer Form Angaben zu Konfiguration, sequenziellen Abläufen von Triggern und Beobachtungen der beteiligten Komponenten oder Systemen vorgeben.

- Performanz Test (20 Min)

IoT-QE LZ 52 (K2) Die besonderen Anforderungen an das Testen von IoT Lösungen auf Performanz und die Anwendungen entsprechender Testmethoden auf unterschiedlichen Ebenen der IoT Architektur erläutern können [20]

IoT Systeme sind verteilte Systeme, die auf der Applikationsschicht teilweise enorme Datenmengen verarbeiten. Für eine effiziente Verarbeitung dieser heterogenen Datenströme ist eine geeignete Architektur zu wählen (z.B. unter Integration von Edge Computing Bestandteilen). Als qualitätssichernder Aspekt ist eine adäquat skalierte Leistungsfähigkeit der einzelnen IoT Komponenten in Entwicklung sicherzustellen und im Betrieb zu monitoren und zu testen. Auf Grund der für Performanz Tests typischerweise erforderlichen hohen Transaktionsvolumina hat Automatisierung hier eine entscheidende Bedeutung.

Beispiele von Methoden und Werkzeuge für Performanz Tests [ISTQB 16]:

Methoden	Werkzeug-Beispiele	Erläuterungen
Dynamische Analyse	Dynamische Analysewerkzeuge decken Fehlerzustände auf, wie sie lediglich zur Laufzeit eines Programms sichtbar werden, also z.B. Zeitabhängigkeiten und Speicherengpässe.	Diese werden typischerweise im Komponenten- und Komponentenintegrationstest sowie im Rahmen der Tests der Middleware verwendet.
Performanz Test / Lasttest / Stresstest	Performanztestwerkzeuge überwachen und protokollieren, wie sich ein System unter verschiedenen simulierten Benutzungsbedingungen verhält, hinsichtlich Anzahl konkurrierender Nutzer, Hochlauf-/Anlaufverhalten (ramp-up pattern) sowie Häufigkeit und relativem Anteil von Transaktionen. Die Last wird durch Erzeugen virtueller Nutzer simuliert, die einen ausgewählten Satz an Transaktionen durchführen, verteilt auf verschiedene Testmaschinen, allgemein bekannt als Lastgeneratoren.	Softwaretest, mit dem eine zu erwartende, auch extreme Last auf dem laufenden System erzeugt und das Verhalten desselbigen beobachtet und untersucht wird.
Monitoring	Testmonitore analysieren, verifizieren und zeichnen kontinuierlich die Verwendung von spezifischen Systemressourcen auf und geben Warnungen zu möglichen Problemen bei der Erbringung von Diensten aus.	z.B. zum Simulieren von Ereignisdaten

Tabelle 7: Methoden und Werkzeuge für Performanz Tests

- Produktzertifizierung (20 Min)

IoT-QE LZ 53 (K2) Die Herausforderungen bei der Prüfung auf Konformität und Zertifizierung erklären können [20]

Wegen der Variantenvielfalt von IoT Geräten und IoT Diensten hat die Zertifizierung eine herausragende Bedeutung. D.h. die Prüfung von Produkten, Prozessen oder Dienstleistungen, um sicherzustellen, dass sie den in Normen und weiteren normativen Dokumenten festgelegten Anforderungen entsprechen [DIN EN ISO/IEC 17065:2013-01]. Die Inspektions- und Prüfergebnisse eines Prüflabors werden hierbei von einer Zertifizierungsstelle evaluiert und gegebenenfalls ein Zertifikat oder zumindest öffentlich empfohlenes Gütesiegels ausgestellt. Mögliche Varianten der Zertifizierung sind auch Prüfungen ohne Begleitung durch eine Zertifizierungsstelle bzw. die sog. „Selbstzertifizierung“ bei der die Prüfung ausschließlich durch den Hersteller erfolgt.

Grundlagen der Zertifizierung bilden Leit-, Richtlinien, Normen und Standards. Gegenwärtig gibt es jedoch keinen „IoT Standard“, sondern allenfalls eine Ansammlung von noch zu unreifen übergeordneten Normen.

