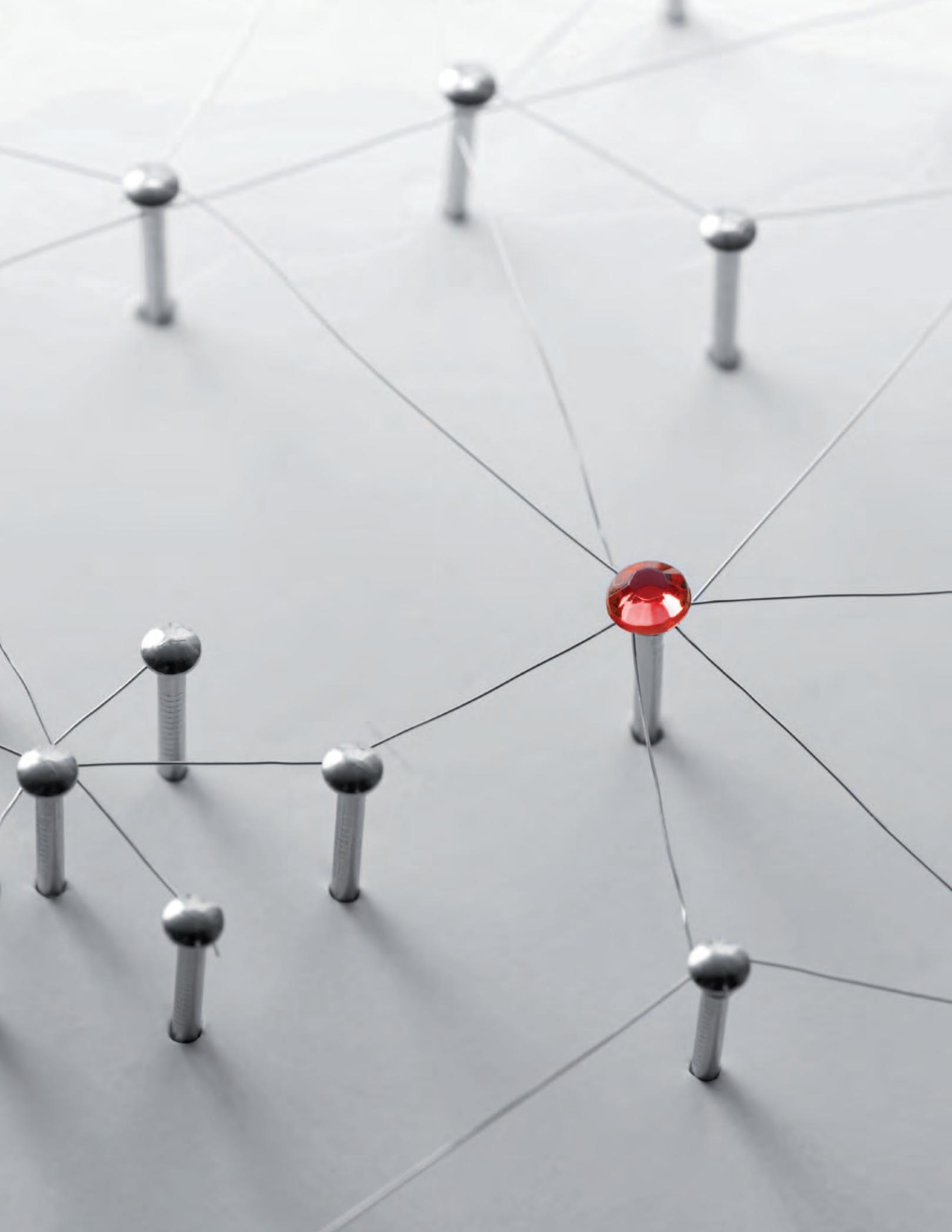


*Sonderedition des SQ Magazins
anlässlich 20 Jahre ASQF e.V.*



**JAHRE
IM DIENST
VON
QUALITÄT**



Qualität von Anfang an

Editorial

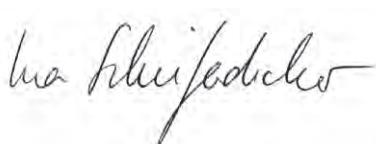
Das Ganze ist mehr als die Summe seiner Teile – noch nie passte dieser Ausspruch besser als in der heutigen Zeit. Was früher dem reinen Austausch diente, ist heute zum Fundament unseres gesellschaftlichen Miteinanders geworden. Informations- und Kommunikations-Systeme spiegeln und beeinflussen unsere Umwelt. Soziale Interaktion, Produktion und Handel, öffentliche Verwaltung – sie alle basieren auf Software-Lösungen, die miteinander vernetzt und voneinander abhängig sind.

Die Diskussion über einzelne Qualitätsaspekte von Software reicht deshalb schon lange nicht mehr aus. Es geht um das große Ganze. Je größer und weitreichender die Vernetzung unserer Systeme, desto mehr müssen sie neben Funktionalität und Stabilität auch sicherheitskritische Funktionen erfüllen. Hohe Anforderungen in diesen Bereichen schlagen sich natürlich in den entsprechenden Aufwänden und damit Kosten nieder. Wie könnte eine Lösung in diesem Bereich aussehen? Wollen wir einen Kompromiss zulassen?

Im ASQF haben sich Experten zusammengeschlossen, um über diese Fragen zu diskutieren und gemeinsam an Lösungen zu arbeiten. Dank des tatkräftigen Engagements unserer ASQF-Mitglieder kann der Verein seit zwei Jahrzehnten als Taktgeber seinen Anteil an der Qualitätssicherung in der Software-Entwicklung leisten. All denjenigen, die sich mit ihrem Wissen und Erfahrungsschatz einbrachten und einbringen, gilt an dieser Stelle ein großer Dank. Wir danken besonders auch den ASQF- und iSQI-Mitarbeiterinnen sowie -Mitarbeitern, Präsidien und ASQF-Beiräten, die sich immer wieder auf ein Neues für die Weiterentwicklung des Vereins einsetzen.

Vor uns liegen weitere aufregende Jahre und zahlreiche Themen in Bezug auf Software- und System-Qualität. In der vorliegenden SQ-Magazin-Sonderedition „20 Jahre im Dienst von Qualität“ widmen wir uns ausgewählten Aspekten. Für die Beiträge konnten wir allesamt hochkarätige Persönlichkeiten gewinnen, die zum Gelingen dieser Ausgabe beigetragen haben. In diesem Sinne wünschen wir Ihnen eine gute Lektüre.

Herzlichst
Ihre



Prof. Dr.-Ing. Ina Schieferdecker
ASQF-Präsidentin



Stephan Goericke
ASQF-Hauptgeschäftsführer



Inhalt

7

20 Jahre Software-Qualität und Fortbildung

Prof. Dr. Bernd Hindel

10

20 Jahre ASQF – Paradigmenwechsel in der Software-Qualität

Christian Knüvener / Thomas Thurner

13

ASQF: Heute und zukünftig

Die Bedeutung von Qualität in der Zukunft

Rudolf van Megen

16

System-Qualität in der Gigabit-Gesellschaft

Ein Blick auf den ASQF in 20 Jahren

Prof. Dr.-Ing. Ina Schieferdecker

22

Software-Qualität als entscheidendes Erfolgskriterium

Dr. Georg Heidenreich

27

Der ASQF – Ein Netzwerk aus Experten

Norbert Kastner

30

Der Weg ist das Ziel

Dr. Stefan Bunzel

33

Digitale Transformation – Der Mensch, das Know-how und die Wünsche

José Díaz Delgado

40

Anspruchsvolle Projekte erfolgreich führen

Illusion oder Wirklichkeit?

Katrin Greßer / Ewa Sadowicz

47

POSMuT – Ein ganzheitliches Modell

zur Prozessverbesserung und Erzielung von Automotive SPICE® 3.0-Konformität

Horst Pohlmann

Inhalt

53

No risk, no fun(ction), no future!

(Wie) lassen sich software-basierte Systeme in einer komplexer werdenden Welt noch beherrschen?

Michael "Pul" Paulsen / Florian Prester

64

Software-Qualität in Zeiten von Digitalisierung und Internet der Dinge

Matthias Schneider / Stephan Oswald

70

Sichere Software im Umfeld von Industrie 4.0 Best Practices für sichere Entwicklung, Einsatz und Wartung von Maschinen-Software

Prof. Dr. Sachar Paulus

84

Wanted: Testspezialisten dringend gesucht!

Graham Bath

95

Gemeinsam sind wir stärker: GTB und ASQF gestern, heute und morgen

Dr. Frank Simon

60

Requirements-Engineering

Ein interaktives, iteratives Schauspiel in vier Akten

Chris Rupp / Karin Schlör / Christian Pikalek

66

Networking – die zeitgemäße Währung

Warum lokales Netzwerken und Expertenrunden so wichtig sind

Alexander Weichselberger

76

Veränderte Anforderungen an die Qualitätssicherung von Data-Science-Lösungen

Prof. Dr. Andreas Schmietendorf / Jan Hentschel

89

Das Qualitäts-Tripel: Design by Contract, Clean Code, Lean Testing

Prof. Dr. Andreas Spillner / Prof. Dr. Ulrich Breymann



**ASQF
IM
WANDEL**

20 Jahre Software-Qualität und Fortbildung

7

Prof. Dr. Bernd Hindel

– ASQF-Präsident von 1996 bis 2007 / CEO, Method Park Holding AG –



„Qualität ist die Gesamtheit von Eigenschaften eines Produkts oder einer Dienstleistung bezüglich ihrer Eignung, festgelegte und vorausgesetzte Eigenschaften zu erfüllen.“ So lautete die Definition für Qualität in der Version der ISO 9000, die damals gültig war, als der ASQF gegründet wurde. Diese Definition auf Software-Entwicklung und Software-Produkte anzuwenden, war das Kernstück des europäischen ESPITI Programmes (European Software Process Improvement Training Initiative), das Mitte der 1990er Jahre europaweit Veranstaltungen zum Thema Software-Qualität förderte. Die 3SOFT GmbH organisierte derartige Veranstaltungen in Erlangen, Nürnberg, Fürth, München und Stuttgart. Die Fördermittel wurden damals in ECU (European Currency Unit) ausgewiesen, denn den EURO gab es noch nicht. Aus diesen ESPITI-Veranstaltungen sind 1995 die Arbeitskreistreffen entstanden, die am 10.10.1996 zur Vereinsgründung des ASQF führten. In den vergangenen 20 Jahren ist es dem ASQF hervorragend gelungen, mit seinen zahlreichen Arbeitskreistreffen, Fachgruppen- und Regionalgruppentreffen und anderen erfolgreichen Veranstaltungen, wie den ASQF-Days, seinem Satzungszweck treu zu bleiben und die ESPITI-Idee weiterzuentwickeln. Wenn im Sinne der obigen Qualitätsdefinition für Software „festgelegte und vorausgesetzte Eigenschaften zu erfüllen“ sind, dann geht es um Anforderungsmanagement, Umsetzung von Anforderung (Implemen-

tierung) und den Nachweis ihrer Umsetzung (Testen). Der ASQF hat es in den letzten beiden Jahrzehnten geschafft, die Trends, Innovationen und etablierten Techniken des Software Engineerings immer zu Themen seiner gemeinnützigen Veranstaltungen zu machen:

Die 1990er

Strukturiertes Design, prozedurale Programmiersprachen (Fortran, Pascal, C), das V-Modell waren „State of the Art“. Die ersten Reifegradmodelle (CMM, BOOTSTRAP) basierend auf dem V-Modell eroberten die große Software-Entwicklung in den USA und Indien sowie bei BOSCH und Siemens. Die Objektorientierung kam gerade in der Industrie an (C++, Java). Rumbaugh, Jacobson und Booch vereinheitlichten ihre Ansätze für objektorientiertes Design zur sogenannten UML.

Der ASQF machte all das zu den Themen seiner ersten Fachgruppen und der jährlichen Konferenz CONQUEST (Conference on Quality Engineering in Software Technology). Die erste Internationalisierung des ASQF begann mit der Verpflichtung von Referenten wie Tom DeMarco und dem Mitwirken bei der „European Organisation for Quality – Software Group“ und deren Konferenzreihe „World Congress for Software Quality“. >>



Glückwunsch ASQF

**„ICH WÜNSCHE DEM ASQF UND SEINEN AKTIVEN MITGLIEDERN,
DASS SIE AUCH IN DEN NÄCHSTEN 20 JAHREN MIT DER
THEMENAUSWAHL FÜR FACHGRUPPEN UND VERANSTALTUNGEN
GENAU SO RICHTIG LIEGEN WIE IN DEN ERSTEN 20 JAHREN.“**

Prof. Dr. Bernd Hindel, ASQF-Präsident a.D. / CEO, Method Park

2000 bis 2010

Objektorientierte Ansätze verdrängten prozedurales Programmieren in der Industrie. CMMi, Automotive SPICE®, UML, C++, Java, C# (C-Sharp) wurden zu Industriestandards. Es beginnt die Diskussion um agile Ansätze: traditionelle, V-Modell-basierte Entwicklungsprozesse versus Management und Engineering-Ansätze wie Scrum, X-treme Programming. SAP führt 2008 Scrum flächendeckend ein. Die etablierten Reifegradmodelle CMMi und SPICE® ermöglichen bzw. erwarten eine klare Rollendefinition gemäß V-Modell, um Verantwortung in Projekten festzumachen. Dies ist die Grundlage für den Erfolg der Zertifizierung von IT-Professionals gemäß Rollenanforderung. Beim ASQF werden die ersten Lehrpläne für den Certified Tester in Deutsch geschrieben. Andere Rollenzertifizierungen folgen. Der Tester bleibt jedoch die erfolgreichste Rollenzertifizierung und ermöglicht die zweite Internationalisierungswelle für den ASQF durch die weltweite Verbreitung des „Certified Testers“ und Mitwirkung in den nationalen und internationalen Boards und als Zertifizierer.

2010 bis 2020

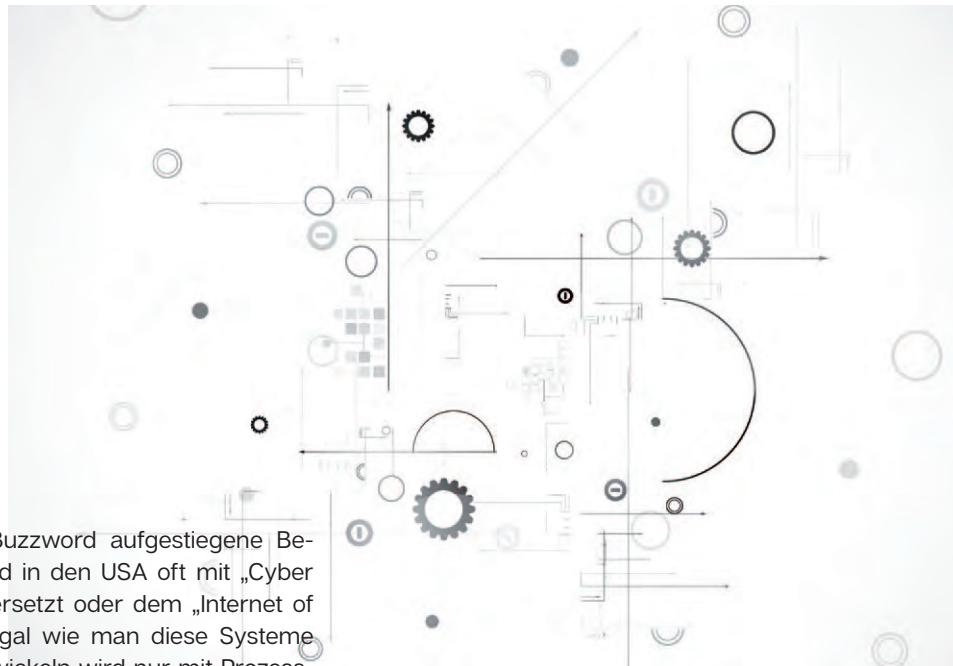
Scrum etabliert sich als die „Agile Management Methode“ für Software-Projekte. Software übernimmt immer mehr sicherheitskritische Anwendungen – im Sinne von „Safety“ – in Alltagsprodukten, wie z.B. im Automobilbereich. Durch die Allgegenwärtigkeit von Smartphones wird die Vision des „Internet of Things“ Realität. Innovative Produkte muss man

heute mit einem Smartphone steuern können. Für die Software-Entwicklung heißt das jedoch, Sicherheit – jetzt aber im Sinne von „Security“ – wird immer wichtiger. Entwicklungsmodelle basierend auf dem V-Modell haben ausgedient. Der Entwicklungsprozess muss in ein „Continuous Development“ eingebettet werden, das eine Vielzahl von „Suppliern und Providern“ mit einschließt. Das bringt Modelle wie SPICE® und CMMi an ihre Grenzen.

