VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

Testen vernetzter Systeme für Industrie 4.0

Zusammenfassung

Die Digitalisierung im Bereich der industriellen Automatisierungstechnik und die damit einhergehenden Veränderungen bei der Produktentwicklung und im Betrieb haben weitreichende Folgen. Ziel dieses Statusreports ist es, die Auswirkungen der Digitalisierung auf das Testen von Industrie-4.0-Komponenten und -Systemen zu beschreiben sowie Interessierte aus Industrie und Forschung für neue Herausforderungen zu sensibilisieren, wenn sie durch die Komplexität ihrer Produkte, Produktionsanlagen sowie deren IT-Umgebungen an die Grenzen ihrer gegenwärtigen Testverfahren kommen. Außerdem erhalten sind ein Überblick über bestehende Standards und Normen im Bereich der Informationstechnologie.

Mittels Use Cases werden die Veränderungen anschaulich dargestellt und die Ergebnisse im Anschluss tabellarisch zusammengefasst. Abschließend folgt eine detailliertere Erläuterung der einzelnen Bereiche und Komponenten, die sich durch Industrie 4.0 wandeln.

Als Ergebnis des Statusreports ergibt sich:

- Zur Beherrschung zunehmender Komplexität ist eine Unterteilung der betrachteten Systeme in Teilsysteme notwendig. Diese Unterteilung erfordert den Einsatz wohldefinierter Schnittstellen.
- Proprietäre Fremdlösungen lassen sich vom Nutzer vielfach nicht vollständig testen. Die zertifizierte Durchführung standardisierter Tests kann hier nötiges Vertrauen schaffen.
- Einige bewährte Standards aus anderen Domänen könnten nach Anpassungen im Bereich der Automatisierungstechnik übernommen werden.
- Die Qualitätsanforderungen, die wir heute für Standardsoftware kennen, werden zukünftig auch für in Losgröße 1 produzierte Software vorausgesetzt.
- Die Herausforderungen liegen in der Modellbildung für den Testaufbau und die Simulationsumgebung.
- Zur effizienten und reproduzierbaren Durchführung von Tests in komplexen und sich ändernden Umgebungen werden modellbasierte Verfahren bei der Testautomatisierung zunehmend an Relevanz gewinnen.
- Zur Abschätzung einer ausreichenden Absicherung gewinnen Metriken an Relevanz.

Düsseldorf im April 2018

Prof. Dr.-Ing. Michael Weyrich
Vorsitzender des Fachausschusses
„Testen vernetzter Systeme in Industrie 4.0“

Autoren

An der Erstellung dieses Statusreports haben folgende Mitglieder des Fachausschusses 7.25 „Testen vernetzter Systeme für die Industrie 4.0“ mitgearbeitet:

Prof. Christian Facchi; Technische Hochschule Ingolstadt, Ingolstadt

Sebastian Heidepriem; SICK AG, Waldkirch

Dr. Jürgen Jähnert; bwcon GmbH, Stuttgart

Tobias Jung; Universität Stuttgart, Stuttgart

Dr. Christian Köllner; Vector Informatik GmbH, Stuttgart

Alexander Kraas; T-Systems International GmbH, Nürnberg

Dr. Jan Krause; ifak e. V. Magdeburg, Magdeburg

Dr. Kai Krüning; BASF SE, Ludwigshafen

Alexander Kugler; RWTH Aachen, Aachen

Benjamin Maschler; Universität Stuttgart, Stuttgart

Christian Schleicher, Festo AG & Co KG, Esslingen

Daniel Siegrist; SEW Eurodrive GmbH & Co KG, Bruchsal

Hendrik Simon; RWTH Aachen, Aachen

Dr. Christoph Störmer; ETAS GmbH, Stuttgart

David Thönnessen; RWTH Aachen, Aachen

Erik Wassermann; Helmut-Schmidt-Universität Hamburg, Hamburg

Prof. Michael Weyrich; Universität Stuttgart, Stuttgart

Dr. Thomas Wimmer; Wittenstein SE, Igersheim

Andreas Zeller; Universität Stuttgart, Stuttgart

Inhalt

1 Einleitung

Dass die Komplexität vernetzter Komponenten und deren Entwicklung zunehmen, ist hinlänglich bekannt. So zieht sich das Motiv „Komplexitätssteigerung“ wie ein roter Faden durch die verschiedenen Abschnitte des Statusreports. Deshalb wird auf das Motiv und dessen Ausprägung im Anschluss eingegangen. Wir haben folgendes Verständnis von Komplexität [1]:

Die Komplexität eines Systems ergibt sich aus der Gesamtheit ihrer voneinander abhängigen Merkmale und Elemente, die ein ganzheitliches Beziehungsgefilde bilden. Im Kontext von Industrie 4.0 ist eine Steigerung der Komplexität von Systemen unter anderem durch diese Faktoren erkennbar:

- Heterogenität der Komponenten
- Wechselwirkung der Komponenten innerhalb des Systems
- systemübergreifende Kommunikation
- Änderung von Systemstrukturen durch Ad-hoc-Vernetzung und Rekonfiguration
- Skalierung des Systems
- gemeinsame Betrachtung von Software, Hardware und Kommunikation

Im Vergleich zu herkömmlichen, nicht vernetzten Systemen erfüllen Industrie-4.0-Systeme viele oder sogar alle Merkmale dieser Faktoren. Die wachsenden Anforderungen an die Systeme beeinflussen diese. Zukünftige Produktionsanlagen sollen beispielsweise anhand von Ad-hoc-Vernetzung und Rekonfiguration wesentlich flexibler auf Produktänderungen reagieren. Außerdem wird die Nachverfolgbarkeit eines Produkts über den gesamten Lebenszyklus erstrebt. Industrie-4.0-Systeme sind intern und extern vernetzt: Die Komponenten innerhalb des Systems interagieren, es findet systemübergreifende, domänenübergeifende und Machine-to-Machine-Kommunikation (M2M-Kommunikation) statt. Neben dem Testen der Systemfunktionen wird es folglich notwendig, den Einfluss des durch austauschbare Komponenten veränderlichen Systems auf ebensolche kooperierende Systeme und umgekehrt zu berücksichtigen. Damit erhöht sich der Testaufwand.

Es existieren bereits Ansätze, um die Komplexität von Industrie-4.0-Systemen bzw. Internet-of-Things-Systemen (IoT-Systemen) beherrschbar zu machen. Dafür wurden in den letzten Jahren mehrere Referenzarchitekturen entwickelt. Zu nennen sind hier die Industrial Internet Reference Architecture (IIRA) des Industrial Internet Consortium (IIC) und das Reference Architecture Model Industry 4.0 (RAMI 4.0) des Zentralverbands Elektrotechnik- und Elektronikindustrie (ZVEI). Hierbei schließt RAMI 4.0 die Verwaltungsschale mit ein und ermöglicht eine Kapselung vieler Aspekte. Trotzdem kann durch diese Referenzarchitekturen nur die Komplexität verringert werden; sie beschreiben keine systematischen Testprozesse.

Die steigende Komplexität der Systeme resultiert in einer gesteigerten Komplexität der Testfälle. Des Weiteren werden mehr Testfälle benötigt, um eine ähnliche Testabdeckung bei zusätzlichen zu berücksichtigenden Aspekten zu erreichen. So werden neue Standards, Normen und Testverfahren benötigt, um Änderungen an Industrie-4.0-Systemen handhaben zu können.

Testszenarien in solchen Systemen sind der Systemtest sowie der Komponententest aus der Innen- (White-Box-Perspektive) sowie der Außensicht (Black-Box-Perspektive). Um den Herausforderungen zu begegnen, können bestehende Testprozesse angepasst bzw. neue Testprozesse entwickelt werden. Gleiches gilt für Testsprachen, um eine Durchgängigkeit bei der Entwicklung von Industrie-4.0-Systemen zu erreichen. Außerdem müssen Metriken zur Messung der Komplexität und der Testabdeckung an die neuen Gegebenheiten angepasst werden, um sowohl den Testaufwand als auch den Erfolg quantifizierbar zu machen.