Hauptfokus für IoT ist die Prüfung der IT-Sicherheit:

- funktionale Security-Anforderungen
- Stabilität
- Konformität und Fehleranfälligkeit einzelner IoT typischer Kommunikationsprotokolle

Deren Prinzipien unterscheiden sich nicht grundsätzlich von den IT-Sicherheitsprinzipien aus dem IT-Umfeld. Die Prüfkriterien sind jedoch auf das Umfeld und die speziellen Risiken abzustimmen. Insbesondere ist zu beachten, dass in der für IoT typischen langen Betriebsphase neue Sicherheitsvorfälle bekannt werden können und es zu zahlreichen Patches kommen kann, die die Frage nach der Gültigkeitsdauer eines Zertifikats aufwerfen und die Durchführung neuer bzw. eine Wiederholung von bestehenden Prüfungen erforderlich machen.

Auf diese generischen Standards kann zurückgegriffen werden [Wardaschka 17]:

- Protokollspezifische IoT Normen und -Standards
- Standards mit funktionalen Anforderungen
- Standards mit nicht-funktionalen Anforderungen (Performanz, Verfügbarkeit, Zuverlässigkeit, Dokumentation, (Arbeits-)Prozesse)
- Zu entwickelnde IoT spezifische Standards

5.7 Zusammenfassung [10 Min]

6 Lifecycle [45]

Begriffe

IIoT	Industrielles Internet der Dinge bedeutet die Anwendung des IoT auf die verarbeitende Industrie (Industrielles Internet oder Industrie 4.0)
Industrie 4.0	Industrie 4.0" steht für ein "Zukunftsprojekt" der deutschen Bundesregierung und bezeichnet die „vierte industrielle Revolution“. Wesentliche Merkmale der vierten industriellen Revolution sind Individualisierung bzw. Hybridisierung der Produkte und die Integration von interdisziplinären Stakeholdern und Geschäftsprozessen.

Als abschließende Betrachtung wird im vorliegenden Kapitel noch einmal der Bogen über den Zusammenhang der Aktivitäten im gesamten Lebenszyklus geschlagen. Im IoT Kontext treffen unterschiedlichste Lebenszyklen von IoT Geräten, Software wie Applikations-Services oder Infrastruktur wie Netzwerke aufeinander. Einen allgemein gültigen IoT-Lebenszyklus gibt es daher in den referenzierten Normen (siehe auch Kap. 1) aktuell nicht. Dieser Lehrplan bezieht sich deshalb auf eine vereinfachte Lebenszyklus-Darstellung aus ISO/IEC CD 30141 und die darin aufgeführten Phasen Initiate, Build, Develop, Operate, Update und Decommision (siehe auch Kap. 3).

6.1 Im IoT-Kontext verknüpfte Lebenszyklen mit ihren Phasen und ihre Bedeutung aus QE-Sicht [15]

IoT-QE LZ 54 (K2) Die Bedeutung der im IoT Kontext verknüpften Lebenszyklen für das QE verstehen [15]

Unter Lebenszyklus wird je nach Kontext Unterschiedliches verstanden. Es gibt domänenspezifische Lebenszyklen wie etwa nach RAMI 4.0 im Manufactoring Umfeld oder ISO 26262 im Automotive Umfeld, aber auch generische Lebenszyklen wie nach ISO/IEC CD 30141 [ISO/IEC CD 30141].

Gemäß ISO/IEC CD 30141 besteht ein Produkt Lebenszyklus aus den Phasen Initiate, Build, Develop, Operate, Update und Decommision. In Tabelle 8 werden diese Phasen beschrieben. Für jede dieser Phasen ist Quality Engineering relevant.

Phase	Beschreibung
Initiate	Die Entwicklung oder die Einbindung einer im IoT Kontext einzusetzenden Entität (z.B. Gerät, Service, Infrastruktur, Daten) wird angestoßen.
Build	Die Entität wird konzipiert und notwendige Voraussetzungen wie IoT Infrastruktur und Organisationsstrukturen werden geschaffen.
Develop	Die Entität wird entwickelt und in die IoT Zielumgebung ausgerollt.
Operate	Die Entität wird in ihrer IoT Zielumgebung betrieben und unterstützende Kundendienstleistungen erbracht.
Update	Die Entität wird gewartet wobei Korrekturen aber auch funktionale wie nicht-funktionale Verbesserungen ausgerollt werden.
Decommision	Die Entität hat das Ende ihres Lebenszyklusses erreicht und wird außer Betrieb genommen.