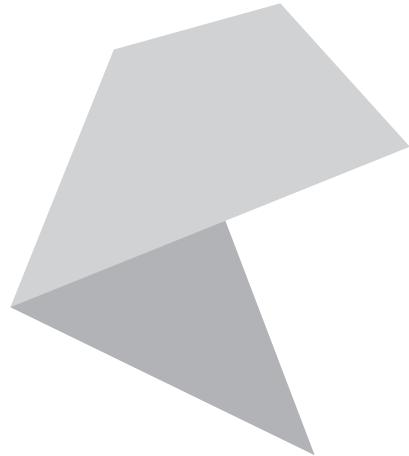
Der ASQF schafft es erneut, Fragen, Diskussionen und zum Teil Antworten in seinen Fachgruppen und anderen Veranstaltungen zu präsentieren. Jetzt ist es auch an der Zeit, eine richtige Internationalisierung mit Niederlassungen im Ausland aufzubauen. Über die Standorte seiner Tochter, der iSQI GmbH, wird die ASQF-Idee weiterverbreitet.

... und nach 2020

Die Innovationen im Software Engineering werden sich weiter verschieben, weg von Implementierungsansätzen und Programmiersprachen, über die Design-Methoden und Design-Sprachen, hin zu Prozess-Modellen und Prozess-Sprachen. Die immer komplexeren Projekte für die Welt mit dem Internet der Dinge erwarten dies. Nur so kann Effizienz und Qualität beibehalten oder gesteigert werden, in einer vernetzten Welt mit immer größeren Problemstellungen.



Der in Europa zum Buzzword aufgestiegene Begriff „Industry 4.0“ wird in den USA oft mit „Cyber Physical Systems“ übersetzt oder dem „Internet of Things“ zugeordnet. Egal wie man diese Systeme bezeichnet, sie zu entwickeln wird nur mit Prozessmodellen gehen, die es erlauben „Continuous Development“ für Entwicklungs- und Nutzungsphasen der Systeme flexibel, anpassbar und reaktiv für dezentrale Entwicklungs- und Betreiberteams zu modellieren. Die Methode dazu könnte „Prozessmanagement 4.0“ heißen. Wo kann man sich darüber informieren? Fragen Sie doch mal in ein bis zwei Jahren beim ASQF nach. |



KONTAKT ZUM AUTOR

bernd.hindel@methodpark.de

20 Jahre ASQF – Paradigmenwechsel in der Software-Qualität

Christian Knüvener

– Principal Consultant, Knüvener Consulting –

Thomas Thurner

– ASQF-Präsident von 2007 bis 2013 –

War vor 20 Jahren das Thema Software-Qualität noch eine Nische für einzelne Vordenker, geprägt von individuellen Vorgehensweisen, so bedarf es heute zu Beginn einer vernetzten „Welt 4.0“ (Internet of Things) weitreichende Qualitätsstandards, um diese Vernetzung verlässlich zum Funktionieren zu bringen und am Laufen zu erhalten.

Der ASQF hat in den zurückliegenden Jahren mit großem Engagement zur Entwicklung von Standards in der Software-Qualität beigetragen. Schon früh wurde erkannt, dass eine wesentliche Grundlage dafür eine standardisierte und nachweisbare Experten-Qualifikation darstellt. Aber auch der Informationsaustausch unter den Experten spielt eine wichtige Rolle. Sprechen die Experten eine gleiche Sprache und verstehen sie unter den Begriffen die gleiche Semantik, können standardisierte und überprüfbare Prozesse zur Qualitätssicherung wesentlich beitragen. In beiden Bereichen hat der ASQF seinen Beitrag geleistet – über 30 Zertifikate stehen heute bei der iSQL GmbH im Portfolio und wesentliche Beiträge zur Etablierung von z.B. SPICE® und INTACS® wurden vom ASQF initiiert bzw. umgesetzt. Der Anteil der Produkte, die Softwareanteile enthalten, steigt zunehmend und der Vernetzungsgrad der Produkte „explodiert“ geradezu. Hardware, Software, Nutzung und Service verschmelzen immer mehr zu einem virtuellen Produkt, bei dem der Hardwareanteil immer weniger Bedeutung zukommt und der Service zunehmend im Vordergrund steht. Immer mehr Software-Komponenten und Services unterschiedlichster Hersteller und Provider bauen aufeinander auf und hängen somit auch voneinan-

der ab. Eine entsprechend standardisierte und integrale Betrachtung auf allen Ebenen von Software-Komponenten und Software-Services muss die Produkt- und Prozess-Qualität über Firmengrenzen und sogar über Branchen hinweg absichern. Durch diesen integrierten Ansatz der Qualitätsbetrachtung entstehen neue Möglichkeiten und Nutzen einer zuverlässigen, weil standardisierten Vernetzung für ein Internet of Things.

Was zeichnet eine integrierte Qualitätssicherung aus?

- Standards in der Entwicklung für Kern- und Stützprozesse auf Produkt-, System- und Subsystemebene (Software, Hardware und Mechanik),
- Transparenz in der Entstehung, kontinuierliche Reifegraderfassung, selbstorganisierte Quality Gates und gemanagte Flexibilität,
- Managementkultur und Leadership für Stabilität und Nachhaltigkeit zur Erreichung von Entwicklungszügen,
- qualifizierte Mitarbeiter,
- Security gegenüber Missbrauch und Cyber-Kriminalität,
- funktionale Sicherheit der Produkte,
- eine durchgängige Qualitätssicherung entlang der Produktentstehung. Das bedeutet, die konsequente Anwendung der integralen Betrachtung bei allen Entwicklungs- und Servicepartnern (Lieferanten und Kunden – extern sowie intern).



Trefflich kann man darüber streiten, welcher der wichtigste oder schwierigste Punkt in dieser Liste ist. Dies hängt meist von den Produktanforderungen sowie von der jeweiligen Sichtweise und dem Zusammenspiel der Arbeitskulturen ab. Speziell für sicherheitsrelevante Applikationen im Umfeld eines „Internet-of-Things“, bei denen Menschen oder die Umwelt zu Schaden kommen können (z.B. autonomes Fahren), spielt die funktionale Sicherheit die wichtigste Rolle. Diese wiederum kann in einer komplexen vernetzten Welt nur durch eine integrale Qualitätsbetrachtung abgesichert werden.

Unstrittig ist, dass die Anzahl der zu integrierenden Fremdsoftware und -funktionen weiter deutlich ansteigt. In diesem Zusammenhang entwickelt sich die Fähigkeit zur Integration von Fremdsoftware zusehends zu einer der entscheidenden Kernkompetenzen in der Produktentstehung.

Das hierfür notwendige Management der Komplexität, sowohl in der Definition des Produktes selbst

als auch in der Zusammenarbeit mit gegebenenfalls branchenübergreifenden beteiligten Partnern, sowie den einzelnen Phasen des Produktlebenszyklus, kann nur auf Basis eines integralen Qualitätsmanagementsystems mit seinen bereits erwähnten Produkt- und Prozess-Standards funktionieren. Die Integration von bereits implementierten und quasi sofort „im Regal“ verfügbaren Funktionen durch Standard-Software-Komponenten oder auch Free-and-Open-Source-Software konzentriert sich direkt auf die Absicherung der funktionalen sowie nicht-funktionalen Anforderungen, also insbesondere auch auf die der Qualitätsstandards. Sind diese nicht nachweisbar, kann diese Komponente nicht integriert werden, allein schon aus rechtlichen Gründen. Sind die Nachweise jedoch verfügbar, ist der Entwicklungstechnische Aufwand gering und der Nutzen schnell sichtbar. Das Angebot solcher standardisierten Lösungen steigt ständig. Noch behindern die oft komplexen Nutzungsbestimmungen die weitere Verwendung. Diese sind noch zu wenig standardisiert. >>

Bei einigen Innovationsschritten sind keine „fertigen“ Lösungen verfügbar. Sie müssen erst noch entwickelt werden. Oft sind sogar die Anforderungen und benötigen Schnittstellenkompetenzen anfangs noch unklar, so dass flexibel auf Änderungen reagiert werden muss. Dennoch muss die Prozess- und Produktqualität nachweislich erreicht werden. In diesem Fall gilt die bewährte Regel, dass die Qualität der Ergebnisse direkt abhängig von der Qualität der Kommunikation ist. Übertragen auf die gestiegenen Anforderungen der integrierten Qualitätssicherung, bedeutet dies eine transparente, qualitätsorientierte, kontinuierliche Entwicklungszusammenarbeit, wie man sie oft fordert, selten sieht, aber doch zusehends häufiger aufgrund der Notwendigkeiten erlebt.

Fazit

Ist die Software-Qualitätssicherung schon eine Schlüsselkomponente für ein singuläres Produkt, so ist sie es umso mehr für vernetzte und aufeinander aufbauende Services in allen Bereichen des täglichen Lebens, speziell unter Security- und Safety-Gesichtspunkten.

Diesen Anspruch muss der ASQF bei der Betrachtung des Themas Software-Qualität im Fokus haben und die Weiterentwicklung von Qualifizierungszertifikaten und Diskussionsforen im Sinne zukünftiger Standards weiterhin aktiv vorantreiben. |



KONTAKT ZU DEN AUTOREN

knuevener@knuevenerconsulting.com
thomas.thurner@gmx.net

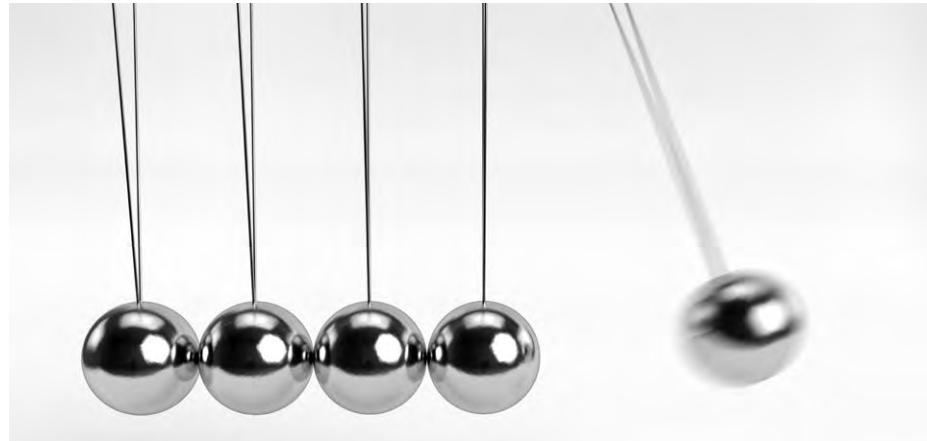
ASQF: Heute und zukünftig

Die Bedeutung von Qualität in der Zukunft

Rudolf van Megen

– ASQF-Präsident von 2013 bis 2015 /

Vorstandsvorsitzender GIMI German ICT & Media Institute e.V. –



Wenn ein Verband bereits seit 20 Jahren existiert und seine Fachzeitschrift schon zehn Jahre erscheint, kann man daraus unterschiedliche Schlussfolgerungen ziehen:

Erstens: Die durch den Verband beziehungsweise vom Magazin aufgegriffenen Inhalte besitzen hohe Relevanz und sind deshalb für die Fachwelt und die Gesellschaft von besonderem Interesse.

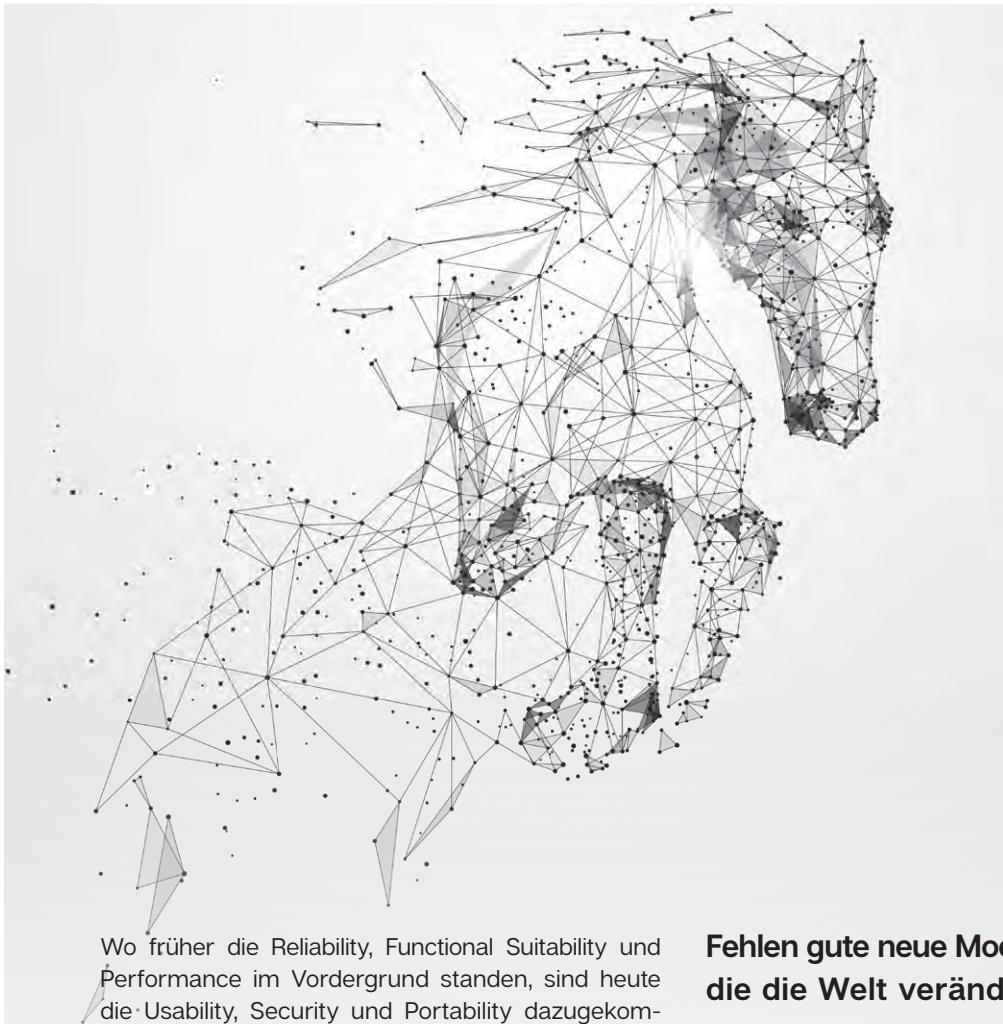
Zweitens: Verband oder Zeitschrift bestehen vornehmlich aufgrund von Tradition.