Der folgende Abschnitt beschreibt anhand von Use Cases die Änderungen durch Industrie 4.0 in verschiedenen Bereichen. Nach einer Darstellung der Ergebnisse folgt die Erläuterung der wichtigsten Begrifflichkeiten.

2 Use Cases

Ausgehend von der Komplexitätssteigerung in Industie-4.0-Systemen, beschreibt dieses Kapitel die zu erwartenden Änderungen in konkreten Bereichen der Automatisierungstechnik. Diese werden anschaulich von Experten der jeweiligen Domäne in Anwendungsfällen dargestellt.

2.1 Test von Prozessleitsystemen

2.1.1 Heutiges Vorgehen

Prozessleitsysteme (PLS) werden beim Test notwendigerweise in Teilsysteme untergliedert. Zu der Absicherung der Teilsysteme gehören beispielsweise Interoperabilitätstests für die Integration intelligenter Feldgeräte in das Leitsystem oder Tests des Batch-Systems. Weiterhin wird das Gesamtsystem während des Factory Acceptance Test (FAT) beim Hersteller und beim Site Acceptance Test (SAT) in der Anlage geprüft. Für alle Tests existieren vorgesehene Zeitpunkte und Testvorschriften (beispielsweise DIN EN 62381).

2.1.2 Aktuelle Herausforderungen

Die Herausforderungen ergeben sich heute aus den unterschiedlichen Einsatzszenarien und dem Einsatz von Prozessleitsystemen verschiedener Hersteller bzw. unterschiedlicher Prozessleitsysteme des gleichen Herstellers. Der Komplexität wird zunächst durch genaue Testvorgaben für die einzelnen (Teil-) Systeme der Hersteller begegnet. Weiterhin sind jedoch ein hohes Know-how und eine genaue Kenntnis des zu testenden Systems bei den Testern notwendig. Für Spezialfälle wird auf das Wissen der zuständigen Spezialisten zugegriffen.

2.1.3 Zukünftiger Use Case

Mit der zunehmenden vertikalen Systemintegration, z. B. als Voraussetzung für entstehende Wertschöpfungsnetzwerke, steigt der Kommunikationsbedarf über die einzelnen Ebenen der Automatisierungspyramide. Die Abhängigkeiten zwischen Prozessleitsystem (PLS), Produktionsleitsystem (Manufacturing Execution System, MES) und Produktionsplanung (Enterprise Resource Planning, ERP) werden zunehmend größer. Ein ausschließlicher Test des PLS ist folglich nicht mehr ausreichend. Es ist zudem notwendig, die Kommunikation und das Verhalten des PLS auf Ereignisse in dieser Kommunikation zu testen. Typische Fragestellungen sind hier in Zukunft unter anderem:

- Welche Daten werden ausgetauscht?
- Wer erwartet welche Reaktionen?
- Was passiert bei Kommunikationsausfall oder fehlerhafter Kommunikation?
- Wie wird die Kommunikation abgesichert (Security)?

Eine weitere Schwierigkeit ergibt sich aus unterschiedlichen organisatorischen Zuständigkeiten: Typischerweise sind ERP und PLS in verschiedenen Abteilungen verortet, (z. B. IT bzw. Automatisierungstechnik).

Eine Lösungsstrategie bietet die Standardisierung von Testprozessen und Testfallbeschreibungen. Dies betrifft sowohl die einzelnen Systeme (PLS, MES, ERP) und deren Schnittstellen als auch die Inhalte der Kommunikation. Es ist erforderlich, jede Systemschnittstelle auf typische Fehlerfälle von außen zu testen. Dafür müssen die Abhängigkeiten zwischen den Systemen eindeutig beschrieben sein. Die Systeme selbst könnten dann weiter in bewährter Weise getestet werden.

2.2 Feldgeräte (Komponenten testen)

2.2.1 Heutiges Vorgehen

Jede Produktentwicklung wird nach einem definierten Prozess zweifach spezifiziert. Zum einen mit dem Lastenheft, das die Frage „Was wird benötigt“? beantwortet, und zum anderen mit dem Pflichtenheft, das über die Art der Realisierung aufklärt. Gemäß V-Modell werden mehr oder weniger gleichzeitig die Tests für die Abnahmeprüfung (Freigabe) definiert. Hierfür ist eine eigene, unabhängige Abteilung zuständig. Alle in der Spezifikation beschriebenen Eigenschaften werden **gegen die Spezifikation** getestet.

Des Weiteren werden Integrationstests durchgeführt. Dazu wird das Produkt im Zusammenspiel mit beteiligten Komponenten (z. B. SPS) zusammen getestet. Das Ziel dabei ist das Entdecken und Umgehen von Fehlern oder Schwächen in der Implementierung bei Komponenten Dritter.

2.2.2 Aktuelle Herausforderungen

Annähernd jedes kommunikationsfähige Gerät weist heute eine Komplexität auf, die nicht mehr, auch nicht annähernd, hinsichtlich aller denkbaren Ein- und Ausgangsdaten geprüft werden kann. Der Zustandsgraph des Gesamtsystems wird somit annähernd unendlich komplex. Beim funktionalen Testen wird auf die zugehörige Spezifikation fokussiert, das heißt es wird versucht, alle relevanten Anforderungen aus der Spezifikation gründlich zu testen. Das erfordert ein systematisches Vorgehen, was die Testerstellung betrifft, und eine hohe Automatisierung, was die Testdurchführung betrifft.

Mit Stresstests wird versucht, komplexe Überlastsituationen zu simulieren und durch geeignete Gegenmaßnahmen zu stabilisieren. Der Umgang mit Fehlern auf der Gegenseite bei Integrationstests ist herausfordernd: Nicht jedes Gerät anderer Hersteller ist zum Testen verfügbar und ein Workaround bei Kompatibilitätsproblemen nicht immer erwünscht. Im Gegensatz zum Vorgehen der finalen Freigabeprüfung geht heute die Produktentwicklung großer Produktfamilien zu einer stetigen Verbesserung der Produktfamilie (Continuous Integration) über. Die Fehlerkategorie „Erwartungshaltung der Anwenderrobustheit“ gewinnt immer mehr an Bedeutung. Bei falscher Anwendung stellt der Nutzer meist das Produkt und dessen Robustheit oder die Intuitivität der Bedienung infrage.

2.2.3 Zukünftiger Use Case

Durch schnelle Eingriffe in der Software, wie das Hinzufügen von Funktionen oder Wiederverwenden von Bibliotheken, ist das Gesamtsystem immer schwieriger verifizierbar. Die resultierende Komplexität wird zukünftig noch größere Probleme darstellen, als dies schon heute der Fall ist. Testen sollte zunächst vollständig automatisiert werden, um bei Systemänderungen die Tests sofort wiederholen zu können. Darauf aufbauend könnte mit Methoden der Testgenerierung (z. B. modellbasiertes Testen) die Testerstellung systematisiert und automatisiert werden (siehe Bild 1).

Außerdem ist die Reduktion der Komponentenkomplexität nötig. Eingangsdaten für Komponenten müssen auf ein Minimum beschränkt werden. Dadurch sollen über das Abschalten nicht genutzter Funktionen die mögliche Angriffsfläche (attack surfaces) reduziert werden (IEC 62443). Dabei sollen Penetrationstests so durchgeführt werden, dass durch dabei potenziell entstehende „Backdoors“ keine zusätzlichen Angriffsvektoren entstehen. Die statische Code-Analyse oder ein menschliches Review sind heute schon sehr effektive Methoden, um solche Schwachstellen zu finden. Die verbleibenden Schnittstellen müssen wieder vereinfacht werden, um vollständige Tests zu ermöglichen. Dafür ist es notwendig, die Tests auf Module oder Funktionen (Units) herunterzubrechen.