Tabelle 8: Phasen im IoT Produkt Lebenszyklus nach ISO/IEC CD 30141

Im typischen IoT Umfeld befinden sich beteiligte Entitäten durchaus in unterschiedlichen Phasen ihres jeweiligen Lebenszyklusses. Für Industrie 4.0 bzw. Smart Manufactoring Szenarien wurde das in RAMI 4.0 explizit adressiert. Für das Quality Engineering ist daher im IoT Umfeld die Abstimmung der QE Aktivitäten der sich in unterschiedlichen Phasen ihres Lebenszyklus befindenden Entitäten von großer Bedeutung. Gleichzeitig leitet sich daraus auch ab, dass wie bereits in Tabelle 8 dargestellt alle Phasen im IoT-Lebenszyklus mit QE Aktivitäten zu unterstützen sind. Die Planung und Durchführung von Verifikations- und Validationsaktivitäten benötigen aufgrund der hohen Komplexität im IoT und IIoT Umfeld die Auswahl der passenden Quality Engineering Maßnahmen/Methoden durch spezifische Fachkräfte in jeder Phase.

6.2 Die besondere Bedeutung der Interdisziplinarität für den IoT-Lebenszyklus verstehen [30]

- Die interdisziplinäre Natur des IoT-Lebenszyklus (15 Min)

IoT-QE LZ 55 (K2) Die interdisziplinäre Natur des IoT-Lebenszyklus verstehen [15]

Die unterschiedlichen Perspektiven im IoT-Lebenszyklus beziehen sich auf die jeweils zu betrachtende Phase und welche Daten und Services in dieser Phase Wem zur Verfügung gestellt werden. Generell werden die gesammelten Daten an einen cloudbasierten Service gesendet, dort mit anderen Daten aggregiert (Daten/Informationen werden aus unterschiedlichen Quellen zusammengefasst) und in Interaktion mit dem Endbenutzer genutzt.

Was daran interdisziplinär ist, zeigt sich in der enormen Vielfalt der Möglichkeiten in Bezug auf:

- Daten sammeln, aggregieren, sicher übertragen,
- die Auswahl an Lösungen für Gateways,
- genutzte Software und Tools für IoT Applikationen,
- verschiedenste Services sowie
- diverse IoT Plattformen und -Frameworks.

Diese Vielfalt an technischer Infrastruktur wird durch die am Lebenszyklus beteiligten unterschiedlichen Branchen samt der diversen Experten ergänzt und erhöht damit die interdisziplinäre Natur des IoT-Lebenszyklus.

Hinzu kommt die interdisziplinäre Abstimmung bezüglich Produkt und Prozessen auf technischer, organisatorischer und Management Ebene (Geschäftsebene) in jeder Phase.

- Drittbeteiligte im IoT-Lebenszyklus und ihre Bedeutung (15 Min)

IoT-QE LZ 56 (K2) Bedeutung von Drittbeteiligten im IoT Kontext verstehen [15]

Grundsätzlich sind im IoT-Lebenszyklus in allen Phasen erforderliche Prozesse, Rollen und Verantwortliche entsprechend des Services, der Daten, der technischen Infrastruktur der jeweiligen Produkte, Geräte und Branche verankert. Weiteren Einfluss auf die Quality Engineering Aktivitäten nehmen Drittbeteiligte.

Drittbeteiligt können ganze Allianzen und Ökosysteme sein. Strategische Allianzen sind in den Bereichen Produktentwicklung, Vertrieb, Technologie, Service sowie Research & Development üblich, damit Unternehmen ihr Produktportfolio sowie den Status der Digitalisierung von Marketing & Vertrieb IoT tauglich gestalten können.

Unter Ökosystem ist hier ein dynamischer Komplex von Gemeinschaften zu verstehen, die als funktionale Einheit in Wechselwirkung stehen – ein Beziehungsgefüge der Gemeinschaften im IoT-Lebensraum. Beispielsweise Gemeinschaften von Zulieferern, Technologieunternehmen, Franchise-Nehmern, Auslieferern.