Ich bin überzeugt, dass der erste Punkt, also die Themen und Inhalte des Verbands, die wichtigsten Gründe für das 20-jährige Fortbestehen des ASQF sind. Wer sich mit den Themen Software, Qualität, Testen und Fortbildung beschäftigt, kommt einfach nicht umhin, sich mit dem ASQF und dann auch mit dem SQ-Magazin zu beschäftigen. Dass der Verein dabei „nur“ über eine 4-stellige Mitgliederzahl verfügt, ist ein Fakt. Viel wichtiger ist jedoch die Frage:

Warum sind es nicht mehr, die sich in einem Expertennetzwerk wie dem ASQF engagieren?

Im deutschsprachigen Raum müsste es eigentlich viel mehr Menschen geben, die sich mit den vorgenannten Themen beschäftigen! Der Anteil der Menschen, die die ständig komplexeren Software-Systeme testen, sich um die Qualität kümmern, liegt bei mindestens 25 bis 35 Prozent der Software-Entwickler – und dazu zählen auch alle aus der neuen Denkwelt, die „nur eine App“ entwickeln (z.B. für Fintechs oder Insurtechs), ein Auto zum selbstfahrenden Vehikel machen oder sich vielleicht um funktionierende Zimmerbuchungen bei airbnb oder hrs kümmern. Das Potential für Verband und Zeitung ist riesig!

Die Bedeutung von Qualität war in der Vergangenheit hoch und sie ist heute durch die steuernde Intelligenz von Applikation oder Applikationssystemen noch viel größer. Konkret müssten sich also viel mehr Menschen mit dem ASQF als Branchenverband beschäftigen und das SQ-Magazin lesen. >>



Wo früher die Reliability, Functional Suitability und Performance im Vordergrund standen, sind heute die Usability, Security und Portability dazugekommen – also nicht weniger, sondern mehr Qualitätsmerkmale sind wichtig und damit auch noch mehr Menschen, die sich damit beschäftigen (müssen).

Ein Ausweg in der Argumentation wäre, es allein auf die deutsche Sprache zu schieben. Aber auch im deutschsprachigen Raum haben wir signifikant mehr Personen als die oben genannten Verbandsmitglieder und potentiellen Lesekandidaten des SQ-Magazins.

Ich habe jedoch manchmal den Eindruck, dass sich die Menschen, die mit Qualität zu tun haben, nicht wirklich um die neuen Aspekte kümmern. Daher ist es umso wichtiger, möglichst global nach neuen Trends zu suchen und über diese auch global zu berichten.

Fehlen gute neue Modelle und Theorien, die die Welt verändern?

Ein paar Beispiele:

1. Die Qualitäts-Anwender haben es auch in zehn, 20 oder 30 Jahren nicht geschafft, mehr in die frühzeitige Fehlerkennung zu stecken: Fehler werden trotz z.B. Agile Development immer noch (viel) zu spät gefunden. Es sind weniger die Fehler im Sprint, als Fehler, die die grundlegenden Zusammenhänge betreffen! Es kommen zwar immer wieder neue Begriffe auf („Early Error Detection“, „Shift left“), aber die Statistiken zeigen immer noch, dass viele Fehler erst nach der Entwicklung gefunden werden, obwohl sie bereits in den Use-Cases/Requirements verursacht wurden.



Glückwunsch ASQF

*„SOLANGE IMMER NEUE SOFTWARE-PRODUKTE ENTWICKELT WERDEN,
WERDEN WIR AUCH IMMER WEITER UNSER WISSEN, UNSERE ERFAHRUNG
UND UNSERE BEGEISTERUNG FÜR DAS SPANNENDE THEMA TESTEN EIN-
BRINGEN. ICH GRATULIERE DEM ASQF ZUM 20-JÄHRIGEN JUBILÄUM
UND SAGE 'HAPPY BIRTHDAY ASQF!'.“*

Rainer Deussen, Capgemini Deutschland GmbH,
stellvertretener ASQF-Fachgruppenleiter Software-Test NRW

2. Die Komplexität dessen, was heute im Zusammenspiel funktionieren muss, wird immer größer: da wo früher eine Applikation war, diskutieren wir heute über komplexe Business-Prozesse, die auf ihre Qualität hin überprüft werden müssen. Wo sind die Qualitäts-Systeme, die das heute bewältigen?
3. Wir sehen neue Organisationsformen – z.B. das Crowd-Testing, bei dem Durchschnitts-Benutzer unvoreingenommen an die Benutzung von Systemen herangehen. Aber auch hier brauchen wir in Zukunft mehr als nur „probieren“, ob etwas funktioniert. Ein weiteres Beispiel ist Cloud-Testing, bei dem manche meinen, alles ginge von allein. Eine weitere neue Organisationsform ist das Internet of Things, in dem viele persönliche Daten verarbeitet werden und Sicherheit einen hohen Stellenwert haben wird.

Wie ist die Schlussfolgerung? Es gab, gibt und wird in Zukunft viele guten Gründe geben, sich mit den Themen von Software, Qualität, Testen und Fortbildung auseinanderzusetzen. Die Bedeutung von Qualität wird in der Zukunft noch herausragender sein, wenn sich Menschen auf Systeme verlassen wollen oder müssen, beispielsweise beim selbstfahrenden Auto.

Viel Erfolg für die nächsten 20 Jahre beim ASQF und die nächsten zehn Jahre beim SQ-Magazin! Ich freue mich, dass mit dem „SQ mag“ nun auch eine internationale, englischsprachige Version des SQ-Magazins erschienen ist. Gleichzeitig bin ich zuversichtlich, dass es dabei hilft, noch mehr Menschen als bisher die Relevanz der genannten Themen zu vermitteln. |



**KONTAKT
ZUM AUTOR**

r.van_megen@gimi.cologne

System-Qualität in der Gigabit-Gesellschaft

Ein Blick auf den ASQF in 20 Jahren

Prof. Dr.-Ing. Ina Schieferdecker

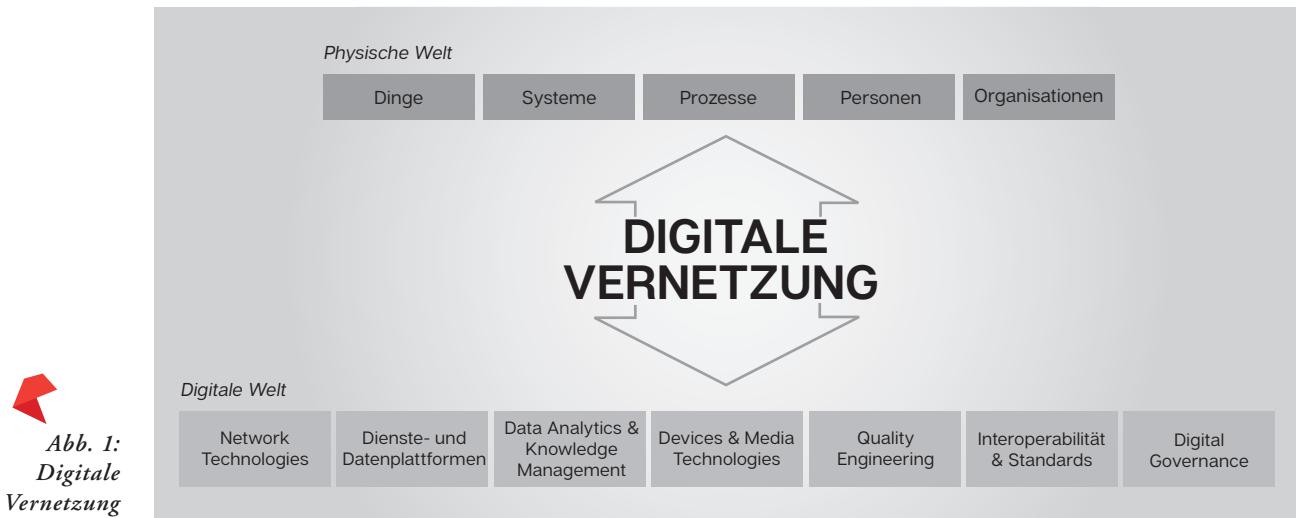
– ASQF-Präsidentin, Leiterin Fraunhofer-Institut für Offene Kommunikationssysteme –

Wie wir alle kann auch ich nicht in die Zukunft schauen – dennoch möchte ich mit meinen 20 Jahren Erfahrung in der System-Qualität zum 20. Jubiläum des ASQF einen Blick nach vorne werfen.

Lassen Sie uns mit den technologischen Veränderungen beginnen, die in den Augen vieler Menschen revolutionäre Veränderungen unserer Gesellschaft sind: Gebäude, Autos, Züge, Fabriken und die meisten Dinge unseres Alltags sind bereits oder werden in naher Zukunft mittels der überall verfügbaren digitalen Infrastruktur unserer zukünftigen Gigabit-Gesellschaft verbunden sein (siehe auch [Hauswirth 2016]). Dies wird den Informationsaustausch sowie die Kommunikation und Interaktion in allen Lebens- und Arbeitsbereichen verändern – sei es im Gesundheitswesen, in Verkehr, Handel oder Produktion. Für diese durch die digitale Vernetzung getriebene Technologie- und Domänen-Konvergenz, gibt es viele Begriffe: Internet of Things, Smart Cities,

Smart Grid, Smart Production, Industrie 4.0, Smart Buildings, Internet of Systems Engineering, Cyber-Physical Systems oder Internet of Everything. Trotz unterschiedlicher Zielrichtungen und Anwendungsbereiche liegt all diesen Begriffen als Basiskonzept ein allumfassender Austausch von Informationen zwischen technischen Systemen zu Grunde – eben die »Digitale Vernetzung«, siehe auch Abb. 1:

Mit Digitaler Vernetzung bezeichnen wir die durchgehende und durchgängige Verknüpfung der physischen Welt mit der digitalen Welt. Dazu gehören die digitale Erfassung, Abbildung und Modellierung der physischen Welt sowie die Vernetzung der daraus entstehenden Informationen. Diese ermöglicht die zeitnahe und teilautomatisierte Beobachtung, Auswertung und Steuerung der physischen Welt.





*„DIE KOMPLEXITÄT DER SYSTEME IST INZWISCHEN SO GROSS,
DASS ALLE WELT VERSUCHEN WIRD, FEHLER ZU VERMEIDEN
UND SO FRÜH WIE MÖGLICH ZU FINDEN.“*

Rudolf van Megen für 2035

Die digitale Vernetzung ermöglicht einen nahtlosen Informationsaustausch zwischen den digitalen Abbildern von Personen, Dingen, Systemen, Prozessen sowie Organisationen und baut ein weltweites Netz von Netzen – ein Inter-Net – auf, das weit über die Vision des ursprünglichen Internets hinausgeht. Bei dieser neuen Form der Vernetzung geht es aber nicht mehr nur um das Vernetzen an sich. Vielmehr werden einzelne Daten zu Informationen zusammengefasst, um weltweit vernetztes und vernetzbares Wissen aufzubauen.

Das „Netz der Netze“ wird verschiedene physikalische Kommunikationslösungen (drahtlos, drahtgebunden, für unsichere Umgebungen, schmalbandig, breitbandig, etc.) und gleichzeitig eine gemeinsame logische Ebene für die Vernetzung und Integration bieten. Dies wiederum ermöglicht vielfältige höherwertige Kommunikationslösungen, beispielsweise für industrielle Steueraufgaben, Vernetzung im Verkehr, 3D-Echtzeit-Visualisierung, Warndienste oder soziale Netzwerke. Darüber hinaus wird das „Netz der Netze“ Informationen aus unterschiedlichen, privaten, öffentlichen und kommerziellen Quellen und für unterschiedliche Anwendungen und Domänen bereitstellen müssen. Nur so können Informationen ungehindert vernetzt werden und Wissen erzeugen, das unter Umständen noch gar nicht bekannt oder erkannt ist. Dafür müssen Standards etabliert werden, die dafür sorgen, dass Informationen für zukünftige Entwicklungen flexibel kombiniert und effizient maschinell verarbeitet werden können. Zudem werden zukünftig ins Netz integrierte innovative Dienste die Verarbeitung von Daten bereits am Ort der Entstehung und im Netz ermöglichen, was eine fortschreitende Verschmelzung von Kommunikations- und Informationstechnik impliziert.

Mit Hinblick auf diese digitale Vernetzung nimmt die zentrale Rolle der Software weiter zu. Es werden nicht nur die digitalen Abbilder, also die Strukturen, Daten und Verhaltensmodelle der Dinge, Systeme und Prozesse der physischen Welt, mittels Software realisiert. Ebenso werden alle Algorithmen mit denen diese digitalen Abbilder visualisiert, interpretiert und weiterverarbeitet werden, als auch alle Funktionen und Dienste der Infrastrukturen und Systeme wie Server und (End-)Geräte im Netz der Netze umgesetzt. Während noch vor nicht allzu langer Zeit die wesentlichen Eigenschaften der Infrastrukturen und Systeme durch die Eigenschaften der Hardware definiert wurden und es im Wesentlichen um Software- und Hardware-Codesign ging, tritt die Hardware aufgrund generischer Hardwareplattformen und -komponenten in den Hintergrund und wird durch Software definiert oder gar aus Nutzersicht virtualisiert. Aktuelle technische Entwicklungen sind hierzu Software Defined Networks inklusive der Network Slices oder auch Cloud-Dienste wie Infrastructure as a Service oder Platform as a Service.

Des Weiteren beeinflussen diese software-basierten Systeme heutige kritische Infrastrukturen wie die Strom-, Wasser- oder Notfallversorgung als auch das zugrundeliegende Netz der Netze wesentlich: Sie sind integraler Bestandteil der Systeme, so dass die enthaltene bzw. genutzte Software als auch die Infrastrukturen selber zur sogenannten kritischen Infrastruktur werden. „Software-basiertes System“ nutze ich als Oberbegriff für derartige Systeme, die maßgeblich durch Software in ihrer Funktionalität, Leistungsfähigkeit, Sicherheit und Qualität bestimmt werden. Dazu gehören vernetzte und nicht vernetzte Steuerungssysteme, wie z. B. Steuergeräte in Automobilen, Flugzeugen, Systeme für das >>



*„VIELLEICHT (HOFFENTLICH!) IST ERSTMALS EIN MENSCH INS
GEFÄNGNIS GEKOMMEN, WEIL ER SCHLAMPIGE SOFTWARE
GEMACHT ODER ZUGELASSEN HAT.“*

Prof. Jochen Ludewig für 2010

vernetzte und autonome Fahren und Systeme von Systemen, wie z.B. die Verlängerung des Automobils in die Backbone-Infrastruktur der OEMs. Aber auch Systeme (von Systemen) in Telekommunikationsnetzen, der IT, Industrieautomatisierung und Medizintechnik werden darunter verstanden.

Software-basierte Systeme sind heutzutage oftmals verteilt und vernetzt, unterliegen Echtzeitanforderungen (weichen bzw. harten), sind offen über ihre Schnittstellen in die Umgebung eingebunden, stehen mit anderen software-basierten Systemen in Interaktion und nutzen lernende oder autonome Funktionalitäten zur Beherrschung der Komplexität. Unabhängig davon, ob wir uns mit der Digitalisierung nun in einer vierten Revolution oder in der zweiten Welle der dritten Revolution befinden: Die fortschreitende Konvergenz von Technologien und die Integration von Systemen und Prozessen wird über Software vermittelt und getragen. Auch neue Entwicklungen wie für Augmented Reality, Fabbing, Robotik, Datenanalytik und Künstliche Intelligenz stellen zunehmende Anforderungen an die Zuverlässigkeit und Sicherheit software-basierter Systeme. Aspekte der Ethik sowie der Gewährleistung und Haftung ergänzen die Dimensionen der Software-Qualität [ISO 25010] zu funktionaler Angemessenheit, Effizienz, Zuverlässigkeit, Nutzbarkeit, Sicherheit, Kompatibilität, Wartbarkeit und Portierbarkeit. Wir müssen über die Kritikalität software-basierter Systeme sprechen (siehe auch [Schieferdecker 2016]).