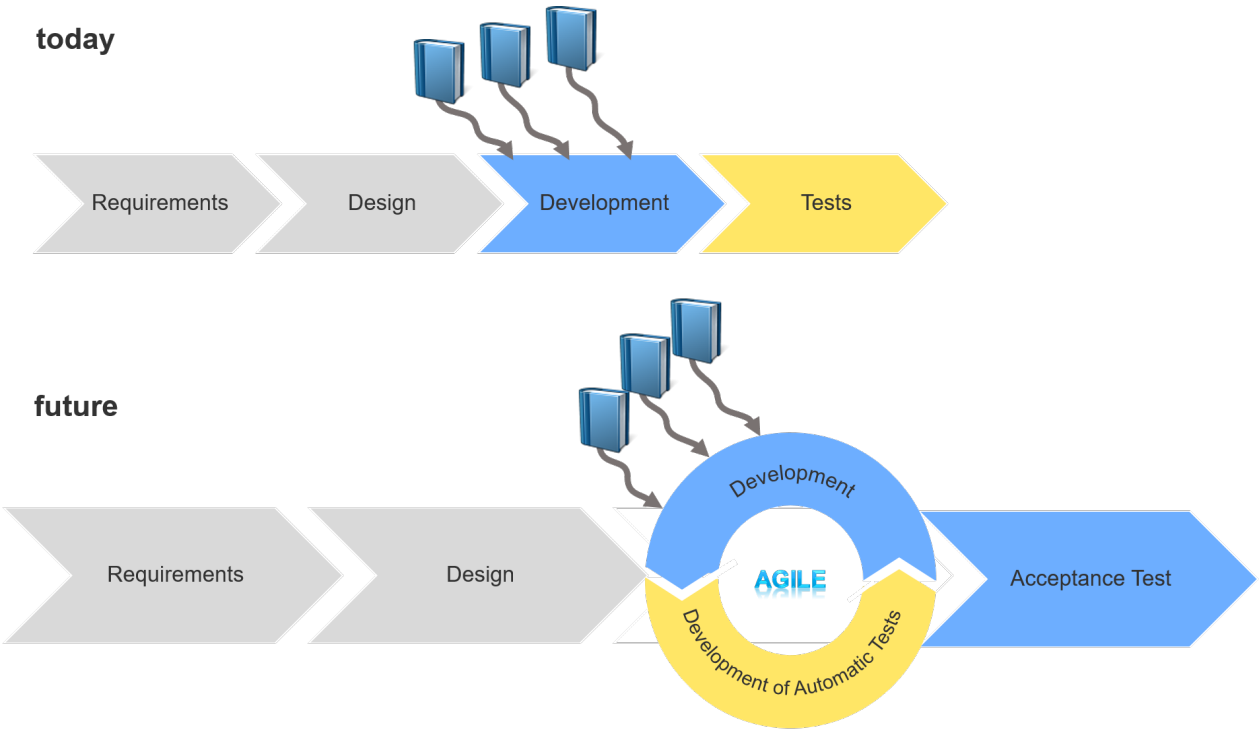

Bild 1. Änderung der Testzyklen im Entwicklungszyklus einer Komponente

Da von der Anwenderseite nicht verlangt werden, kann eine Bibliothek im Detail zu testen, ist eine Art Gewährleistung für die Wiederverwendung von Softwarefunktionen nötig. Diese bestätigt die erfolgreiche Durchführung eines möglichst umfassenden Tests. Die größte Herausforderung besteht künftig in der Frage, ob eine Funktion verständlich strukturiert ist und spezifiziert wird. Hierbei sind keine absoluten Aussagen möglich, da ähnlich zu den Penetrationstests Abhängigkeiten zur testenden Person unvermeidbar sind.

2.3 Softwareproduktion für Standardapplikationen

2.3.1 Heutiges Vorgehen

Die Produktion von Standardsoftware für Standardapplikationen in der Anlagen- und Maschinenautomatisierung erfolgt heute nach einem klassischen, sequenziellen Entwicklungsmodell. Dadurch bedingt fallen die Hauptaktivitäten des Testens am Ende der linearen Anordnung der Entwicklungstätigkeiten an. Da der erstellte Code (Programmiersprachen gemäß DIN EN 61131-3) erst in Verbindung mit einer Steuerung ausführbar ist, steigt zusätzlich die Bedeutung des abschließenden Systemtests. Mit Methoden der virtuellen Inbetriebnahme (VIBN) können auch diese Systemtests weitestgehend im Labor und/oder am Arbeitsplatz des Testingenieurs durchgeführt werden. Die verwendete Teststrategie sieht eine hauptsächlich anforderungsbasierte Testfall-Auswahl für den dynamischen Test vor. Die Testdokumentation basiert auf einem unternehmensspezifischen „Tailoring“ der IEEE 829.

2.3.2 Aktuelle Herausforderungen

Die Fokussierung auf die oberen Teststufen in Verbindung mit dem gewählten Entwicklungsmodell führt zu einer verhältnismäßig langen Fehlerverweildauer (Zeitraum zwischen Fehlhandlung und Aufdeckung der Fehlerwirkung). Abhilfe schafft hier die Einführung eines iterativen Entwicklungsmodells mit der Verkürzung der Entwicklungszyklen in Verbindung mit einer Parallelisierung der Testzyklen.

Zusätzlich wird das Konzept der Phaseneingrenzung eingeführt. Durch das Anwenden von statischen Testmethoden und der Einführung zusätzlicher Teststufen sollen Abweichungen in der Phase des Entwicklungsprozesses aufgedeckt und behoben werden, in der sie entstehen. Das Monitoring von Metriken zur Bewertung des Testprozesses ermöglicht die stetige Verbesserung der Test- und Entwicklungsprozesse.

2.3.3 Zukünftiger Use Case

Bedingt durch die zunehmende Vernetzung zeichnet sich ab, dass sowohl die Menge und Komplexität an benötigter Software steigt als auch deren Bedeutung für die Qualität des Gesamtprodukts stetig zunimmt (Systeme werden zu Multisystemen). Gleichzeitig entwickelt sich die Softwareentwicklung von einem eher wissenschaftlichen Prozess hin zu einem Produktionsprozess (Industrialisierung der Softwareentwicklung). Dabei ist die Produktionsphilosophie, die Industrie 4.0 mit sich bringt, inbegriffen (vgl. [2]). Die Qualitätsanforderungen, die wir heute für Standardsoftware kennen, werden also zukünftig auch für in Losgröße 1 produzierte Software vorausgesetzt. Es ist zu erwarten, dass sich das Testen, bei dem bisher der Fokus auf dem (Standard-)Produkt lag, hin zu einem standardisierten Prozess mit hoher Automatisierung der Testerstellung und Testdurchführung wandelt. Die Zunahme von Testprozesswissen verschiebt den Fokus hin zu einer ganzheitlichen Qualitätssicherungsstrategie, die den gesamten Produktlebenszyklus abdeckt (siehe Bild 2). Hierbei tritt neben dem Testen die Fehlerverhinderung in den Vordergrund. Die Grundidee **Die Qualität der Software wird wesentlich durch die Qualität des Erstellungsprozesses beeinflusst** schlägt sich bereits heute in verschiedenen (Test-)Prozessreferenzmodellen nieder.

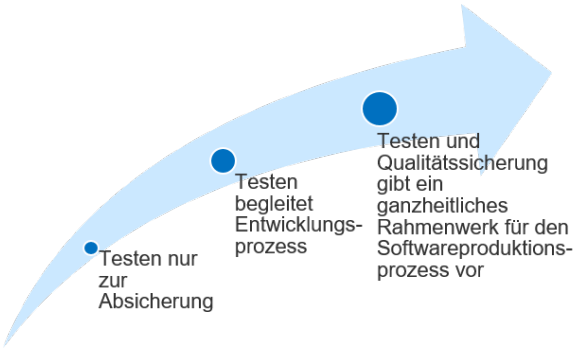

Bild 2. Entwicklungsstufen beim Softwaretest

2.4 Fahrzeugfunktionen im IoT-Kontext

2.4.1 Heutiges Vorgehen

Gegenwärtig sind Testmethoden in der Automobilindustrie primär auf elektronische Steuergeräte (Electronic Control Units – ECU) ausgerichtet: Das klassische „System under Test“ (SUT) ist eine ECU, die in einer weitestgehend statischen Netzwerktopologie eingebettet ist. Die Systemgrenzen sind dabei die Busframes und die I/O-Interfaces. Klassische Tests operieren auf Bussignalen, Diagnoseprotokollen und teilweise Busframes.