Weniger abstrakt gibt es im IoT Life Cycle maßgebliche Drittbeteiligte, die direkten Einfluss auf QE Maßnahmen haben. Sie initiieren Anforderungen und Vorschriften, die wiederum über den gesamten

IoT-Lebenszyklus bei Planung, Durchführung, Kontrolle und Optimierung von Quality Engineering Maßnahmen zur berücksichtigen sind. Zu diesen Drittbeteiligten gehören:

- Gesetzgeber
- Normierungsgremien
- Prüfstellen/Zertifizierungsstellen
- Betriebsräte/Personalräte
- Verbände/Vereinigungen

Am Beispiel der Produkthaftung sind QE-Maßnahmen direkt durch mehrere dieser Drittbeteiligten beeinflusst. Gesetze enthalten Anforderungen an die Eigenschaften des Produkts, welches in Verkehr gebracht wird. Unabhängig davon, ob das Produkt ein Service, ein Gerät oder ein Tool ist, bestehen Anforderungen in Bezug auf die Eigenschaften und Sicherheit des Produkts. Über den gesamten Lebenszyklus des Produkts müssen diese Anforderungen eingehalten werden und explizite Nachweise hierzu geschaffen werden. Mit Hilfe von Normen, Standards, Zertifikaten, die konkrete Quality Engineering Maßnahmen einfordern werden diese Nachweise geschaffen.

Anhang A Begriffe / Glossar

Anhang B Referenzen

[Anderson 11] Anderson, Michael; Anderson, Susan Leigh (Hrsg.), Machine Ethics (2011)

[AIOTI 16] AIOTI WG03 – IoT Standardisation, High Level Architecture (HLA) – Release 2.1 (2016) https://aioti.eu/wp-content/uploads/2018/06/AIOTI-HLA-R4.0.7.1-Final.pdf

[Bandyopadhyay 11] Debasis Bandyopadhyay, Jaydip Sen: Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Pers Commun. 58, 49-69 (2011)

[Bendel 16] Bendel, Oliver: 300 Keywords Informationsethik: Grundwissen aus Computer- Netz- und Neue-Medien-Ethik sowie Maschinenethik (2016)

[DIN SPEC 91345] DIN SPEC 91345, Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), April 2016

[ETSI 2016] ETSI EG 203 251 V1.1.1, Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies (2016) http://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_60/eg_203251v010101p.pdf

[Heidrich 16] Mike Heidrich und Jijun Luo, Industrial Internet of Things, Referenzarchitektur für die Kommunikation, https://www.esk.fraunhofer.de/content/dam/esk/dokumente/Whitepaper_IoT_dt_April16.pdf

[Humble 10] Jez Humble, David Farley: Continuous Delivery. Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley, Upper Saddle River (2010)

[IEC 61508] Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements, (2010) https://www.vde-verlag.de/iec-normen/preview-pdf/info_iec61508-1%7Bed2.0%7Db.pdf

[IEEE 1028] IEEE Standard 1028-2008 - IEEE Standard for Software Reviews and Audits (2008) https://standards.ieee.org/findstds/standard/1028-2008.htmlhttps://standards.ieee.org/findstds/standard/1028-2008.html

[ISO/IEC/IEEE 24765] ISO/IEC/IEEE 24765:2010: Systems and software engineering – Vocabulary

[ISO/IEC 25010] ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models (2011) https://www.iso.org/standard/35733.html

[ISO 27034] ISO/IEC 27034:2011, Information technology — Security techniques — Application security (2011) http://www.iso27001security.com/html/27034.html

[ISO/IEC CD 30141] ISO/IEC CD 30141, Internet of Things Reference Architecture (IoT RA) (2016) https://www.w3.org/WoT/IG/wiki/images/9/9a/10N0536_CD_text_of_ISO_IEC_30141.pdf

[ISO/IEC/IEEE 42010] ISO/IEC/IEEE 42010:2011, Systems and software engineering -- Architecture description (2011) https://www.iso.org/standard/50508.html

[ISO 9126] ISO/IEC 9126-1:2001, Software engineering -- Product quality -- Part 1: Quality model

https://www.iso.org/standard/22749.html

[ISTQB 11] International Software Testing Qualifications Board: ‘Certified Tester Foundation Level Syllabus’ deutschsprachige Ausgabe V 2011 1.0.1 (2011)