Offenheit, Vernetzung, Allgegenwärtigkeit, Autonomie und Sicherheit dieser software-basierten Systeme – kurz die Kritikalität – definieren originäre Aufträge an den ASQF als größten software-orientierten Fachverband im deutschsprachigen Raum: Wir müssen das Verständnis der und das Wissen über die Kritikalität von software-basierten Systemen stärken, Aus- und Weiterbildungsangebote entwickeln sowie Platz für den Informationsaus-

tausch und die Diskussion schaffen. Zudem müssen auch immer neue Konzepte, Methoden, Technologien und Werkzeuge erarbeitet werden, die mit den technischen Entwicklungen in den verschiedenen Anwendungsbereichen Schritt halten und die dortigen software-basierten Systeme effektiv und effizient abzusichern helfen. Aber mehr noch: Uns muss der Sprung aus der Entwicklungsbegleitung in die Betriebsbegleitung und -absicherung gelingen. Aufgrund der sich permanent verändernden Umgebungen der software-basierten Systeme als auch ihrer eigenen Auslegung (beispielsweise durch Software-Updates oder neue Konfigurationen) müssen wir die Systeme auch während des Betriebs absichern. Dafür reicht es nicht, dass wir die Betriebsprozesse überprüfen und Personal ausbilden – die software-basierten Systeme müssen auch technisch im Betrieb überprüft werden. Das erfordert neuartige Methoden zur technischen Überprüfung in der Produktionsumgebung, gegebenenfalls auch während des laufenden Betriebs, da ja software-basierte Systeme oftmals keine Ruhezeiten kennen. Aus heutiger Sicht ist das wohl auch der einzige Zugang zur Absicherung autonomer Systeme, da sich deren Funktionalitäten und Eigenschaften erst im laufenden Betrieb ausprägen.

So geht der ASQF das topaktuelle Thema der Qualität von software-basierten Systemen im Internet der Dinge an. Auf Einladung des ASQF und in Kooperation mit dem GTB und Fraunhofer FOKUS erarbeiten namhafte Treiber und Experten der Digitalisierung in der Industrie ein neues Ausbildungsschema für IoT. Wichtig ist der Gruppe „Quality Engineering für das Internet der Dinge“ (kurz IoT-QE) nicht die reine Validierung „am Ende“, sondern die vorausschauende Qualitätssicherung des Internets der Dinge von den ersten Entwicklungsschritten an. Dabei spielt beispielsweise die Priorisierung der relevanten Qualitätskriterien einer IoT-Lösung eine entscheidende Rolle. Die Zusammensetzung der Arbeitsgruppe –

u.a. mit Experten der DB Systel GmbH, der SAP Deutschland SE & Co. KG, der Atos IT Solutions and Services GmbH, der Sulzer GmbH, der imbus AG und der tecmata GmbH – soll sicherstellen, dass das Thema aus allen relevanten Blickwinkeln betrachtet wird: Geschäftsprozesse, Systementwicklung, Absicherung, Betrieb als auch Forschung und Entwicklung, welche durch IoT geprägt sind. Dadurch wird ein Schema entstehen, welches einen Foundation Level Einstieg ermöglicht, d.h. den Überblick und die Kenntnisse über die relevanten Aspekte des Themas Quality Engineering für das Internet der Dinge vermittelt.

2005 fragte Prof. Andreas Spillner die Software-Testing-Community nach der Zukunft des Prüfens und Testens in fünf (also 2010), 15 (also 2020) und 30 Jahren (also 2035), siehe [Spillner 2006]. Aus meiner heutigen Sicht waren darunter interessante Aussagen, beispielsweise:

- Prof. Jochen Ludewig für 2010: „Vielleicht (hoffentlich!) ist erstmals ein Mensch ins Gefängnis gekommen, weil er schlampige Software gemacht oder zugelassen hat. Aber wahrscheinlich geht auch das nicht so schnell.“

 *Das haben wir nach meinem Kenntnisstand noch nicht erlebt. Es mehren sich aber die Stimmen nach justizierbarer Software-Haftung, so dass wir hoffentlich nicht mehr nur finanzielle Auseinandersetzung wegen Software vor Gericht erleben werden.*

- Prof. Martin Glinz für 2010: „Wir werden uns an Software-Tote gewöhnen, wie wir uns an die Verkehrstoten gewöhnt haben. Einzelne, besonders verantwortungslose Entwickler werden zwar be-

straft werden (so wie im Straßenverkehr ein Raser, der Menschen auf dem Gewissen hat), aber allgemeine, durchgreifende (und wirksame!) Maßnahmen werden bei der Software ebenso wenig ergriffen werden, wie im Straßenverkehr, da zu aufwendig und zu unbequem.“

 *Zum Glück mussten wir uns noch nicht an Software-Tote gewöhnen, doch halte auch ich wegen der oben beschriebenen Kritikalität der software-basierten Systeme dieses Szenario für immer wahrscheinlicher. Gleichzeitig gehöre ich zu den Optimisten und glaube daran, dass wir geeignete Regularien, Methoden und Techniken entwickeln werden, um Verantwortungslosigkeit auch bei software-basierten Systemen zurückzudrängen.*

- Prof. Mario Winter für 2020: „Es gibt ein weltweit anerkanntes Curriculum in Sachen Qualitäts-sicherung (QS), und viele der ehemals analytischen Verfahren sind nun bereits konstruktiv berücksichtigt und somit aus dem QS-Kanon ‚verschwunden‘. QS betrachtet zunehmend die ‚polisch-sozialen‘ Auswirkungen und Möglichkeiten der Systeme.“

 *Das haben wir zumindest für das Software-Testen geschafft, auch wenn zur Qualitätssicherung software-basierter Systeme noch so viel mehr zu sagen und zu vermitteln ist. Hier werden wir in den Anstrengungen zu Ausbildungsmodulen nicht nachlassen. Und in der Tat bewahren uns schon heute verschiedene Automatismen und vorgefertigte Komponenten in der Entwicklung software-basierter Systeme vor prinzipiellen Fehlern, jedoch bringt meines Erachtens jede neue Systemtechnologie auch neue Schwierigkeiten und Risiken mit sich, so dass der QS-Kanon eher weiter anwachsen wird und muss. >>*

- Rudolf van Megen für 2035: „Die Komplexität der Systeme ist inzwischen so groß, dass alle Welt versuchen wird, Fehler zu vermeiden und so früh wie möglich zu finden.“

 *Dies befolgen Entwickler schon heute, doch erübrigt sich dadurch nicht der Bedarf an analytischer Qualitätssicherung.*
- Prof. Bernd Hindel für 2035 (nicht unbedingt ernst zu nehmen): „Es gibt nur noch selbst heilende SW.“

 *In der Tat ist das ein mögliches Szenario für autonome Systeme, jedoch sehe ich das als zu erstrebende Vision, die wir wohl nicht erreichen werden – eben wie wir keine 100-prozentig sicheren software-basierten Systeme entwickeln werden können.*
- Und meine Meinung für 2035: „Wir haben endlich den SSTV (Software- und System-Test Verein), der unabhängig, systematisch und automatisiert Software und software-basierte Systeme welcher Art auch immer – prüft, Gütesiegel vergibt – und darüber Qualitätsstandards durchsetzt.“

 *Ich glaube noch immer, dass wir einen solchen Verein benötigen (und er sich wohl nicht aus den etablierten technischen Prüfvereinen heraus ent-*

wickeln kann, da software-basierte Systeme eine Klasse für sich darstellen). Jedoch sehe ich mittlerweile mehr Varianz in der Auslegung von Gütesiegeln – bisherige sind zu statisch und arbeiten im Wesentlichen mit Snapshots der software-basierten Systeme, was nicht deren Offenheit, Flexibilität, etc. adressieren kann. Und gleichsam hoffe ich, dass wir diesen Verein noch vor 2035 aus unserer Community heraus etablieren und er sich dann 2036 bereits erfolgreich etabliert hat und für eine umfassende Absicherung kritischer software-basierter Systeme sorgt.

Zusammenfassend

Es stehen spannende Herausforderungen zu software-basierten Systemen in der digitalen Transformation an, die gleichsam unsere gesellschaftliche Verantwortung einfordern. Als ASQF werden wir uns diesen Herausforderungen stellen und unseren Blick von der reinen Software auf software-basierte Systeme erweitern. Seien Sie bei diesen Gestaltungs- und Absicherungsprozessen dabei! |



KONTAKT ZUR AUTORIN

ina.schieferdecker@asqf.de

Referenzen:

- [Hauswirth 2016] M. Hauswirth, I. Schieferdecker, et al, Fraunhofer FOKUS: Netzinfrastrukturen für die Gigabitgesellschaft, Studie für das BMVI, Nov. 2016.
- [ISO 25010] ISO 25010: Qualitätsmodell für Software-Produkte, siehe <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>.
- [Schieferdecker 2016] I. Schieferdecker: Qualität software-basierter Systeme im Fokus, Beitrag im Online-Issue Testing, Sigs Datacom, Okt. 2016.
- [Spillner 2006] A. Spillner: Antworten auf meine kleine Umfrage zur Zukunft des Prüfens und Testens im Jahr 2010, 2020 und 2035; Auszüge in A. Spillner: Von der Motte zur Roboter-Revolte Geschichte, Gegenwart und Zukunft des Prüfens und Testens von Softwaresysteme, 11.Juli 2006, GI-Regionalgruppe Bremen/Oldenburg.



**AUS DER
GESCHICHTE
DES ASQF**

Software-Qualität als entscheidendes Erfolgskriterium

Dr. Georg Heidenreich

– Manager IT-Standards, Siemens Healthcare GmbH –

Was macht Ihr denn da so? So oder ähnlich lauten die nüchternen Fragen von Vorgesetzten, wenn ich mal wieder Arbeitszeit oder gar Reisemittel für eine Veranstaltung des ASQF e.V. beantrage. Eben noch begeistert von neuen Methoden und Werkzeugen der Software-Technik, werde ich durch solche Fragen immer wieder auf die materiellen Realitäten der Software-Entwicklung zurückgeworfen: Je nach Intonation und Kontext enthält diese Frage ja mehr oder weniger hörbar die Botschaft: Da pas-

siert doch nichts, was dem Unternehmen (oder der Kostenstelle) nützen könnte!

In Wirklichkeit sind der ASQF e.V. und seine Mitglieder gemeinsam Zeugen eines enormen Aufschwungs, nicht nur von Technik und Methoden für Software-Qualität, sondern auch von Bewusstsein und Anerkennung für diese Methoden. Dabei muss man auch feststellen, dass ohne neue Methoden für Software-Qualität die Komplexität und Größe heutiger Software-Produkte und -Services überhaupt nicht realisierbar wäre. [>>](#)





Glückwunsch ASQF

„SEIT EINIGEN JAHREN WEISS ICH DEN ASQF ZU SCHÄTZEN:
BEI DEN REGIONALTREFFEN LERNT MAN IMMER WIEDER NETTE UND
OFFENE MENSCHEN KENNEN. DER ASQF ORGANISIERT UND BEWIRBT
DIESE TREFFEN HERVORRAGEND! DANKE, LIEBER ASQF UND AUF DIE
NÄCHSTEN 20 JAHRE!“

Dr. Andreas Reuys, CGI Deutschland Ltd & Co. KG,
ASQF-Fachgruppenleiter Software-Test, NRW

Gehen wir einmal zurück: In den frühen 1990er Jahren war Software-Qualität im Bewusstsein vieler hauptsächlich mit Testen verbunden. Der zugehörige Bereich in typischen Software-Unternehmen wurde Qualitätssicherung (QS) genannt und mancher QS-Beauftragte hatte die undankbare Aufgabe, am Ende einer Kausalkette (die man nicht als Entwicklungsprozess bezeichnen kann) durch Testen und die dabei induzierten „findings“ die Software zu verbessern. Die akademische Forschung beschäftigte sich damals mit Testmethoden und leitete Vorhersagen über Software-Qualität aus Teststatistiken ab.

Aus technischen Prozessbeschreibungen im Bereich der NATO entstand die Norm ISO 9000, die 1994 publiziert wurde und in den folgenden Jahren im Markt angenommen wurde – übrigens zunächst ohne ein entsprechendes Echo im universitären Umfeld. Für die damalige Praxis der Software-Entwicklung war der starke Fokus der ISO 9000 auf Prävention durch Prozesse eine bedeutsame Umstellung, denn bis dahin dominierte die Idee des gründlichen Testens als Evidenz für Korrektheit.

Die große Stärke der Prozessnorm ISO 9000 lag auch darin, nun auch nicht-funktionale Merkmale anstatt nur die (funktionale) Korrektheit zu fördern. Erst seit der ISO 9000 bedeutet Software-Qualität auch, die nicht-funktionalen Software-Merkmale systematisch optimieren zu können.

Dabei war die Anwendung der Norm auf die Software-Entwicklung keineswegs klar geregelt, so dass Erwartungen und Ergebnisse zunächst mal weit auseinanderklafften: Hersteller warben pauschal mit dem Hinweis auf ISO 9000, ohne dass Kunden bestimmte spezifische Maßnahmen hinsichtlich Software-Qualität voraussetzen konnten. Die Aussicht für Software-Unternehmen, mit ISO-9000-Zertifizierungen mehr Aufträge zu bekommen, war jedoch sehr verlockend und so ließ man sich von einer schnell wachsenden Beraterschar gerne in die neue Sichtweise der Prozesse einführen, um dann eben so ein Zertifikat zu ergattern. Fairerweise muss man auch aus Anwendersicht feststellen, dass solche Zertifikate oft das einzige Mittel waren, die Professionalität einer Software-Entwicklung zu beurteilen, und so wuchs mit dem Software-Markt auch der Bedarf nach Schulung und Zertifizierung.

Jedenfalls war die Prozessorientierung „en vogue“ und der Begriff QS wurde mehr und mehr von Qualitätsmanagement (QM) abgelöst. Testen war nur ein Teil dieser QM-Welt und verschwand für eine Zeit aus dem Bewusstsein der Führungsebene. Für die Region Mittelfranken bleibt festzuhalten, dass die Qualität von kommerzieller Software zusammen mit der Effizienz der dabei eingesetzten Entwicklungsprozesse sich in dieser Zeit enorm verbessert hat. >>

Besonders kleine und mittelständische Unternehmen erkannten die Chancen und so bildete sich ein zunächst informelles Netzwerk von Experten und Organisationen, das man durchaus als Vorläufer des ASQF e.V. betrachten kann.

Eine „Kleinigkeit“ in den Normen ISO 9000/1/2/3 darf da nicht verschwiegen werden:

Der Norm-Text an sich ist derart generisch, dass er gar keine Aussagen über konkrete Methoden und technische Verfahren für die Software-Entwicklung macht!

Dieses „Loch“ musste natürlich durch konkrete Auslegungen für die Software-Entwicklung gefüllt werden, und so entstanden viele branchenspezifische oder programmiersprachenspezifische Standards für Software-Prozesse und auch für Metriken, die zunächst qualitätsrelevante Merkmale von Software systematisch und wiederholbar bewerten konnten.

Auf diese Weise konnten einige innovative Anbieter komplett Software-Prozesspakte für umfangreiches QM anbieten und bei deren Implementierung beraten. Dabei wurde es für Software-Unternehmen zunehmend wichtig, die Erfüllung dieser Prozess-Standards durch ihre Prozesse nicht nur mit JA oder NEIN zu beantworten, sondern eine Graduierung attestiert zu bekommen.