2.4.2 Aktuelle Herausforderungen

ECUs sind seit einiger Zeit keine einfachen Regelblöcke mehr. Sie weisen komplexe Algorithmen und Datenmodelle, Kommunikationsschnittstellen aus der PC-Welt, asynchrone Programmierparadigmen, eine „Service oriented Architecture oder Adaptive AUTOSAR“ auf. Der Simulations- und Konfigurationsaufwand steigt damit erheblich und eine direkte Zuordnung von konkreten Funktionen zu einzelnen ECUs wird zunehmend schwierig. In der Folge ist es notwendig, verstärkt auf Anwendungsebene zu testen, um Funktionen in den Vordergrund zu rücken und Kommunikationstechnologien in einem möglichen und sinnvollen Rahmen zu abstrahieren.

2.4.3 Zukünftiger Use Case

Das Testen von Fahrzeugfunktionen wird zukünftig virtuell und hoch automatisiert ablaufen und insbesondere auf deren Interaktion mit dynamischen Ad-hoc-Netzwerken (IoT) abzielen.

Die Herausforderungen liegen dabei einerseits in der Modellbildung für den Testaufbau und die Simulationsumgebung, z. B. in der Identifikation geeigneter Domänenmodelle, und andererseits in der Definition geeigneter Schnittstellen für Testfälle. Außerdem bestehen Schwierigkeiten in der Wahl der geeigneten Abstraktionsebene, z. B. in Bezug auf die Genauigkeit der Simulation von Verhalten, Zeit oder Kommunikation, sowie in der Berücksichtigung von unzähligen Teilnehmerkonstellationen bei gleichzeitig begrenzten Ressourcen zur Durchführung von Tests.

Erste Schritte in diese Richtung lassen vermuten, dass Test- und Modellierungssprachen zukünftig verstärkt optimiert und erweitert werden müssen, insbesondere hinsichtlich der Abbildung dynamischer Vorgänge. Außerdem besteht die Notwendigkeit, die Vorteile moderner Programmierparadigmen auch im Kontext des Testens von Fahrzeugfunktionen schneller und konsequenter nutzbar zu machen.

2.5 Security

2.5.1 Heutiges Vorgehen

In der Fertigungs- und Prozessindustrie kommt es zunehmend zur Vernetzung von Automatisierungseinrichtungen zum Zweck der Steuerung und Überwachung der Produktion. Dabei stellt die Netzwerkschnittstelle eine wichtige Angriffsfläche für Bedrohungen wie Computerviren und andere Malware dar. Durch verschiedene Faktoren kommt es zu einer großen Vielfalt an gleichzeitig im Einsatz befindlichen Betriebssystemversionen und Anwendungen. Als Teil des Asset Management wäre ein Soll-Ist-Vergleich durch Netzwerkscans hilfreich. In Kombination mit einem Schwachstellenscanner könnten außerdem bekannte Schwachstellen von Betriebssystemen und Anwendungen automatisiert identifiziert werden.

2.5.2 Herausforderungen

Durch die lange Lebensdauer der netzwerkfähigen Systeme und die geringere Patch-Frequenz kann nicht ohne Weiteres davon ausgegangen werden, dass jedes System in einem industriellen Netzwerk einen Netzwerkscan oder Schwachstellenscan ohne Fehlfunktion übersteht. Dies erschwert das Testen mit Security Tools, deren Einsatz in anderen Bereichen für den zuverlässigen Betrieb notwendig ist. Durch die drohenden Verluste bei ungeplanten Stillständen von Anlagen und dem Risiko von nicht oder schwer behebbaren Fehlern durch das Testen auf zuvor nicht bekannte Sicherheitsprobleme während der Lebensdauer ist das Testen im Betrieb nur in seltenen Fällen möglich.

2.5.3 Zukünftiger Use Case

Aufgrund der fortschreitenden Vernetzung steigt das Risiko für Security-Schwachstellen in der Betriebsphase, die über das Netzwerk ausnutzbar wären. Diese können aber nicht durch Tests erkannt werden, da viele der verbauten Komponenten nicht für einen vernetzten Einsatz ausgelegt wurden. Dadurch liegt die Gefahr von Stillständen oder Fehlfunktionen beim Betreiber. Es können folgende Trends beobachtet werden:

- Bei der Beschaffung von Neuanlagen spielen Security-Anforderungen zunehmend eine Rolle, die von Abnahmetests begleitet sein können.
- Es entstehen Teststandards für den Test von industriellen Netzwerkkomponenten, z. B. vom Bundesamt für Sicherheit in der Informationstechnik.
- Eine stärkere Segmentierung von Netzen ermöglicht es, testbare Komponenten in kleineren, weniger empfindlichen Subnetzen zu betreiben, z. B. durch Virtual Local Area Networks(VLANs).

2.5.4 Resümee der Use Cases

In den beschriebenen Anwendungsfällen aus verschiedenen Domänen finden sich bestimmte Aspekte wieder. Tabelle 1 veranschaulicht die Abhängigkeiten.

Anhand der oben genannten Use Cases hat sich gezeigt, dass die dargestellten Begriffe besonders relevant sind. Daher werden sie im folgenden Abschnitt explizit betrachtet und erläutert.

Tabelle 1. Mapping der Begriffserläuterungen mit den Use Cases

	Agile Methoden	Schnittstellen	Standards zur Beschreibung von Testfällen	Testautomatisierung	Modellbasiertes Testen	Metriken
Test von Prozessleitsystemen		x	x	x		
Feldgeräte (Komponenten testen)	x	x	x	x		
Softwareproduktion für Standardapplikationen	x		x	x		x
Fahrzeugfunktionen im IoT-Kontext		x	x	x	x	
Security		x	x			

3 Begriffe und deren Bedeutung

3.1 Agile Methoden

3.1.1 Einführung

Zur Strukturierung von Entwicklungsprojekten sind zahlreiche Vorgehensmodelle und -methoden entstanden. Da agile Methoden bei der Produktentwicklung zunehmend an Bedeutung gewinnen, wird auf diese explizit eingegangen. Dies kann unter anderem auf die höhere Integration des Kunden in den Entwicklungsprozess zurückgeführt werden. Es beinhaltet ausdrücklich keine Wertung etablierter Entwicklungsmethoden, die weiterhin ihre Berechtigung besitzen.

Der Begriff „Agilität“ bezeichnet vor allem Geschwindigkeit und Flexibilität in der Umsetzung. Dies sind wichtige Attribute in der schnelllebigen IT-Welt und somit finden agile Methoden wachsenden Einsatz in Unternehmen. Eines ihrer Hauptziele liegt darin, die steigende Komplexität in der Entwicklung kontrollierbar zu machen. Diese wird laut der IBM-Global-CEO-Studie [3] zukünftig die größte Herausforderung für Unternehmen sein. Darüber hinaus haben kürzer werdende Lebenszyklen, ein rasanter Technologiewandel, die zunehmende Wettbewerbsintensität und die Globalisierung der Märkte eine große Auswirkung auf die Produktentwicklung. Insbesondere Innnovationsprojekte, wie Industrie 4.0, sind charakterisiert durch eine Vielzahl unvollständiger Informationen, Unsicherheit und interdisziplinärer Zusammenarbeit. Deshalb ist eine starre Struktur des Entwicklungsprozesses oftmals nicht zielführend.

Aus den oben genannten Punkten sind die agilen Methoden bei heutigen Produkten von großem Interesse. Bei diesen Prozessen wird ein besonderes Augenmerk auf kontinuierliche Interaktion mit dem Kunden und regelmäßige Abstimmung in den Projektteams gelegt und das Testen findet entwicklungsbegleitend statt. Dieses „Agile Testing“ ist bei komplexen Projekten essenziell. Hierbei steht vor allem das automatisierte Testen im Fokus, auf das im Weiteren näher eingegangen wird.

3.1.2 Bedeutung

Agile Prozesse, z. B: SCRUM, setzen automatisierte Testmethoden voraus. Ansonsten kann die geforderte Qualität eines Softwareprodukts nicht zugesichert werden [4]. Darüber hinaus ist eine Kerneigenschaft des Testens die Sicherstellung der Reproduzierbarkeit der Tests. Dies ist insbesondere bei verteilten Systemen, wie sie bei Industrie 4.0 vorkommen, ohne Automatisierung kaum sicherzustellen.