[ISTQB 16] ISTQB® Certified Tester Advanced Level Syllabus “Test Automation Engineer” (2016) http://www.Istqb.Org/Certification-Path-Root/Test-Automation-Engineer.html

[ISTQB 17] ISTQB®/GTB Standardglossar der Testbegriffe Deutsch / Englisch Version 3.11 (2017)

[Jäkel 17] Frank-Walter Jäkel et al, R2.2: Testarchitekturen (2017) http://www.iot-t.de/wp-content/uploads/sites/11/2017/07/IoT-T_R2.2.pdf

[Kuhlen 04] Kuhlen, Rainer. Informationsethik: Umgang mit Wissen und Informationen in elektronischen Räumen (2004)

[oneM2M 16] oneM2M Technical Report, Use Cases Collection (2016) http://www.onem2m.org/images/files/deliverables/Release2/TR-0001-Use_Cases_Collection-V2.4.1.pdf

[oneM2M 18] oneM2M Technical Specification, Functional Architecture (2018), http://www.onem2m.org/component/rsfiles/download-file/files?path=Release_2_Draft_TS%255CTS_0001-Functional_Architecture-V2_19_0.docx&Itemid=238

[oneM2M drafts] oneM2M published drafts, http://www.onem2m.org/technical/published-drafts

[RAMI 4.0 15] Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0), Martin Hankel und Bosch Rexroth (2015)

[Riedel 16] Oliver Riedel et al, Modellbasierte modulare Shopfloor IT - Integration in die Werkzeuge der Digitalen Fabrik (2016) http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-3162488.pdf

[Rötzer 16] Rötzer, Florian (Hrsg.): Programmierte Ethik: Brauchen Roboter Regeln oder Moral? (2016)

[Schieferdecker 16] I. Schieferdecker et al, Das Ende der Unsicherheit – Quality Engineering für IoT (2016) https://www.sigs-datacom.de/uploads/tx_dmjournals/Schieferdecker_Metzger_Rennoch_IOT_16.pdf

[VDI/ZVEI 15] VDI/ZVEI Statusreport Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) (2015) https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf

[Wardaschka 17] André Wardaschka et al, R4.1: Stand IoT Labore / Auswahl viel versprechender Protokolle (2017) http://www.iot-t.de/wp-content/uploads/sites/11/2017/07/IoT-T_R4.1.pdf

[Weyrich 16] Michael Weyrich et al, Referenzarchitekturen für das IoT: Überblick zum Stand der Technik und wesentliche Trends, https://vector.com/portal/medien/vector_consulting/publications/IoT_Architektur_2016.pdf

Anhang C Lernziel / Kognitive Ebenen des Lernens

Auszug aus **[ISTQB 11]:**

Die folgende Taxonomie für Lernziele bildet die Grundlage des Lehrplans. Jeder Inhalt wird entsprechend den zugeordneten Lernzielen geprüft.

Taxonomiestufe 1: **Kennen (K1)**

Der Lernende ruft im Gedächtnis gespeicherte Informationen (z.B. Begriffe, isolierte Fakten, Abfolgen, Prinzipien, Mittel und Wege) ab. Typische beobachtbare Leistungen sind erkennen, nennen, bezeichnen.

Schlüsselworte: sich erinnern, erkennen, wiedergeben, kennen

Taxonomiestufe 2: **Verstehen (K2)**

Der Lernende begründet oder erläutert Aussagen zum Thema. Typische beobachtbare Leistungen sind beschreiben, zusammenfassen, vergleichen, klassifizieren, begründen, erklären, Beispiele für Testkonzepte nennen.

Schlüsselworte: zusammenfassen, verallgemeinern, abstrahieren, klassifizieren, vergleichen, auf etwas übertragen, etwas gegenüberstellen, erläutern, interpretieren, übersetzen, darstellen, rückschließen, folgern, kategorisieren, Modelle konstruieren, erklären, Beispiele geben, begründen, verstehen

Taxonomiestufe 3: **Anwenden (K3)**

Der Lernende überträgt erworbenes Wissen auf gegebene neue Situationen oder wendet sie zur Problemlösung an. Typische beobachtbare Leistungen sind ausführen, anwenden, beurteilen, ermitteln, entwerfen, analysieren.

Schlüsselworte: anwenden, einsetzen, ausführen, nutzen, Verfahren verstehen, Verfahren anwenden