In dieser Situation entstanden Metriken für Prozesse und Organisationen – ein Reifegrad gewissermaßen. Dieser Reifegrad – der durch eine Kombination der Erfüllungsgrade von Teilanforderungen errechnet wird – hilft den Unternehmen, ihre Bemühungen bei der Prozessverbesserung gezielt zu steuern und die Erfolge dabei zu messen.

Nach und nach gingen deshalb die Ergebnisse der anfänglich noch akademischen Software-Prozess-Bewertung und -Verbesserung in die kommerzielle Software-Welt ein: Das Capability Maturity Model wurde die Basis für prominente Ansätze, wie etwa CMMI und SPICE®, die auch heute noch anerkannt und etabliert sind.

Die Software-Unternehmungen verfügten somit über eine systematische und reproduzierbare Maßangabe, wie weit sie noch vom ersehnten Zertifikat entfernt sind.

In dieser Situation wollte die europäische Kommission auch die Regionen Europas fördern und gründete die European Software Process Improvement Training Initiative (ESPITI), die als Förderziel unter anderem die Gründung von regionalen Kompetenznetzen hatte. Der ASQF ging 1996 aus solch einem ESPITI-Projekt hervor, als institutionalisiertes und dauerhaftes Kompetenznetzwerk im Raum Mittelfranken.



Glückwunsch ASQF

„VOR 20 JAHREN IN FRANKEN GESTARTET UND NUN BUNDESWEIT
„TAKTGEBER“ – DANKE, LIEBER ASQF!“

“Erwin Probst, telexiom AG, ASQF-Fachgruppenleiter,
Mobile Devices & Apps, Rhein-Main

Die Vielzahl interessanter Beiträge im ASQF e.V. führte sehr bald zu themenspezifischen Fachgruppen. Der Wunsch nach einer besseren Verbreitung dieser geballten Expertise nach außen führte dann folgerichtig zur eigenen jährlichen ASQF-Konferenz, die sich unter dem Namen CONQUEST schnell etabliert hatte.

In den speziellen thematischen Fachgruppen des ASQF treffen sich seitdem Experten verschiedener Branchen zum fachlichen Austausch, zu heute 16 Themen, von denen ich zwei hervorheben möchte: Zum einen wurden die Techniken des Requirements Engineering methodisch weiterentwickelt

und können heute durch Verfahren sowie durch Werkzeuge unterstützt werden.

Zum anderen brachte die Software-Modellierung drei große Ansätze hervor (Use-Cases, Booch-Notation und Object-Modeling Technique OMT) deren Protagonisten schließlich als drei Amigos feierlich die Unified Modeling Language (UML) präsentierten, die heute noch in der Object Management Group (OMG) weiterlebt – und eben auch in einer Fachgruppe des ASQF.

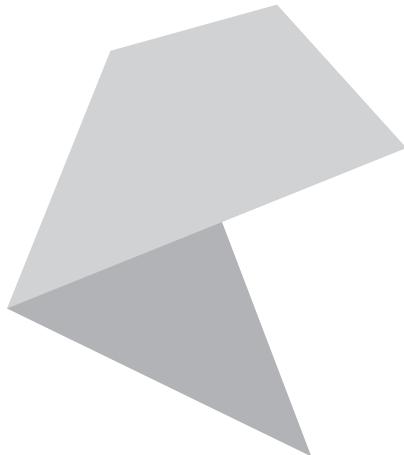
Beide Themen wurden recht früh als Aufgabengebiete der Software-Entwicklung mit starkem Bezug zur Software-Qualität erkannt. >>



Viele Ansätze im Bereich Software-Qualität werden heute durch Veranstaltungen des ASQF unterstützt. Man kann sie zur eigenen Weiterbildung nutzen, als Ideenschmiede für Neues, aber auch ganz profan als Marktplatz für Produkte und für Dienstleistungen rund um das Software-QM.

Insgesamt strahlt der ASQF e.V. seit zwanzig Jahren derart über Mittelfranken hinaus, dass sich dem erfolgreichen Muster entsprechend andere regionale Gesellschaften bildeten.

Die Fachgruppen und Konferenzen des ASQF e.V. sind die fachliche Antwort auf die eingangs gestellte Frage „Was macht Ihr denn da so?“. Es bleibt noch anzumerken, dass das konsequente Streben nach Software-Qualität auch manches Unternehmen in der Region Mittelfranken wirtschaftlich sehr gefördert hat – auch für große Unternehmen kann Software-Qualität ein entscheidendes Erfolgskriterium sein. Betriebswirtschaftlich zeigt sich Qualität deswegen nicht als Kostenfaktor des Produkts, sondern als Investition in eine Unternehmung! |



KONTAKT ZUM AUTOR

georg.heidenreich@siemens.com

Der ASQF – Ein Netzwerk aus Experten

27

Norbert Kastner

– ASQF-Vizepräsident –



Repräsentanz im Sinne von Software-Qualität



Qualitativ hochwertige Fachdiskussionen



Wissenstransfer durch Forschung und Praxis



Idealer Einstieg für Young Professionals



10 überregionale und 37 regionale Fachgruppen

Den ASQF lernt man am besten über seine Fachgruppen kennen, denn die Stärke dieses Netzwerkes liegt im persönlichen Austausch. Es wird von Experten getragen, die sich zu interessanten Themen austauschen, ihre Erfahrungen einbringen, Meinungen einholen, Fragen und Thesen diskutieren und bestrebt sind, an Lösungen zu arbeiten.

Hierzu engagieren sich die ASQF-Mitglieder in Fach- und Arbeitsgruppen. Sie dienen ihren Protagonisten, die diskutieren, lernen und, als Mehrwert, Best Practices sozusagen mit nach Hause nehmen können.

Die Fachgruppen – Herzstück des ASQF

Ein wesentlicher Beitrag der ASQF-Arbeit wird von den Fachgruppen geleistet. Das Außergewöhnliche daran ist, dass dieses Engagement von Mitgliedern ausgeht, die in ihrer jeweiligen Arbeitsumgebung regelmäßige regionale Treffen organisieren. Im Erfahrungsaustausch vermittelten anerkannte Referenten Know-how über Technologien und aktuelle

Trends. Darüber hinaus wird der persönliche Kontakt und Austausch der Teilnehmer untereinander gefördert. Eine Fachgruppe kann ein technologisches (z.B. Automatisierung), methodisches (z.B. Requirements Management) oder marktspezifisches (z.B. Automotive, Medizintechnik) Thema als Arbeitsgegenstand haben.

Jede Region beschäftigt sich aufgrund ihrer industriellen Struktur mit ganz individuellen Themen und besonderen Zukunftsherausforderungen. Diese werden von den Fachgruppen aufgegriffen. So werden beispielsweise aus der Medical Valley Metropolregion Nürnberg oder im Automotive Cluster Rhein-Main-Neckar relevante Inhalte aufgegriffen und Experten vor Ort zusammenführt.

Diese Art von Regionalität ermöglicht es vielen Menschen mit geringem Kosten- und Zeitaufwand die Verbindung zum Netzwerk zu halten und neben dem Joballtag, ihre Erfahrungen aus dem Berufsleben in Vorträgen und Diskussionen einzubringen. >>

Möglich wird das alles durch die inhaltliche Arbeit und das Engagement der Fachgruppenleiter, denen hier der besondere Dank gilt. Der ASQF unterstützt die organisatorische Abwicklung durch sein Back-Office und vielfach mit Hilfe von Partnerfirmen im Verein, die z.B. Räume für Treffen zur Verfügung stellen. Nicht selten ergeben sich durch diese Kooperationen weitere, konkret inhaltliche Arbeiten. So entstand bereits in der Anfangszeit des ASQF die Arbeitsgruppe für die Erstellung des Lehrplans zum Certified Tester Foundation Level, aus der sich heraus der GTB e.V. entwickelte.

Die Arbeitsgruppen – Aufgabenlösung in einer Hand

Die ASQF-Arbeitsgruppen übernehmen bestimmte Aufgaben innerhalb des Netzwerkes. Sie erarbeiten Inhalte und sind die Stimme des ASQF nach außen. Jeweils eine Arbeitsgruppe verfügt über definierte Aufgaben und legt sich damit auf ein zu erststellendes Arbeitsergebnis mit inhaltlichen und zeitlichen Zielen fest.

Ziele einer AG könnten sein:

- den Status quo eines spezifischen Themenbereichs festzustellen (z.B. mit einem Whitepaper),
- eine Prognose über zukünftige Entwicklungen herzuleiten,
- einen Lehrplan für die Weiterbildung in berufsrelevanten Themen zu erstellen oder zu überarbeiten.

Ein gutes Beispiel dafür ist die ASQF-Arbeitsgruppe CPPM, die in den vergangenen 18 Monaten einen komplett neuen Basis-Lehrplan, Prüfungsfragen und einen kompletten Foliensatz zum Thema Projektmanagement erstellt hat.

Neu gegründet wurde die Arbeitsgruppe Quality Engineering für das Internet der Dinge (IoT-QE), die sich in Form eines Lehrplans und Glossars den Anforderungen des Themas stellt, welches momentan in aller Munde ist. Hier gilt es, einheitliche Sprechweisen (Glossar) zu entwickeln, Sichtweisen auf eine gemeinsame Basis zu stellen und Quasi-Standards zu etablieren.

CPPM und IoT werden nicht das Ende der ASQF-Fachgruppen bleiben. Im Aufbau weiterer Arbeits- und Fachgruppen liegt ein wesentlicher Schwerpunkt der Arbeit des ASQF in den nächsten Jahren, immer mit dem Ziel, seinen Mitgliedern eine Stimme in der Branche zu geben. |



KONTAKT ZUM AUTOR

norbert.kastner@asqf.de



**QUALITÄT
IN PRAXIS UND
FORSCHUNG**

Der Weg ist das Ziel

Dr. Stefan Bunzel

– Principal Technical Expert Embedded Real-time Software Architectures
Continental - Corporate Systems & Technology –

„Software is the new oil“ oder „Software is the new wheel of the industry“ sind derzeit geflügelte Worte in der Automobilindustrie. Der Grund: Software ist „der“ Schlüssel zu vielen Innovationen und Verbesserungen. Fahrerassistenz- oder Sicherheitssysteme bis hin zum automatisierten Fahren, modernes Infotainment oder Cloud-basierte Dienste sind nur einige Beispiele. Seit den Anfängen der Software im Automobil ist der Trend ungebrochen, dass der Software-Umfang in neuen Fahrzeugen ständig wächst. Treiber hierfür sind vor allem neue und erweiterte Funktionen wie etwa in der Mensch-Maschine-Schnittstelle oder bei Fahrerassistenzsystemen im Fahrzeug.

Mit dem Umfang und den Funktionen wächst zugleich die Komplexität, weil Funktionen interagieren und Abhängigkeiten entstehen. Diese zunehmende Komplexität führt zu einem überproportional gestiegenen Qualitätsrisiko für die Software, dem die Entwicklung unbedingt Rechnung tragen muss. Weitere Herausforderungen resultieren aus dem steigendem Kostendruck, den verkürzten Entwicklungszeiten, den größeren Entwickler-Teams und einer stärker verteilten Entwicklung, sowohl örtlich – häufig international – als auch hinsichtlich der Zahl der beteiligten Partner. Daraus entsteht der Bedarf,

Software-Entwicklung kontinuierlich zu verbessern und in der Effizienz zu steigern. Da die Erwartungen und Ansprüche an Software-Qualität – insbesondere der Endkunden – trotz einem Mehr an Funktionen und höherer Komplexität nicht nachlassen, sondern eher noch steigen, müssen sich gerade die Entwicklungsbereiche, die Einfluss auf Software-Qualität haben, ständig weiterentwickeln.

Entsprechend der Bandbreite von Software-Entwicklung gibt es viele Ansatzpunkte für Verbesserungen: von der erforderlichen breiten Qualifikation der Mitarbeiter, leistungsfähigen Werkzeugen bis zu den qualitätsfähigen Methoden und Prozessen. In der Vergangenheit gab es in der Automobilindustrie noch allgemeinere Prozessverbesserungsmodelle wie CMMI und SPICE®. Mittlerweile hat sich die automotive spezifische Variante ASPICE durchgesetzt. ASPICE zielt auf grundlegende Aspekte der Entwicklung wie z.B. Anforderungen, Architektur, Design, Software-Erstellung, -Integration oder -Test ab. Weitere wichtige Themen sind die Qualitätssicherung, Konfigurations- und Änderungsmanagement und das gesamte Projektmanagement. Auch der Technologiekonzern Continental nutzt ASPICE seit langem und engagiert sich bei dessen Weiterentwicklung.



Glückwunsch ASQF

„FÜR DIE INFOTEAM SOFTWARE AG IST DER ASQF MIT SEINEM NETZWERK UND DEM FOKUS AUF PROFESSIONELLER SOFTWARE-ENTWICKLUNG EIN WICHTIGER FACHVERBAND. SOWOHL DIE FORTBILDUNGSPROGRAMME MIT PERSONALQUALIFIZIERUNG WIE AUCH DIE THEMENBEZOGENEN VERANSTALTUNGEN (Z.B. ASQF DAYS) SIND EIN FESTER BESTANDTEIL IN DER WEITERBILDUNG UNSERER MITARBEITER.“

Michael Sperber, infoteam Software AG, ASQF-Fachgruppenleiter Automatisierung, Franken

Unter dem Stichwort „Agile“ haben sich in den vergangenen Jahren eine ganze Reihe von Methoden und Praktiken in der automotive Software-Entwicklung verbreitet. Dabei sind Kanban oder SCRUM die wohl populärsten agilen Methoden, um die Arbeit von Teams zu organisieren bzw. zu koordinieren. Einerseits soll mit den Prinzipien der sogenannten „Lean“-Entwicklung, wie Warteschlangen-Optimierung, Vermeidung unnützer Puffer oder Begrenzung von Parallelaktivitäten, der Arbeitsdurchsatz gleichmäßiger, aber doch auf hohem Niveau, mit hoher Teammotivation und letztlich hoher Qualität gestaltet werden. Andererseits helfen gerade SCRUM und Kanban dabei, besser auf Änderungen

zu reagieren. Das können beispielsweise geänderte Anforderungen sein oder Randbedingungen, z.B. durch neue technische Erkenntnisse während der Entwicklung. Ein anderer Aspekt, oft auch als agile Praktik bezeichnet, betrifft die kontinuierliche Integration (engl. „Continuous Integration“). Diese wird – neben Übersetzen, Linken der Anwendungen – mit der kontinuierlichen Verifikation (z.B. durch automatisierte Tests) und Validierung und sogar mit der Auslieferung von Software gekoppelt. Die Continental bleibt bei Anwendung agiler Methoden aber nicht bei der Software-Entwicklung stehen. Wir sehen die volle Nutzung des Potentials, wenn die gesamte Entwicklung und der Vertrieb eingebunden werden. [>>](#)



Gute Ansatzpunkte für Verbesserungen liegen auch in den Details von Methoden, wie z.B. bei modellbasiertter Entwicklung oder der Anwendung von Kodierrichtlinien (z.B. MISRA) und Software-Metriken. Gerade Entwicklungswerkzeuge wie z.B. Test- und Designtools können nicht nur durch ihren Funktionsumfang, sondern durch ihre Benutzerfreundlichkeit großen Einfluss auf die Qualität der damit entwickelten Software haben.