Dabei ist auch das automatisierte Testen von verteilten Systemen sehr herausfordernd. Dies ist insbesondere wegen der Vielzahl der beteiligten Komponenten der Fall, die für sich selbst agieren. Der Aufwand steigt weiter, wenn die Komponenten von unterschiedlichen Herstellern entwickelt werden und dezentral organisiert sind. Dann ist eine Synchronisation des Testprozesses wegen der Vielzahl der beteiligten, weitgehend autarken Teilsysteme sehr komplex.

3.1.3 Erläuterung

Die Definition einheitlicher Testprozesse und -inhalte ist zur Erhöhung der Modularität sinnvoll. Dies folgt einerseits aus der Notwendigkeit einer Testautomatisierung und andererseits aus der Vielzahl der verwendeten Komponenten mit gegebenenfalls unterschiedlichen Herstellern. Damit eine einheitliche Basis gewährleistet werden kann, ist die Kooperation der Hardware- und Softwarehersteller sowie der Systemintegratoren notwendig.

3.2 Schnittstellen

3.2.1 Einführung

Die allgemeine Zunahme von Komplexität in Industrie-4.0-Umgebungen erfordert Gegenmaßnahmen, um den Aufwand für Konzeption, Aufbau und Betrieb solcher Systeme auch zukünftig beherrschbar zu halten. Ein Ansatz zur Komplexitätsreduktion ist die noch stärkere Aufteilung von Systemen in Teilsysteme, die über Schnittstellen (engl. interfaces) interagieren. Zu unterscheiden sind dabei mindestens drei verschiedene Schichten von Schnittstellen:

- **Prozess-Ebene** (Welche Informationen werden zu welchem Zweck übertragen?)
- **Software-Ebene** (Wie sind die übertragenen Informationen strukturiert?)
- **Hardware-Ebene** (Wie werden die Informationen physikalisch übertragen?)

Entscheidend für die Interoperabilität der Teilsysteme wird die exakte Definition und fehlerfreie Implementierung dieser Schnittstellen auf all ihren Schichten sein. In Bezug auf Tests von Schnittstellen gibt es

darüber hinaus mindestens drei verschiedene Dimensionen:

- **White-Box-Perspektive** (Wie wird die Schnittstelle im Innern abgebildet?)
- **Black-Box-Perspektive** (Wie wird die Schnittstelle nach außen abgebildet?)
- **System-Perspektive** (Wie verhält sich die Schnittstelle im Kontext des Gesamtsystems?)

Der Test von Schnittstellen bzw. von aus Teilsystemen zusammengesetzten Gesamtsystemen ist damit eine sehr relevante, mehrdimensionale Herausforderung.

3.2.2 Bedeutung

Am Beispiel der Softwareebene werden im Folgenden die Herausforderungen in Bezug auf Schnittstellen im Industrie-4.0-Kontext herausgearbeitet:

Die Annahme, dass Software grundsätzlich fehlerbehaftet sei, ist bereits heute weit verbreitet. Obschon in ihrer Absolutheit inkorrekt, beleuchtet sie trotzdem einen wichtigen Zusammenhang: Die Fehlerquote hängt maßgeblich von der Komplexität eines Systems ab. Eine Fortschreibung dieses Trends von heutigen, vergleichsweise einfachen Systemen auf zukünftige, hochkomplexe Industrie-4.0-Systeme stimmt wenig optimistisch.

Die Reduktion der Komplexität ist dabei am Markt meist chancenlos. Ein erfolgversprechenderer Ansatz ist hingegen das Herunterbrechen eines komplexen Systems in kleinere, weniger komplexe Teilsysteme. Jedes dieser Teilsysteme ist dadurch mit wenig Aufwand zu realisieren und erst bei der (Re-)Integration dieser Teilsysteme zu einem Gesamtsystem werden Komplexität und damit Aufwand wieder potenziell problematisch. Lösen lässt sich dieses Problem über wohldefinierte Schnittstellen.

Ist die Definition einer Schnittstelle bekannt, können Einzelkomponenten vollständig anhand dieser Schnittstellendefinition getestet werden. Auf diese Weise lässt sich eine grundlegende Interoperabilität sicherstellen und das Risiko bei der Systemintegration deutlich reduzieren. Aber auch für die Themen **Wiederverwendbarkeit** und **Arbeitsteilung** (parallele Entwicklung) ist die vollständige Definition von Schnittstellen von zentraler Bedeutung: Nur die Interface-Definition wird dabei verwendet und die konkrete Implementierung anderer Komponenten braucht nicht beachtet oder bekannt zu sein.

Dieser Ansatz lässt sich auch für das Testen verwenden. Ist die Definition einer Schnittstelle maschinell lesbar definiert (z. B. als Teil der Programmiersprache), so kann das Funktionsverhalten mit allen Eigenschaften bis zum Einfluss auf Ressourcen automatisch getestet werden. Beim Herunterbrechen dieser Vorgehensweise bis auf Funktionsebene kann von Modul-Tests oder Unit-Tests gesprochen werden.

3.2.3 Erläuterung

Neben dem Test aller Schnittstellenschichten sind insbesondere Tests in allen Schnittstellendimensionen wichtig. Dabei gilt, dass jede einzelne Dimension notwendig ist, aber nur alle gemeinsam hinreichend für eine abschließende und umfassende Betrachtung des Gesamtsystems sind. Gleichzeitig ist zu berücksichtigen, dass die einzelnen Dimensionen von unterschiedlichen Akteuren getestet werden müssen: In der Regel kann nur der Hersteller einen White-Box-Test und nur der Kunde einen Systemtest vornehmen.

Es wird daher notwendig sein, Schnittstellentests in ihrer Mehrdimensionalität und -schichtigkeit zu untersuchen und allgemeine (Meta-)Kriterien für das Durchführen derselben zu entwickeln, um durch Transparenz und Vereinheitlichung von Tests eine Basis für den Einsatz von Industrie-4.0-Komponenten zu schaffen. Spezifische Testanweisungen zu erstellen ist hingegen nicht möglich, da sich diese notwendigerweise von Anwendungsfall zu Anwendungsfall unterscheiden werden.

In Bezug auf Systemtests sind außerdem sinnvolle Grenzen zu definieren, da mit zunehmender Vernetzung Tests des vollständigen Systems aufgrund deren Komplexität und Variabilität unmöglich werden.

Darüber hinaus gilt es, technische Konzepte in der Umsetzung von Schnittstellen zu untersuchen, beispielsweise Datendioden und Gateways. Außerdem können allgemeingültige Erweiterungen der gängigen Programmiersprachen notwendig sein, beispielsweise maschinell lesbare Schnittstellendefinitionen für die statische Codeanalyse und für das (zusätzliche) automatische Testen von kleinen Einheiten. Damit soll erreicht werden, die immer feinere Aufteilung komplexer Systeme in Einzelkomponenten beherrschbar zu machen und die Funktionsfähigkeit der Komponenten im Verbund sicherzustellen.

3.3 Standards zur Beschreibung von Testfällen

3.3.1 Einführung

Seit einigen Jahren konnte sich im Bereich des Softwaretests der Standard ISO/IEC/IEEE 829 (bzw. dessen Nachfolger ISO/IEC/IEEE 29119-3) zur Dokumentation von Softwaretests etablieren. Dieser Standard definiert unterschiedliche Arten von Dokumenten zur Erfassung der verschiedenen Testaspekte. Hiervon sind die Testentwurfs-, Testfall- und Testablaufspezifikation für die Beschreibung von Testfällen relevant. Durch die Verwendung dieser Dokumentarten soll eine einheitliche, nachvollziehbare und reproduzierbare Spezifikation und Durchführung gewährleistet werden.

Aufgrund der wachsenden Komplexität, einer zunehmenden Vernetzung und einem hohen Grad der durch Software realisierten Funktionalitäten im Kontext der Industrie 4.0 wird auch für diese Domäne ein zukünftiger Bedarf an standardisierten Testbeschreibungen erwartet. Da der IEEE-829-Standard in erster Linie nur die Strukturierung und den groben Inhalt der unterschiedlichen Dokumentarten vorgibt, stellt sich die Frage, ob die getroffenen Festlegungen für Testfälle im Umfeld von Industrie 4.0 ausreichend und geeignet sind. Für die Beschreibung von Testfällen im Bereich der Telekommunikation wurden bereits vor einigen Jahren Anforderungen identifiziert, die über die in IEEE 829 getroffenen Festlegungen hinausgehen. Daher hat das European Telecommunications Standards Institute (ETSI) weiterführende Standards für den Telekommunikationssektor erarbeitet. Eine Verwendung der bereits etablierten Standards für Testfallbeschreibungen im Kontext Industrie 4.0 erscheint aufgrund bereits vorhandener Werkzeuge und entsprechender Expertise vorteilhaft.

3.3.2 Bedeutung

Durch die Verwendung von standardisierten Testbeschreibungssprachen haben sich im Bereich der Telekommunikation folgenden Vorteile gezeigt, die auch im Kontext Industrie 4.0 als wichtig erachtet werden:

- Unterstützung eines vereinfachten Reviews
- einheitliches Mittel zur Beschreibung von Testfällen
- Sicherstellung der Interoperabilität zwischen Testwerkzeugen

Im Detail bietet sich die Verwendung der folgenden ETSI-Standards für die Beschreibung von Testfällen im Bereich der Industrie 4.0 an:

- Test Description Language (TDL) [6] für eine abstrakte Beschreibung eines Testfalls
- Testing and Test Control Notation Version 3 (TTCN-3) [7] für eine Spezifikation von ausführbaren Testfällen

Die TDL dient der Spezifikation von Testbeschreibungen und der Präsentation von Testausführungsergebnissen. Hierbei wird durch TDL eine methodische Lücke zwischen abstrakten Testanforderungen und ausführbaren Testskripten geschlossen. Eine in TDL verfasste Testbeschreibung setzt sich aus den Bestandteilen Test Configuration, Test Description und Test Data zusammen. Die Testbeschreibung kann entweder grafisch oder mittels textueller Spezifikation erfolgen. Ein großer Vorteil von TDL ist deren Erweiterbarkeit auf andere Domänen, z. B. Industrie 4.0. Auch wird durch die ETSI eine Referenzimplementierung der TDL als Open Source zur Verfügung gestellt.

Bei der Testing and Test Control Notation Version 3 (TTCN-3) handelt es sich um eine Beschreibungssprache zur Spezifikation von ausführbaren Testfällen zum Test von Kommunikationsprotokollen und Services, wobei auch APIs von Softwarekomponenten angesprochen werden können. Insbesondere aufgrund der Möglichkeit zur gleichzeitigen Ansteuerung und Beobachtung von mehreren Testpunkten erscheint der Einsatz von TTCN-3 für Industrie-4.0-Tests als geeignet. Neben reinen funktionalen Tests können mittels TTCN-3 auch Interoperabilitäts-, Robustness- und Performance-Tests durchgeführt werden. Dabei ist TTCN-3 schon heute bei der Testautomatisierung im Bereich der Kommunikationstechnik etabliert.

3.3.3 Erläuterung

Obwohl die beiden Standards TDL und TTCN-3 auf den ersten Blick für die Spezifikation von Testfällen für Industrie 4.0 auf unterschiedlichen Abstraktionsniveaus geeignet scheinen, sind weiterführende Analysen unerlässlich. Insbesondere ist hier zu überprüfen, ob domänenspezifische Anforderungen in diesem Kontext bestehen, die einer Verwendung entgegenstehen oder etwaige Anpassungen zur Folge haben.

3.4 Testautomatisierung

3.4.1 Einführung

Ein Testprozess kann in die Aktivitäten Testplanung und -steuerung, Testanalyse und -design, Testrealisierung und -durchführung und Testauswertung und -bericht (siehe fundamentaler Testprozess in Bild 3) eingeteilt werden.

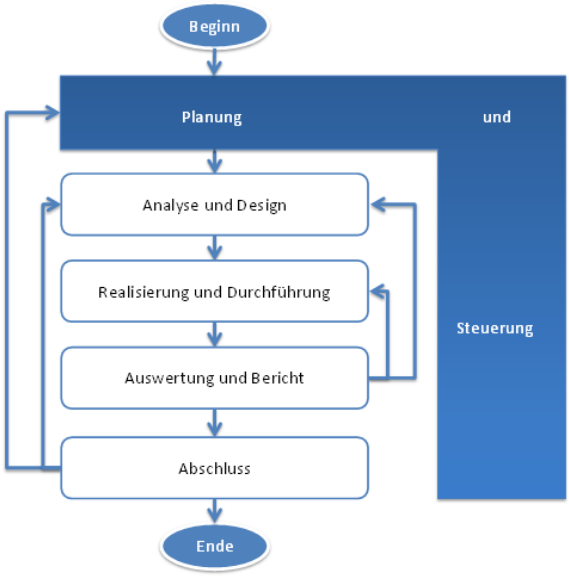

Bild 3. Fundamentaler Testprozess nach ISTQB®

Testautomatisierung betrifft prinzipiell alle Aktivitäten innerhalb eines Testprozesses. In der Praxis steht aber aktuell die Automatisierung der Aktivitäten Testrealisierung und -durchführung sowie Testauswertung und -bericht im Vordergrund. Des Weiteren gibt es Anstrengungen, die Testerstellung innerhalb der Aktivität Testplanung durch Methoden der (modellbasierten) Testgenerierung zu automatisieren.

Testautomatisierung bietet Effizienzvorteile, wenn eine große Anzahl ähnlicher Testfälle oder dieselben Testfälle nach Softwaremodifikationen routinemäßig durchgeführt werden müssen (Regressionstests). Daneben wird die Flexibilität eines Entwicklungsprozesses durch Testautomatisierung stark erhöht, da schnelles, aufwandarmes und systematisches Feedback zu realisierten Änderungen erfolgt. Somit ist die Automatisierung des Testens ein unverzichtbarer Bestandteil der agilen Entwicklung.

In der Praxis verwendete Testautomatisierungssysteme sind dabei technologisch meist individuell realisiert. Es fehlt, neben der TTCN-3 [8], an Normen und Standards. So ist aktuell ein Austausch von Testdaten zwischen Tools verschiedener Hersteller nur schwierig realisierbar.

3.4.2 Bedeutung

Testautomatisierung wird zukünftig eine fundamentale Rolle bei der Bereitstellung qualitätsbewusster Systeme und Funktionen spielen. Zur Sicherung der korrekten Funktionsweise brauchen diese einen umfassenden, systematischen und vor allem automatisierten Test. In Anbetracht einer zu erwartenden hohen Rekonfigurationsrate ist kontinuierliches Testen unverzichtbar. Der Zwang zur Automatisierung ergibt sich zudem aus der Komplexität zukünftiger Systeme, die eine Vielzahl von Testfällen für die Qualitätssicherung benötigen. Sowohl die Erstellung dieser Testfälle als auch die Testdurchführung ist nur durch hochgradige Automatisierung mit beherrschbarem Aufwand zu realisieren. Zur effizienten und reproduzierbaren Durchführung von Tests in komplexen und sich ändernden Umgebungen werden modellbasierte Verfahren bei der Testautomatisierung zunehmend an Relevanz gewinnen.

3.4.3 Erläuterung

Die Herausforderung an zukünftige Testsysteme für den Einsatz bei Industrie-4.0-Applikationen oder -Funktionen ist deren hochgradige Vernetzung. Testsysteme müssen diese Vernetzung beherrschen und gegebenenfalls steuern können. Heutzutage ist der klassische Testansatz des Black-Box-Testens noch zeitgemäß, aber für das Testen hochgradig vernetzter Systeme ungeeignet. Wichtige Testfunktionen, wie die Identifizierung fehlerhafter Komponenten und Teilsysteme, gezielte Fehlerindikationen bei verteilten Systemen etc., sind mit diesen Ansätzen sowohl bei der Testbeschreibung und -spezifikation als auch bei der Testdurchführung nicht realisierbar.