Verbesserungen in den oben beschriebenen Bereichen müssen in Unternehmen im Entwicklungsalltag implementiert und gelebt werden. Dies erfordert weit mehr als nur Prozessdefinitionen. Das Wissen aus diesem Bereich muss breit auf die Entwickler verteilt sein und zudem ständig aktuell gehalten werden. Hierbei können standardisierte und zertifizierte Ausbildungsgänge, wie sie vom ASQF u.a. zu den Themenbereichen Anforderungsmanagement, Testen oder Projektmanagement gefördert und an-

geboten werden, das Vorhaben unterstützen. In größeren Firmen sind hierzu unter Umständen firmenspezifische Standards sinnvoll, aber die generellen Ausbildungen mit Zertifikaten wie vom IREB®, ISTQB® oder PMI® legen dafür den Grundstein. Um dieses Wissen effektiv weiterzuentwickeln, bedarf es außerdem eines Austausches über Firmen- und Branchengrenzen hinaus, die in den Hochschulen einbezogen werden müssen. Die regionalen Themengruppen im ASQF sind hier ein gutes Beispiel, wie dieser Austausch praktisch gelebt wird. Aus diesem Grund engagiert sich Continental seit Jahren im ASQF.

Das Bestreben nach Software-Qualität ist aus unserer Sicht eine Reise. Es gilt sich ständig zu verbessern und das entsprechende Wissen zu erweitern. Immer wieder werden Zwischenziele erreicht, aber die Reise geht weiter. In diesem Sinne ist bei der Software-Qualität auch der Weg das Ziel. |



KONTAKT ZUM AUTOR

stefan.bunzel@continental-corporation.com

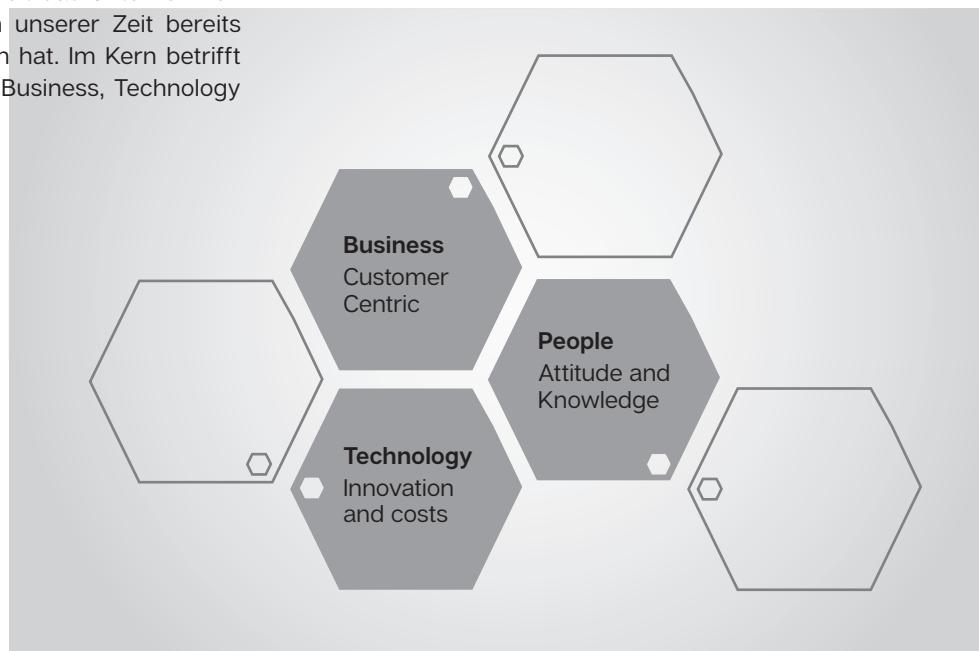
Digitale Transformation – Der Mensch, das Know-how und die Wünsche

33

José Díaz Delgado

– Díaz & Hilterscheid Unternehmensberatung GmbH –

Digital Destiny. Während die Digitalisierung anfänglich wie ein Spielfeld für besonders technikaffine Nerds anmutete, ist sie inzwischen zum entscheidenden Faktor des nackten Überlebens geworden. Alle Unternehmen sind verdammt, sich auf den Weg der Digitalisierung zu machen. Und wenn sie dann glauben, am Ziel angekommen zu sein, müssen sie feststellen, dass der Weg kein Ende nimmt. Manche müssen tiefe und weitgehende Veränderungen durchmachen, andere können mit wenigen Schritten ihre Position am Markt behaupten. Je nachdem, wie weit das Unternehmen die digitalen Veränderungen unserer Zeit bereits in sein Wesen aufgenommen hat. Im Kern betrifft dies folgende drei Bereiche: Business, Technology und People. >>



Zur Veranschaulichung erlauben Sie mir folgendes Beispiel: Bei einem von mir neulich beratenen Finanzdienstleister stellten wir folgenden Status quo fest:

- fehlende innovative FinTech-Produkte,
- fehlendes Konzept für die Entwicklung von Innovationen,
- disruptive Lösungen junger Fintechs verengen und überfordern sie,
- Kundenzugang wird durch Portale wie GOOGLE, IDEALO, CHECK24 etc. bestimmt,
- alte und kostenintensive Infrastruktur und Technologie (Großrechner, sehr große Server-Farm, viel Cobol, wenig moderne Programmiersprache und Methoden),
- Altersdurchschnitt der Arbeitnehmer >48,
- Schwierigkeiten, neue und junge Arbeitnehmer zu gewinnen („muffiges“ Image),
- notwendiges Know-how für die Veränderung vom CEO bis zum Pförtner fehlt,
- 2020 werden massive Probleme im Personalbestand auftreten, da viele Arbeitnehmer in Ruhestand gehen,
- fehlende Kapitaldeckung für Innovationen.

Wenn dieses Unternehmen nicht in allen der eingangs genannten drei Bereiche – Business, Technology und People – massiv investiert, wird es in vier bis sechs Jahren nicht mehr am Markt sein.

Aber was bedeutet das genau?



Glückwunsch ASQF

„ASQF-FACHGRUPPENTREFFEN ‘SOFTWARE-TEST’ IN ÖSTERREICH
SOWIE DIE ZEITSCHRIFT SQ-MAGAZIN HABEN EINEN WICHTIGEN
BEITRAG ZUM VERSTÄRKten AUSTAUSCH UND WISSENSTRANSFER
IN DER ÖSTERREICHISCHEN TEST-COMMUNITY GELEISTET.“

Dr. Mohsen Ekssir-Monfared, SQS Software Quality Systems Ges.mbH,
Fachgruppenleiter Software-Test, Österreich

Business

Das Geschäft eines Unternehmens dreht sich nicht mehr um das Produkt oder die Dienstleistung als solche, sondern wendet sich massiv den Wünschen des Kunden zu. Salopp formuliert heißt das, der Kunde steht neuerdings im Mittelpunkt und nicht mehr im Weg. Der Unternehmensgegenstand ist nicht mehr nur funktional, sondern bietet den Kunden auf seine individuellen Bedürfnisse zugeschnittenen Mehrwert. Diese neuen, disruptiven Produkte oder Dienstleistungen machen alteingesessenen Unternehmen das Leben schwer, manche kostet es das Überleben.

Ein Paradebeispiel hierfür ist die Finanzwirtschaft, in der junge Fintech-Unternehmen den traditionellen Banken und Versicherungen die Kundschaft wegnehmen. Diese großen und mittelgroßen Institu-

ten haben die Veränderungen zwar kommen sehen, aber im Vertrauen auf ihre Größe und Marktpräsenz wenig Anpassungen an die neuen Realitäten unternommen. Hinzu kam die Finanzkrise, die zusätzlich an der Substanz zehrte und etwaige Aufbruchsstimmung oder Wandlungswillen lähmte. Es braucht Zeit und massive Einschnitte, um Jahrzehntelang gewachsene Strukturen so radikal zu verändern, dass sie schnelle und situative Reaktionen ermöglichen. Viele Finanzdienstleister haben veraltete Systeme, viel zu viele und für die neuen Aufgaben nicht ausgebildete Arbeitnehmer, eine kleinteilige und sehr kostenintensive Infrastruktur, aber vor allem haben sie weder die Produkte, die die Kunden wünschen, noch die erwartete moderne Ansprache, geschweige denn den direkten Kundenzugang. >>



Der Fisch stinkt bekanntermaßen zuerst vom Kopf her. Wenn die Geschäftsführung und das Mittelmanagement nie ernsthaft gewillt sind, die notwendigen Veränderungen auch gegen Widrigkeiten und vermeintliche Hemmnisse herbeizuführen, wird es sie auch nicht geben. Irgendwann ist es dann tatsächlich zu spät. Das Wirtschaftsleben ist geprägt durch Entstehung und Niedergang von Ideen und Firmen, die sich dem Markt nicht angepasst haben. Das gilt natürlich nicht nur für alteingesessene Unternehmen, sondern genauso für Start-Ups, wenn sie sich langfristig etablieren wollen.

Im Gegensatz zur oft propagierten, da oftmals geradezu als erdrückend empfundenen, Hilflosigkeit den neuen Trends gegenüber, können Innovationen systematisch erzeugt werden. Sie können erarbeitet werden. Hierzu bedarf es den Mut, neue Vorgehensweisen wie zum Beispiel Design Thinking anzuwenden. Design Thinking stellt den Kunden als Menschen mit all seinen Bedürfnissen in den Mittelpunkt. Airbnb ist durch Design Thinking zu dem geworden, was es heute ist. Es ist höchste Zeit, sich auf dem Weg zu machen, und sich hier weiterzubilden. Citrix zum Beispiel hat jedem Arbeitnehmer einen Design-Thinking-Kurs „gegönnt“ - und mit „Jedem“ ist auch wirklich jeder gemeint, nicht nur Business Analysten und Mittelmanagement. Den Seufzer „Ach, wenn die Kunden nicht wären“ kann man offen ausgesprochen bei vielen Unternehmen noch heute hören. Hier ist die Neuorientierung, dass Mitarbeiter Kun-

den nicht mehr in ihrem Denken als notwendiges Übel, sondern als Mittelpunkt ihres gesamten Geschäftes sehen, oft schon die entscheidende Veränderung.

Technologie

Aus dem oberen Beispiel ist ersichtlich, dass mit alten Systemen und erdrückenden Kosten die digitale Transformation nicht machbar ist. Die Entwicklung von neuen, innovativen Produkten muss inkrementell erfolgen, um das von Kunden und Markterfordernissen vorgegebene Tempo halten zu können. Time To Market und Reaktionszeit sind nicht mehr Schlagworte und Wünsche, sondern der aktuelle Standard. Agile Methoden und die agile Kultur haben sich daher etabliert. Die Entwicklung erfolgt mehr und mehr Tool- und Methoden-basiert und immer dem neuen Produkt dienlich. Die technologische Innovation ist sehr groß. Oft fehlt an entscheidender Stelle das Know-how, das nicht schnell genug in den eigenen Reihen aufgebaut werden kann. Externe Kräfte können an dieser Stelle vorübergehend Abhilfe schaffen. Doch oft macht hier der zentrale Einkauf, der die technologische Entwicklung einer Firma „unbewusst“ stark beeinflusst, diese Möglichkeit zunichte. In vielen Firmen entsteht hier ein Bottleneck, der die Reaktionszeit und die Gestaltungsmöglichkeiten der digitalen Transition ungeheuer hemmt. Ein neues Konzept mit mehr Verantwortung für die bestellenden Teams selbst, ist zeitgerechter und dringend angeraten.



„DAS DIGITALE SCHICKSAL TRIFFT JEDEN
IN DER GESELLSCHAFT.“

José Díaz Delgado



Neben Methoden und Tools haben Hardware und dazugehörige Software sowie deren Anschaffung und Wartung einen entscheidenden Anteil am Gelingen der Digitalisierungsbemühungen. Zum Beispiel unterhält ein Großteil der Firmen Serverfarms, die nicht nur sehr teuer zu haben und vor allem zu halten sind, sondern auch nur schwer skaliert werden können. Ein Trend, der derzeit nicht anzuhalten ist, ist die Auslagerung dieser Server in die Cloud. In Deutschland gibt es schon einige Anbieter, die die rechtlichen Parameter und Sicherheitsbedürfnisse der Firmen erfüllen. Die Auslagerung ganzer Systeme zu Dritt-Anbietern

ist immer eine immense Anstrengung, aber kann dann auch der entscheidende Befreiungsschlag mit der dazugehörigen Kostensenkung sein.

People

Im Digitalisierungsprozess stellt der Mensch an zwei Stellen entscheidend die Weichen: Erstens in seiner Rolle als Kunde und Käufer – er tritt als der Treiber des Geschäfts auf und bestimmt das Geschehen. Zweitens in seinem Wirken als Arbeitskraft – er ist derjenige, der den Prozess hin zur Entstehung des Produktes steuert. Für beides hat sich die Arbeitsweise massiv >>

verändert. Die Unternehmen brauchen Kollegen, die beständige Neuerung und Anforderungsänderungen als Normalität und nicht als zu vermeidende Ausnahme von der Arbeitsroutine auffassen, die teamfähig sind, die kommunizieren können, die eine Fehlerkultur unterstützen und vor allem lernwillig sind und dauerhaft bleiben. Vieles kann man erlernen, das „Verhalten“ aber ist Resultat der inneren Einstellung. Es gibt Firmen, wie zum Beispiel OTTO, die dieses Konzept seit langem forcieren. Der Kulturwandel im Unternehmen ist entscheidend, um erfolgreich weiterzukommen und sich am Markt zu behaupten.

Ein solcher Kulturwandel erfordert Anstrengungen auf beiden Seiten – des Unternehmens wie auch seiner Angestellten. Das bedeutet auch, sich voneinander zu trennen, wenn die Vorstellungen der zukünftigen Arbeitsweise nicht zueinander passen. Der Mitarbeiter muss ehrlich erwägen, ob sein Arbeitsplatz auch in Zukunft mit den neuen Vorzeichen seinen Vorstellungen entspricht. Ob er dem Wandel entsprechend genügend gefördert und gefordert wird. Ob er sich weiterbilden kann.

Unternehmen, die Weiterbildung als Kostenfaktor und nicht als Investition sehen, aber von ihren Mitarbeitern uneingeschränkte Anpassung einseitig einfordern, werden auf lange Zeit doppelt verlieren. Zum einen verlieren sie motivierte Mitarbeiter, die bereit sind, Verantwortung zu tragen und schon über langjährige Erfahrung im Unternehmen verfügen. Zum anderen können sie kein Know-how aufbauen, das sie auf ihrem Weg in die digitale Zukunft so dringend benötigen. Die Wertschätzung eines Arbeitnehmers wird nicht zuletzt dadurch ausgedrückt, in dem das Unternehmen sich um seine

individuelle Zukunft und Weiterbildung als bedeutender Teil des Ganzen kümmert.

Andererseits müssen auch Unternehmen den Mut haben, Entscheidungen über Mitarbeiter zu treffen. Wenn der Mitarbeiter nicht zur Strategie passt, muss dieser geschult und weitergebildet werden. Erfüllt der Mitarbeiter nicht der Anforderungen, muss ihm dies fair und klar gesagt werden. Denn so ist seine Stelle dauerhaft nicht sicher und seine Lebensplanung kompromittiert. Schreckt ein Unternehmen vor dieser klaren Weichenstellung zurück, wird es nur kurze Zeit später die Konsequenzen umso heftiger spüren. Die alteingesessenen Unternehmen, die keine digitale DNA aufweisen, bekommen ein müffiges Image und haben massive Schwierigkeiten, neues, gut ausgebildetes Personal zu bekommen. Und der Bedarf steigt ständig, steht aber einer begrenzten Zahl an Software-Entwicklern in Deutschland gegenüber. Die Zuwanderung könnte das Problem ein wenig lindern, aber wird es nicht lösen. Vor allem, weil dies zuallererst ein hoch politisches Thema ist, das gelöst werden muss. Der Trend, der vor Jahren eingesetzt hat, wird sich deutlicher manifestieren: Near-Shoring wird ein wichtiges Standbein vieler Unternehmen sein. Das heißt, die Partner im Ausland müssen nicht nur technologisch gut sein, sondern auch „kulturell ähnlich“, damit eine schnelle Kommunikation auf allen Ebenen reibungslos stattfinden kann. Erst, wenn alle die gleiche Sprache sprechen und die gleichen sozialen Werte teilen, werden die Reibungsverluste soweit minimiert, dass durch die Arbeit in verteilten Teams wirkliche Wertschöpfung entsteht und nicht zusätzliche Hemmnisse aufgebaut werden.



Glückwunsch ASQF

„ICH BIN SEIT MEHREREN JAHREN AUF ASQF-VERANSTALTUNGEN AKTIV
UNTERWEGS UND SEIT ZWEI JAHREN AUCH STELLVERTRETENDE
FACHGRUPPENLEITERN FÜR DIE FACHGRUPPE SOFTWARE-TEST IN NIEDERSACHSEN.
DURCH DIE ASQF-VERANSTALTUNGEN HAT MAN EINE EXZELLENTE MÖGLICHKEIT,
ANDERE ERFAHRUNGEN, SICHTEN UND MENSCHEN AUS DER REGION ZU HÖREN
UND KENNENZULEREN. HIER TRIFFT MAN DIE FACHEXPERTEN! DIE BREDEX GMBH
FREUT SICH, MITGLIED ZU SEIN – AUF DIE NÄCHSTEN 20 JAHRE!“

Alexandra Schladebeck, BREDEX GmbH, stellvertretene
ASQF-Fachgruppenleiterin Software-Test, Niedersachsen

Zusammenfassend lässt sich sagen:

In allen Bereichen des Unternehmens bedarf es für das Gelingen der Digitalisierung klarer Linien, die von Unten nach Oben und von Oben nach Unten gleichermaßen verlaufen. Der Raum für solch eine technologische Innovation muss geschaffen werden – technisch, methodisch und personell. Die alten Strukturen können dies nicht leisten.

Das digitale Schicksal trifft jeden in der Gesellschaft. Egal, ob es der einzelne Mensch oder ein Unternehmen ist. Das digitale Schicksal bringt schnelle Veränderungen mit sich, die nicht ohne eine strategische Planung und gezielte, maßgeschneiderte Maßnahmen zu meistern sind. Der Mensch steht im Mittelpunkt und nicht im Wege. Der Mensch ist der Schlüssel zum Erfolg. Wenn der Mensch nicht das notwendige Know-how besitzt oder sich aneignen kann, wird das Vorhaben scheitern. Wenn die Innovation nicht den Menschen und seine Werte im Fokus hat, wird sie ebenfalls scheitern. Daher werden einige Unternehmen die digitale Transformation nicht schaffen, andere werden neu entstehen. Alle, die da sein werden, werden uns Menschen in ihren Mittelpunkt stellen und uns viele Wünsche erfüllen – wenn auch nicht alle. |



KONTAKT ZUM AUTOR

jose.diaz@diazhilterscheid.de

Anspruchsvolle Projekte erfolgreich führen

Illusion oder Wirklichkeit?

Katrin Greßer / Ewa Sadowicz

– EinfachStimmig, Nürnberg –

Kennen Sie die zentralen Erfolgsfaktoren im Projektmanagement? Ist es die Fachkompetenz, soziale Kompetenz, emotionale Intelligenz, eine hohe Führungskompetenz? Tatsächlich sind es die Soft Skills, die den Unterschied machen.

Führen im Projekt – eine Meisterschaft!

Dass Führen in Projekten hohe Anforderungen stellt, ist bekannt – zumindest allen, die bereits Projekte geleitet haben. Die Führungssituation ist sehr komplex. Sie stellt besondere Anforderungen

an den/die ProjektleiterIn. Er/Sie hat ohne „strukturelle Macht“ in Form der disziplinarischen Weisungsbefugnis, auch immer wieder Konflikte zwischen der Projekt- und Linienfunktion zu lösen.

Auf der einen Seite steuert der/die ProjektleiterIn das Projekt, oftmals mit knappen zeitlichen und personellen Ressourcen, wenig Routine und einem hohen Koordinierungsaufwand. Auf der anderen Seite erfordert es hohe kommunikative und soziale Kompetenzen beim Führen und Motivieren der ProjektmitarbeiterInnen, um die Qualität zu sichern.





Glückwunsch ASQF

„DER ASQF IST EIN VERLÄSSLICHER PARTNER FÜR DEN FACHLICHEN AUSTAUSCH IN DER TESTING-COMMUNITY. FALLS MAN EIN VORHABEN IN EINEM ÄHNLICHEN BEREICH HAT UND NICHT WEISS, WIE MAN ANFANGEN SOLL, HAT ASQF DIE BENÖTIGTE ERFAHRUNG, UM DEN START ZU ERMÖGLICHEN UND ZU ERLEICHTERN.“

Sergej Mudruk & Daniel Knott, Xing AG,
ASQF-Fachgruppenleiter Mobile Devices & Apps, Norddeutschland

Wer bin ich und wenn ja, wie viele?

Chef? Koordinator? Moderator? Präsentator? Verhandlungsführer? Konfliktmanager? Berater? Coach? Also eine Rollenvielfalt, die eine ganz besondere Kombination von Kompetenzen und dazugehörigen Soft Skills von Ihnen als ProjektleiterIn erfordert.

Wie führen Sie?

Autoritär, demokratisch, kooperativ, situativ oder agil? Der Begriff „Führen“ leitet sich ursprünglich aus dem Althochdeutschen ab und bedeutet „ins Fahren bringen“ – nicht ausbremsen, dressieren oder unterdrücken. Leiten Sie ein klassisch geführtes Projekt und müssen als ProjektleiterIn stark führen? Oder managen Sie ein iteratives, agiles Projekt, das auf eine hohe Selbstorganisation der ProjektmitarbeiterInnen setzt? Als ProjektleiterIn werden Sie sich vermutlich überwiegend im kooperativen und demokratischen Führungsstil wiederfinden. Es wird aber auch Phasen geben, zum Beispiel in Krisen oder bei Engpässen, wo der autoritäre Führungsstil in den Vordergrund rückt.

Flexibel & situativ

Der Ansatz der situativen Führung kann für Sie als ProjektleiterInn besonders hilfreich sein. Gerade in Projekten befinden sich die Teammitglieder

oftmals außerhalb ihrer Komfortzone. Sich immer wieder auf Neues einzustellen ist ständig gefordert. Das Modell der situativen Führung ist sehr flexibel, weil es die Reife einzelner MitarbeiterInnen nicht festlegt, sondern von Aufgabe zu Aufgabe unterscheidet. Es werden zwei Dimensionen betrachtet: das Wollen und das Können. Der Führungsstil passt sich situativ dem jeweiligen Entwicklungsstand des Mitarbeiters an – mehr aufgabenorientiert oder mehr mitarbeiterorientiert.

Der Ansatz der situativen Führung bezieht sich nicht generell auf Personen, sondern immer auf die jeweilige Aufgabe!

Führen heißt vor allem soziales Denken, Entscheidungsfähigkeit, Flexibilität und Vorstellungskraft. Sie brauchen Gefühl für Zeit und Moment, Vielseitigkeit und Entschlossenheit. Es bedeutet Impulse in eine bestimmte und gewollte Richtung und zu einem passenden Zeitpunkt zu initiieren – ganz wie beim Tanzen. Erfolgreiche Führung bedeutet einfach mehr Menschlichkeit. Diese menschliche Reife spielt eine fundamentale Rolle in der Arbeit im Team und ist für Führungsprozesse entscheidend.

„Führung erkennt man am Handeln, nicht an der Hierarchie.“ Nico Rose >>

Vorleben und Vorbild sein

Als ProjektmanagerIn wirken Sie immer, Sie leben vor – ob Sie das nun wollen oder nicht. Im Positiven wie im Negativen. Je nach Persönlichkeit tragen Sie Ihre Werte und Einstellungen, Ihre innere Haltung bewusst oder unbewusst nach außen. Das gilt für ProjektleiterInnen ebenso wie für klassische Führungskräfte.

Doch was sind eigentlich Vorbilder? In der Kindheit spielen sie eine große Rolle. Im Business-Kontext werden sie häufig unterschätzt. Vorbilder fördern die Entwicklung der Identität, sind Quelle des Ansporns und der Inspiration. Sie geben Orientierung, sind Impulsgeber und Wertevermittler. Wer ein Vorbild hat, der arbeitet durchdachter, effizienter und hat mehr Selbstbewusstsein.

Wie kann ich mich als ProjektleiterIn systematisch weiterentwickeln?

Eine der wichtigsten Kompetenzen ist die Reflexionsfähigkeit. Und diese kann trainiert werden.

Gesucht: Fingerspitzengefühl

Empathie ist der Schlüssel für ein erfolgreiches Miteinander im Projekt. MitarbeiterInnen folgen nur widerwillig einer/m empathielosen Chefin. Lernen Sie zunächst eigene Gefühle wertfrei zu betrachten. Beobachten Sie im Alltag, was auf der menschlichen Ebene wirklich passiert. Reflektieren Sie: Wie geht es mir gerade im Meeting? Was macht mir Spaß an dem Projekt? Was macht mich wütend oder ärgerlich? Was nehme ich gerade wahr – wie ist die Stimmung?

Zeigen Sie im Gespräch wirkliches Interesse an Ihrem Gegenüber. Hören Sie zu und vor allem hin. Wechseln Sie auch einmal die Perspektive und nehmen die Sichtweise Ihres Gegenübers ein. Oder fragen Sie sich, wie eine dritte Person über die Situation denken würde. Andere verstehen heißt übrigens nicht – es ihm oder ihr immer Recht zu machen.

Feel the Emotion

Emotionale Intelligenz ist die Fähigkeit, eigene Emotionen und die Emotionen anderer sowie die dahinterstehenden Werte wahrzunehmen und zu differenzieren. Sie glaubhaft anzuerkennen, auszudrücken und zielgerichtet mit ihnen umzugehen. Eine Beziehungspflege mit sich und anderen.

Spieglein, Spieglein an der Wand?

Das Handelsblatt hat 2014 Chefs befragt, wie sie sich selbst als Führungskraft einschätzen. 95 Prozent gaben als Antwort, dass sie glauben bei ihren MitarbeiterInnen eine gute und akzeptierte Führungskraft zu sein.

Die Gallup Studie von 2015 zeigt hier ein ganz anderes Bild! 69 Prozent gaben an, in ihrer beruflichen Laufbahn schon einmal eine schlechte Führungskraft gehabt zu haben.

Das Selbst- und Fremdbild ist oft sehr unterschiedlich. Deshalb holen Sie sich regelmäßig Feedback von Ihren MitarbeiterInnen. Stellen Sie offene Fragen: Was läuft gerade gut im Projekt? Wo drückt der Schuh? Welche Form der Unterstützung brauchen Sie? Was ist mir als Führungskraft für die Zusammenarbeit mit meinem Team wichtig? Beteiligen Sie Ihre MitarbeiterInnen: Welche Ideen und Vorschläge haben sie? Was ist aus Ihrer Sicht wichtig, das Projekt erfolgreich abzuschließen? „Deutschland führt“, eine Studie¹ über Führung der Information Factory zeigt ebenfalls eine deutliche Diskrepanz. Auf die Frage nach den Entwicklungspotenzialen antworten die MitarbeiterInnen und Führungskräfte verschieden. MitarbeiterInnen wünschen sich mehr „weiche Faktoren“ und Führungskräfte sehen „die Welt“ mehr aus der Unternehmersperspektive.

Übrigens: Dass man Führen lernen und trainieren kann und muss, glauben deutlich mehr MitarbeiterInnen als Chefs. >>

¹ Quelle der Studie: http://www.information-factory.com/fileadmin/user_upload/studien/Deutschland_fuehrt_Studie.pdf



Glückwunsch ASQF

*„BEREITS SEIT 20 JAHREN HAT DER ASQF WESENTLICH DAZU
BEIGETRAGEN, DIE VERFAHREN IM BEREICH QUALITÄTSSICHERUNG
ZU DEFINIEREN UND STETIG WEITERZUENTWICKELN.“*

Stephan Oswald, T-Systems Multimedia Solutions GmbH & Matthias Schneider,
T-Systems Multimedia Solutions GmbH, ASQF-Fachgruppenleiter Software-Test, Sachsen



„DER WEG ZUM SPITZENTEAM FÜHRT MANCHMAL
ÜBER HOLPRIGE WEGE..“

Katrin Greßer / Ewa Sadowicz

Wer nicht weiter weiß, bildet einen Arbeitskreis

Gemeinsam erfolgreich

Beziehen Sie Ihr Projektteam von Anfang an ein – das wirkt sich positiv auf die Motivation aller aus.

Die Menschen im Projekt

Diversity im Team ist manchmal anstrengend, hat dafür aber einen großen Nutzen. Bei der Zusammensetzung sollten Sie auf Vielfalt achten. Junge und erfahrene, weibliche und männliche Teammitglieder. Unterschiedliche Fähigkeiten, Expertisen und Kulturen bereichern ebenso die Teamzusammenarbeit.

Der/die richtige ProjektmitarbeiterIn am richtigen Platz im richtigen Projekt. Mischen Sie je nach Projekt und nutzen so unterschiedlichste Eigenschaften und Fähigkeiten für eine erfolgreiche Zusammenarbeit: MitarbeiterInnen mit einer hohen Präzision, Detailkenntnis, Disziplin, Analyse- und Abstraktionsvermögen mit MitarbeiterInnen die sich auszeichnen durch Spontanität, Ideenvielfalt, Durchhalte- und Durchsetzungsvermögen und Erfolgsstreben.

Denken Sie auch an Teammitglieder mit einer hohen Intuition, Werte- und Sinnorientierung, Kontaktfreude, Empathie, Menschlichkeit, soziales Empfinden.

Sorgen Sie auch dafür, dass sich die MitarbeiterInnen laufend weiter qualifizieren, um beim Leistungs niveau mehr Ausgeglichenheit zu erreichen.

Spirit on Tour

Projektspirit ist an einem starken Zusammenghörigkeitsgefühl und guten Diskussionen zu erkennen. Gegenseitige Unterstützung ist selbstverständlich und die Arbeitsergebnisse sind einfach spitze. Doch Vorsicht vor besonders stark ausgeprägtem Wir-Gefühl: Das Team ist so sehr auf Harmonie bedacht, dass es abweichende Meinungen und Zweifel ausgrenzt. Dadurch können Fehlentscheidungen und schlechte Arbeitsergebnisse entstehen. Achten Sie darauf, dass sich aus Ihrem Projektteam kein Kuschelteam entwickelt. Ermutigen Sie stattdessen immer wieder alle, auch in schwierigen Situationen Meinungen zu äußern, die nicht gruppenkonform sind.

Projektkultur fördern und wertschätzen

Bedingungslose Wertschätzung ist ein existenzielles Bedürfnis aller Menschen. Die im Projekt beteiligten Personen wertzuschätzen bedeutet, ihre Standpunkte, Werte und auch Gefühle ernst zu nehmen. Sich für sie zu interessieren und sie verstehen zu wollen, zeichnet eine Beziehung auf Augenhöhe aus. Auch dann, wenn man selbst anderer Meinung ist. Die Voraussetzung für diese offene Haltung ist ein reflektiertes eigenes Wertesystem und -verständnis, das anderen gegenüber entsprechend formuliert werden sollte.

Lassen Sie die Korken knallen!

Wenn die Teammitglieder gut zusammenarbeiten, erreichen sie ausgezeichnete Leistungen und begeistern ihr Umfeld: Chef, Geschäftsleitung, Kunden und KollegInnen. Das sollte sowohl ausgesprochen als auch hin und wieder gefeiert werden.

Anerkennung und qualifiziertes Lob fördern die Leistungsbereitschaft und die Eigenverantwortung!