3.5 Modellbasiertes Testen

3.5.1 Einführung

Die Methode des modellbasierten Testens wurde ursprünglich für die automatische Generierung von Testfällen aus Spezifikationsmodellen eingeführt [9]. Ein Spezifikationsmodell beinhaltet dabei das geforderte Verhalten eines Testobjekts, die dabei unabhängig von der Implementierung aus den Anforderungen abzuleiten ist. Geeignete Algorithmen erstellen dann automatisiert systematische Testdaten und zugehörige Testsequenzen (siehe auch Bild 4). Hierfür existieren bereits akademische und auch kommerzielle Werkzeuge. Gängige Modellierungsnotationen sind dabei grafenbasierte Modelle wie Zustandsmaschinen, Aktivitätsdiagramme, Petrinetze etc. Es werden aber auch textuelle

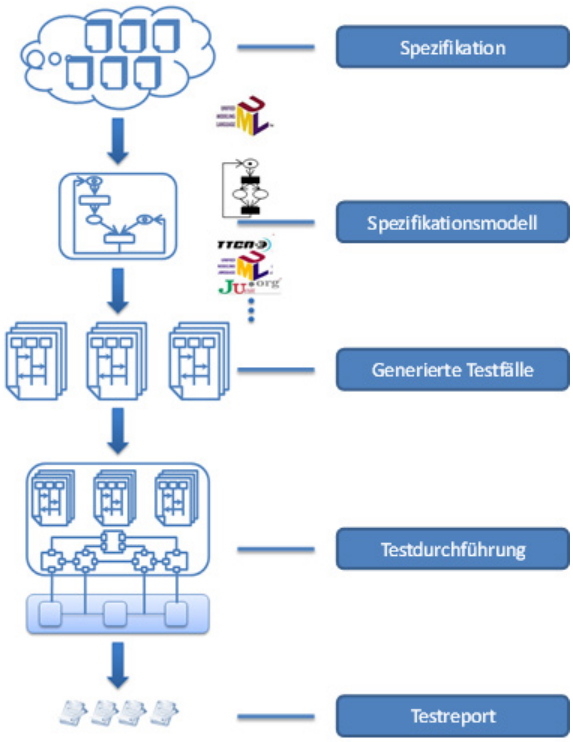

Bild 4. Modellbasierter Testprozess

Modellierungssprachen als Eingabemodell eines Testgenerators verwendet.

Ein Spezifikationsmodell muss korrekt in Bezug zu den Anforderungen sein, sonst können keine korrekten Testfälle abgeleitet werden. Die notwendige Verifizierung ist ein nicht trivialer Prozess. Dabei können Methoden der manuellen Prüfung (Reviews, Modellierungsmuster etc.), der Simulation und der automatisierten formalen Verifikation mithilfe von Werkzeugen zum Model-Checking und/oder automatischen Beweisens verwendet werden.

Im Zuge weiterer Entwicklungen werden unter modellbasiertem Testen auch weitere Methoden verwendet. Es gibt einige Möglichkeiten, Modelle innerhalb eines Testprozesses einzusetzen (Modellierung der Testarchitektur, der Testumgebung, der Testdaten aber auch der Testfälle, siehe UML-Testprofil [10]). Zunächst findet eine Vereinfachung der Erstellung und Dokumentation von Testaktivitäten statt, da durch das Modellieren eine Konzentration auf wesentliche Aspekte erfolgt. Die virtuelle Inbetriebnahme arbeitet ebenfalls mit Modellen, um Anlagen der Fabrikautomatisierung frühzeitig einem fundierten Test zu unterziehen. In erster Linie werden hier SPS-Programme und/oder Robotersteuerungen getestet.

3.5.2 Bedeutung

Modellbasierte Testmethoden besitzen ein hohes Potenzial, Testprozesse in der Praxis aufwandsarm und sehr effizient zu gestalten.

Allerdings werden sehr hohe Anforderungen an den Testingenieur gestellt. Die manuelle Modellierung kann schnell sehr umfangreich und komplex werden und eine Einstiegshürde für den gesamten modellbasierten Testprozess darstellen.

Tabelle 2. Einsparpotenzial durch modellbasiertes Testen nach [5]

Studie	Aufwand ohne MBT	Aufwand mit MBT	Einsparung durch MBT
Ericsson/Conformiq	20 h pro Testfall	5,5 h pro Testfall	73 %
Siemens/Trapeze	2,67 h pro Testfall	0,67 h pro Testfall	75 %
Sepp.med	2,04 h pro Testfall	1,36 h pro Testfall	43 %
Microsoft	2,37 Personentage pro Anforderung	1,38 Personentage pro Anforderung	42 %
Conformiq/Forrester	6.396.565 $ insgesamt	1.288.794 $ insgesamt	30 % bei Testfallspezifikation 84 % bei Testfallwartung

3.5.3 Erläuterung

Aktuell ist die Methode des modellbasierten Testens noch nicht vollständig in der Praxis eingeführt. Das liegt in erster Linie an der hohen Einstiegshürde des Modellierens des Spezifikationsmodells, das häufig aus textuell beschriebenen Anforderungen abgeleitet werden muss. Hier wäre eine Unterstützung von Testingenieuren bzw. -modellierern sinnvoll, um den Schritt der Modellierung wesentlich zu vereinfachen. Dazu bestehen Forschungsbestrebungen in natürlicher Sprache spezifizierte Anforderungen automatisiert in geeignete Modelle zu übertragen (Natural Processing Language).

3.6 Metriken

3.6.1 Einführung

Systeme und Prozesse im Kontext von Industrie 4.0 sind häufig sicherheitskritisch und überaus komplex. Umso wichtiger ist das Testen der eingesetzten Werkzeuge und der Software, um gewisse Qualitäts- und Zuverlässigkeitskriterien nachweisen zu können. Zur Beurteilung der Produktqualität können unter anderem Metriken betrachtet werden. Dabei besteht die Herausforderung darin, geeignete Metriken sinnvoll so zu kombinieren, dass die Qualität eines Systems bzw. Prozesses fundiert anhand weniger Kennzahlen bewertet werden kann. Hierzu bedarf es neuer Richtlinien, die für den Industrie-4.0-Kontext geeignete Metriken und Methoden identifizieren.

3.6.2 Bedeutung

Trotz einer steigenden Verflechtung die Interoperabilität der Systeme zu gewährleisten, stellt eine große Herausforderung dar. Daher kommt dem Testen und der Verifikation der einzelnen Komponenten eine entscheidende Rolle zu. Metriken bieten die Möglichkeit, Qualitäts- und Zuverlässigkeitsanforderungen formal zu beschreiben und so die Qualität eines Produkts nachvollziehbar nachzuweisen. Daher erscheint es sinnvoll, geeignete Metriken zu identifizieren bzw. neu zu entwickeln, um das Testen im Kontext der Industrie 4.0 systematisch und nachvollziehbar zu gestalten.

3.6.3 Erläuterung

Metriken werden in der Softwareentwicklung bereits extensiv eingesetzt und auch im Controlling von Softwareprojekten zur Beurteilung der Produktqualität zu Rate gezogen. Es muss untersucht werden, ob und wie sich existierende Konzepte in den Kontext der Industrie 4.0 übertragen lassen. Um möglicherweise geeignete Metriken aus dem Bereich der Softwareentwicklung für die Verwendung im Kontext der Industrie 4.0 identifizieren zu können, wird im Folgenden ein kurzer Überblick über gebräuchliche Testmethoden und Metriken aus dem IT-Bereich gegeben.

Zu den Testmethoden, aus denen Metriken generiert werden können, zählen unter anderem funktionale Testfälle auf Basis der Anforderungen, I/O- und Kommunikationstests zwischen verschiedenen Komponenten sowie Performance-Tests. Sie betrachten das SuT als Black-Box. Eine sehr häufig genutzte Metrik ist die Anforderungsabdeckung (engl. requirements coverage). Hierbei wird gemessen, wie viele der formal spezifizierten Anforderungen durch Testfälle abgedeckt werden. Bei zusätzlicher Betrachtung der erfolgreich und fehlerfrei durchgeführten Testfälle einer Anforderung ergibt sich daraus die Anforderungskonformität eines Produkts. Mithilfe der Anforderungskonformität lässt sich bereits eine Aussage über die Qualität des Produkts treffen, da eine niedrige Konformität mit hoher Wahrscheinlichkeit mit einer geringen Produktqualität einhergeht.

Neben den auf Anforderungen beruhenden Tests gibt es auch eine Reihe von Tests, die direkt aus dem Quellcode abgeleitet werden und damit das SuT als White-Box behandeln. Diese gehen folglich von der Implementierung aus. Zur Herleitung der Testfälle existieren unterschiedliche Kriterien:

- jede Zeile – Line Coverage
- alle Statements – C0
- alle Verzweigungen mit allen Ausgängen – C1
- jede Speicherplatzzuweisung – D0; etc.

Diese Kriterien dienen dabei direkt als Metrik für die Testabdeckung, daher werden sie auch Abdeckungstests (engl. code coverage) genannt. Dabei lassen sie jedoch nur Rückschlüsse auf die korrekte Implementierung zu und treffen eben keine Aussagen über die tatsächliche Fehlerfreiheit der untersuchten Software – ein Umstand, der eventuell deutlicher kommuniziert werden müsste. Ob der Code auch den Anforderungen entspricht, ist somit auf diesem Wege nicht abschätzbar. Hierzu bedarf es der oben erwähnten funktionalen Tests.

Eine weitere, wichtige Gruppe von Testmetriken stammt aus dem Bereich des probabilistischen Testens. Grundlage aller probabilistischer Testmethoden ist die Erkenntnis, dass eine vollständige Fehlerfreiheit einer Software in der Praxis meist weder verlangt wird noch nachgewiesen werden kann. Stattdessen ist es ausreichend, eine bestimmte Seltenheit des Versa-

gens in der jeweiligen Betriebsumgebung nachzuweisen. Hierzu werden statistische Verfahren eingesetzt, die das SuT aus einer Black-Box- oder Systemperspektive heraus betrachten. Ein Maß für die Zuverlässigkeit ist in diesem Fall die Versagenswahrscheinlichkeit. Weitere Kenngrößen sind nach IEC 60050 beispielsweise die durchschnittliche Fehlereintrittszeit (Mean Time To Failure, kurz MTTF), die die mittlere Betriebsdauer bis zum Ausfall einer Komponente angibt, oder auch die Durchschnittszeit zwischen Fehlern (Mean Time Between Failure, kurz MTBF), die Auskunft über die mittlere Betriebsdauer zwischen Ausfällen gibt. Nachteil all dieser Kenngrößen ist ihre statistische Herkunft. Um sie zu ermitteln, müssen bereits viele Erfahrungswerte mit den jeweiligen Komponenten vorliegen oder entsprechend viele Testreihen (mehrere tausend) durchgeführt werden. Dies wird insbesondere bei kleinen Losgrößen schnell unwirtschaftlich.

Diese Auflistung unterschiedlicher Metriken ist nicht vollständig und soll veranschaulichen, dass bereits eine Vielzahl an Metriken und Kenngrößen aus dem IT-Bereich existiert. Eine aktuelle Herausforderung besteht darin, geeignete Softwaremetriken für den Einsatz in Industrie 4.0 zu identifizieren und gegebenenfalls zu adaptieren. In Bezug auf Hardwaremetriken sollte ein solcher Prozess nicht notwendig sein, da sich diese nicht nennenswert unterscheiden sollten. Ein wichtiger Aspekt, in dem sich beide Bereiche unterscheiden, betrifft die Interoperabilität zwischen Komponenten. Die Bewertung der Interoperabilität eines Systems im Kontext Industrie 4.0 ist mit den gängigen Metriken nur schwer möglich. Der Test gegen standardisierte Schnittstellen wird voraussichtlich deutlich an Bedeutung gewinnen. Eventuell sind für solche Schnittstellen- oder auch Interoperabilitätstests sowie auch zur Bewertung von Security-Fragestellungen neue Metriken erforderlich, die speziell auf den Anwendungsfall Industrie 4.0 abgestimmt sind.

Literatur

[1] http://wirtschaftslexikon.gabler.de/Definition/komplexitaet.html

[2] SEW Industrie 4.0 White Paper

[3] IBM Global CEO Study: The enterprise of the future

[4] Tilo Linz: Testen von SCRUM-Projekten, dpunkt-Verlag

[5] „Modellbasiertes Testen: Hype oder Realität?“ Stephan Weißleder, Baris Güldali, Michael Mlynarski, Arne-Michael Törsel, David Faragó, Florian Prester, Mario Winter, ObjektSpektrum 06/2011

[6] http://www.etsi.org/technologies-clusters/technologies/test-description-language

[7] http://www.etsi.org/technologies-clusters/technologies/testing/ttcn-3

[8] ETSI: Testing and Test Control Notation Version 3 (TTCN-3), www.ttcn-3.org

[9] Pretschner, A.: Zum modellbasierten funktionalen Test reaktiver Systeme, Dissertation, TU München, München, 2003

[10] Object Management Group: UML Testing Profile, http://www.omg.org/technology/documents/formal/test_profile.htm/, 2007

Technische Regeln

ANSI/IEEE 829:2008 Standard for Software Test Documentation

DIN EN 61131-3:2013-04 Automatisierungssysteme in der verfahrenstechnischen Industrie – Werksabnahme (FAT), Abnahme der installierten Anlage (SAT) und Integrationstest (SIT)

DIN EN 62443 IT-Sicherheit für industrielle Automatisierungssysteme (IEC 62443)

DIN IEC 60050 Internationales Elektrotechnisches Wörterbuch

IEEE 829:2008 Standard for Software and System Test Documentation

ISO/IEC/IEEE 29119-3:2013-09 Software-und Systemengineering – Software-Test – Teil 3: Testdokumentation

Der VDI

Sprecher, Gestalter, Netzwerker

Die Faszination für Technik treibt uns voran: Seit 160 Jahren gibt der VDI Verein Deutscher Ingenieure wichtige Impulse für neue Technologien und technische Lösungen für mehr Lebensqualität, eine bessere Umwelt und mehr Wohlstand. Mit rund 150.000 persönlichen Mitgliedern ist der VDI der größte technisch-wissenschaftliche Verein Deutschlands. Als Sprecher der Ingenieure und der Technik gestalten wir die Zukunft aktiv mit. Mehr als 12.000 ehrenamtliche Experten bearbeiten jedes Jahr neueste Erkenntnisse zur Förderung unseres Technikstandorts. Als drittgrößter technischer Regelsetzer ist der VDI Partner für die deutsche Wirtschaft und Wissenschaft.

Weitere VDI-Publikationen zum kostenlosen Download

www.vdi.de/publikationen

VDI-Stellungnahme „Bedeutung der digitalen Transformation für die chemische Industrie“

VDI-Studie „The Digital Engineer and The Changing Workplace“

VDI-Standpunkte „SMART GERMANY – Arbeit in der Digitalen Transformation“

VDI-Statusreport „Industrie 4.0 – Begriffe/Terms“

VDI-Statusreport „Industrie 4.0 Service Architecture – Basic concepts for interoperability”

VDI-Statusreport „Anwendungsszenario DDA – Durchgängiges und dynamisches Engineering von Anlagen“

VDI Statusreport „Arbeitswelt Industrie 4.0“

VDI-Statusreport „Industrie 4.0 Components – Modeling Examples”

VDI-Statusreport „Chancen mit Big Data – Best Practice“

VDI-Statusreport „Chancen mit Big Data – Use Cases“

VDI-Statusreport „Geschäftsmodelle für Industrie 4.0 – Digitale Chancen und Bedrohungen“

VDI-Handlungsfelder „Additive Fertigungsverfahren“

VDI-Thesen und Handlungsfelder „Ingenieurausbildung für die Digitale Transformation“

VDI Verein Deutscher Ingenieure e.V.
VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik (GMA)
Dr.-Ing. Thomas Sowa
Tel. +49 211 6214-223
sowa@vdi.de
www.vdi.de/gma