Mit Performance zum Spitzenteam

Soziale und emotionale Kompetenzen sind der Schlüssel für eine gelingende Projektarbeit. Offen, direkt und zeitnah kommunizieren. Teamfähig sein, sich in die Lage des anderen versetzen und das Gegenüber verstehen. Eigene Standpunkte vertreten und auch bereit sein, diese bei Bedarf zu revidieren.

Aufmerksames Zuhören und unterschiedlichen Meinungen mit Wertschätzung begegnen sowie regelmäßiges Feedback und Reflexion.

Wer ein Projektteam führt, weiß meist viel von kleineren und größeren Turbulenzen zu berichten, die es zu meistern gibt. Wellen der Teamdynamik. Der Weg zum Spitzenteam führt manchmal über holprige Wege. Deshalb lassen Sie sich Zeit für den Teamentwicklungsprozess.

- Fokussieren Sie auf Lösungen: Sprechen Sie über Lösungen statt über Probleme!
- Bauen Sie auf Erfolgen auf: Wenn etwas gut funktioniert, tun Sie mehr davon!
- Beleuchten Sie die Ressourcen: Welche Kompetenzen und Fähigkeiten gibt es in Ihrem Team?
- Gewinnen Sie neue Sichtweisen: Also immer wieder den Fokus der Aufmerksamkeit verändern. [>>](#)



Alles ist Kommunikation

Im Dialog sein und bleiben

Wie können die Arbeitsabläufe im Team und an den Schnittstellen zu anderen Abteilungen verbessert werden? Diesen Punkt sollten Sie fest auf die Agenda für Ihre Projektmeetings nehmen. Was läuft gut? Was läuft noch nicht so gut in der Zusammenarbeit? Diese zwei Fragen bringen schnell die Themen auf den Tisch. Gemeinsam können Sie dann Lösungen erarbeiten. Alle können sich einbringen. Das motiviert und sorgt für eine gute Stimmung. Außerdem sollte geklärt werden, wie der Infofluss im Team läuft. Also, wer informiert wen unaufgefordert und vor allem wie, damit alle wichtigen Informationen reibungslos hin und her laufen.

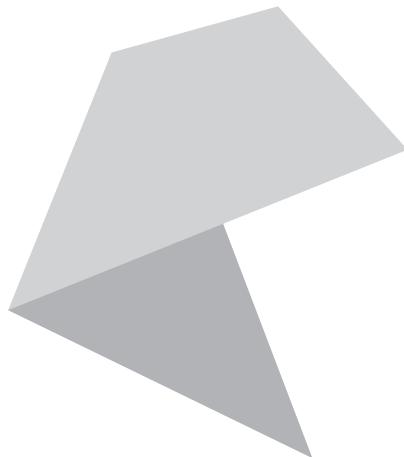
Es ist wichtig, dass es verschiedene Möglichkeiten für den gegenseitigen Austausch gibt. Die Klassiker

sind das Teammeeting oder der Jour-fix-Termin. Eine Alternative dazu ist das Stand-up-Meeting und Taskboard. Es findet im Stehen statt, dauert 5 – 15 Minuten. Der Ablauf ist klar, Small-Talk ist nicht erwünscht. Es geht darum, Hindernisse aus dem Weg zu räumen und demokratisch Entscheidungen zu treffen. Jedes Teammitglied beantwortet folgende drei Leitfragen: Was habe ich seit gestern geschafft? Was werde ich heute tun? Was hindert mich bei meiner Arbeit? Steht eine Entscheidung an, können mittels roter und grüner Klebepunkten Ideen bewertet werden. Entsteht weiterer Gesprächsbedarf so findet dieser bilateral statt.

„Wenn einer nicht mehr mit Dir redet, dann will er damit etwas sagen.“ Joachim Panten

It's up to you now

Denn wirklich gute Projekte sind einfach gut geführt – worauf warten Sie noch? |



KONTAKT ZU DEN AUTORINNEN

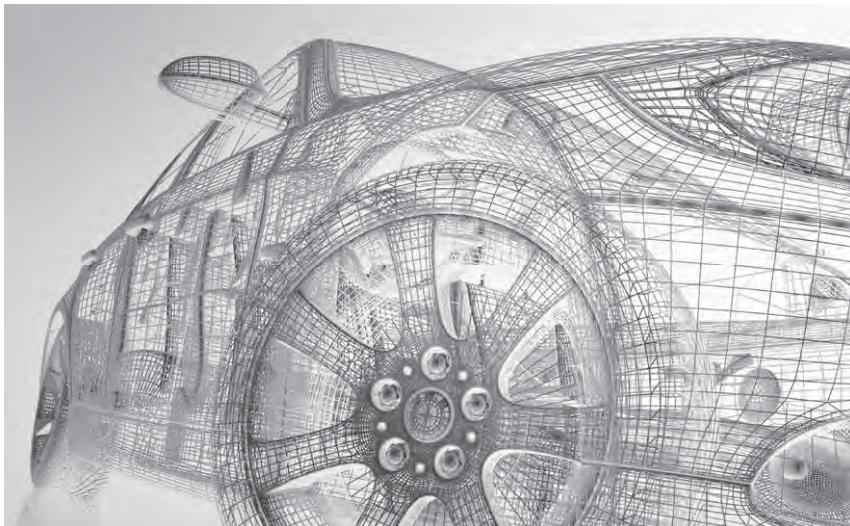
kg@einfachstimmig.de
es@einfachstimmig.de

POSMuT – Ein ganzheitliches Modell

zur Prozessverbesserung und Erzielung von
Automotive SPICE® 3.0-Konformität

Horst Pohlmann

– Leiter Prozesse, Methoden und Tools, Lemförder Electronic GmbH –



Ist mehr Qualität bei gleichzeitiger Steigerung von Effizienz und Effektivität möglich?

Mit Hilfe des vorgestellten POSMuT-Modells (Prozesse, Organisation, Skills, Methoden, Tools) kann die Komplexität von Entwicklungsprozessen besser beherrscht und gleichzeitig die Effektivität und Effizienz der Prozesse signifikant verbessert werden. Darüber hinaus kann die Migration und Umsetzung der Software-Prozesse in einer Organisation hinsichtlich Automotive SPICE® 3.0 -Konformität [VDA2015] auf Grundlage des POSMuT-Modells optimal gestaltet werden.

Einleitung

Bedingt durch die steigende Anzahl von elektronischen Systemen in Fahrzeugen und deren Vernet-

zung hat auch die Komplexität der elektronischen und elektrischen Systeme in der automobilen Entwicklung ständig zugenommen; d.h. „[i]nnerhalb von wenigen Jahren wurde eine neue Klasse von Fahrzeugen entwickelt, die alles in den Schatten stellt, was bisher in Sachen Elektronik und Software im Automobil realisiert wurde“ [JohM2015].

Mit Hilfe des gezeigten Modells kann die Komplexität von Entwicklungsprozessen besser beherrscht werden. Die Grundidee des Modells ist nicht neu, sie geht auf das Jahr 2008 zurück. Damals wurde beobachtet, dass die einseitige Ausrichtung auf die Optimierung der Prozesse im Ergebnis kontraproduktiv ist; sofern nicht – analog dem aus dem Projektmanagement bekannten Magischen Dreieck – andere Eckpfeiler wie Organisation und Qualifikationen (im Weiteren: Skills) mitbetrachtet werden. >>

Im Laufe der Jahre hat sich herausgestellt, dass gut entwickelte und optimierte Prozesse sowie Methoden und Tools ohne qualifizierte Mitarbeiter nutzlos sind. „Neben ‚reifen‘ Prozessen inklusive praxisorientierter Guidelines sind Methoden erst dann wirtschaftlich, wenn sie von Mitarbeitern und Führungskräften angewendet werden, die über die notwendigen Skills verfügen“ [LOCH2008].

Das vorliegende Modell stellt den Zusammenhang zwischen Prozessen, Organisation, Skills, Methoden

und Tools dar. In einem Unternehmen werden Prozesse definiert, die Organisation festgelegt, Methoden ausgewählt und Tools¹ beschafft. Um all dies zu steuern und umzusetzen, ist es wichtig, über Mitarbeiter in der Organisation zu verfügen, die die notwendigen Fähigkeiten und Fertigkeiten aufweisen. Geeignete Methoden werden mittels Werkzeugunterstützung durch qualifizierte Mitarbeiter ausgeführt. Eine festgelegte Organisation steuert und überwacht die Einhaltung von Entwicklungsprozessen, z.B. im Hinblick auf Automotive SPICE®.

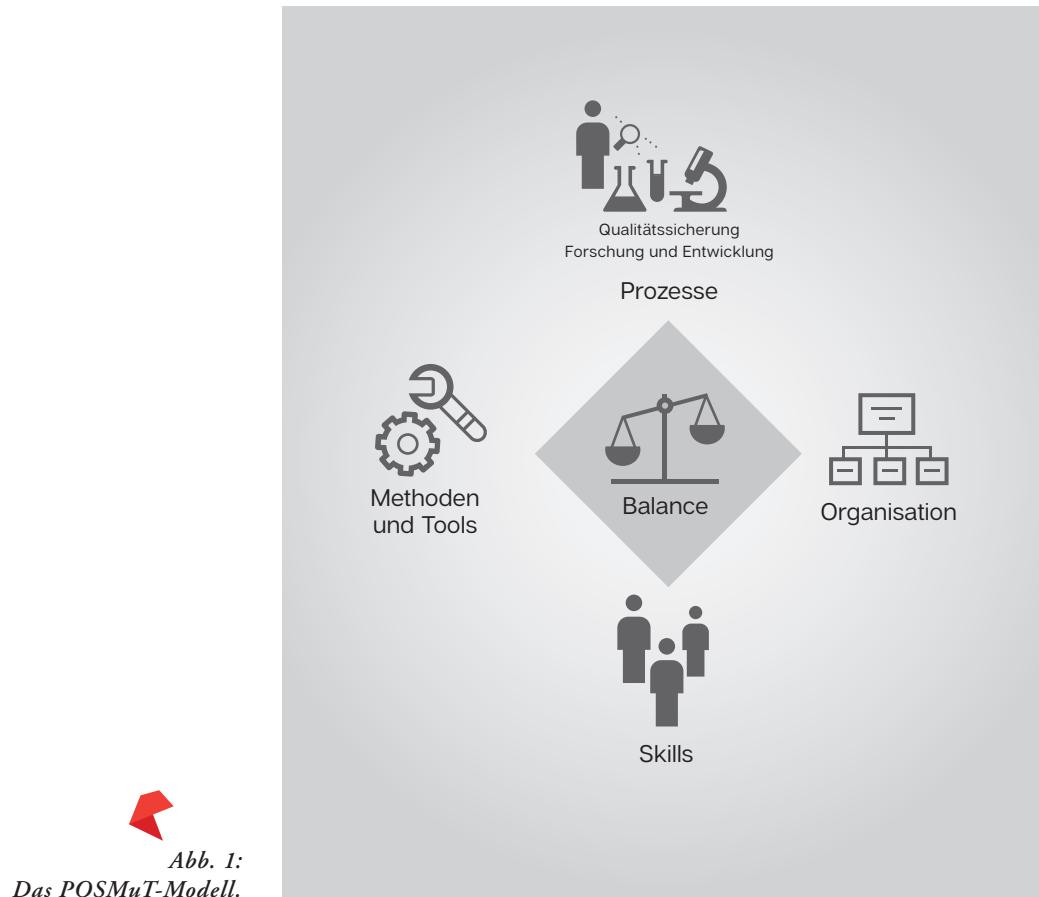


Abb. 1:
Das POSMuT-Modell.

¹ Tools steht für Werkzeugunterstützung.



„DAS GANZE IST MEHR ALS DIE SUMME SEINER TEILE.“

Aristoteles (384 v.Chr. - 322 v.Chr.)

Außerdem treibt sie den Einsatz neuer Methoden und Tools voran und sorgt dafür, dass Mitarbeiter dafür trainiert werden, die Methoden und Tools richtig anzuwenden. Alle diese Faktoren spielen beim Erreichen eines höheren Fähigkeitsgrades eine Rolle und können in einem sogenannten „Magischen Viereck“ veranschaulicht werden (Abb. 1). Dabei ist es wichtig, eine Balance zu finden, wobei jeder Teil des Modells ausgewogen entwickelt und umgesetzt werden muss. Einseitiges „Sich-Verlassen“ auf einzelne Aspekte, zum Beispiel Prozesse, führt nicht weiter (vgl. [LOCH2008]).

Die Erkenntnisse aus den Lessons Learned erlauben es dem Prozesseigner zukünftige Prozesse so zu gestalten, dass bereits „gemachte Fehler“ vermieden werden können. Gelernt wird nicht nur aus den Fehlern, sondern auch aus positiven Erfahrungen. Im Rahmen des Prozess-Monitorings wird ein spezifisches Change Control Board (CCB) aufgesetzt, also eine Gruppe von Mitarbeitern, die verantwortlich für die endgültigen Entscheidungen über Prozessänderungen ist. Ziel dieses speziellen Prozess-CCBs ist es, prozessbezogene Abweichungen, Anregungen, Optimierungen und Änderungen zu erfassen und zu verfolgen.

Von der Prozessentwicklung über das Prozess-Monitoring bis hin zur Prozessverbesserung

Prozesse: In einem Prozess werden definierte Aktivitäten und Schritte durchlaufen, die beispielsweise das Erstellen von Arbeitsprodukten beinhalten.

Prozessentwicklung: In der Prozessentwicklung werden die Prozessbeschreibung für einen Prozess erstellt, die Prozessaktivitäten und der Workflow festgelegt, die Arbeitsprodukte, die Rollen und Metriken (siehe auch [Hefke2015]) für einen Prozess definiert.

Prozess-Monitoring: Die Einhaltung der Prozesskonformität von einzelnen Prozessen und Prozesszielen wird durch das Prozess-Monitoring sichergestellt.

Prozessverbesserung: Definierte und priorisierte Maßnahmen aus Reviews sorgen dafür, dass unter anderem die Prozesse zielgerichtet angepasst bzw. weiterentwickelt werden. Ziel ist es, im Rahmen einer Prozessverbesserung, existierende Prozesse so zu optimieren, dass diese vorliegende Anforderungen an Prozesse aus unterschiedlichsten Quellen erfüllen. Dabei sollten die Mitarbeiter in die Prozessverbesserung miteinbezogen werden. Eine Mitwirkung bei diesen Optimierungsmaßnahmen gibt den Mitarbeitern die Möglichkeit, sich mit den Prozessen zu identifizieren, gesammelte Erfahrungen umzusetzen und Prozesse in Unternehmen zu beleben. Ein wesentliches Instrument der Prozessverbesserung ist, dass die Templates, Prozesse und Metriken usw. unternehmensspezifisch standardisiert werden, um für alle Folgeprojekte wiederverwendbar zu sein (siehe auch [Hefke2015]). [>>](#)

Methoden und Tools

Methoden sind definierte Arbeitsschritte zum Erreichen der festgelegten Ziele. Die Durchführung der Methoden wird mittels einer maßgeschneiderten Werkzeugunterstützung effizient und wirksam gestaltet. Für die Absicherung von Arbeitsprodukten und Prozessen nach Automotive SPICE® wurden Reviews, also die Überprüfung und Beurteilung, als essentielle Methode im Qualitätssicherungsprozess etabliert. Sowohl die Planung als auch die Durchführung von Reviews wird durch ein spezielles datenbankgestütztes Werkzeug effizient und effektiv umgesetzt.

Der kreative Anteil besteht darin, die Wirksamkeit der Maßnahmen aus dem Review sicherzustellen. Wirksame Maßnahmen sind solche Maßnahmen, die dazu beitragen, die Prozesse im Unternehmen grundsätzlich zu verbessern und nicht nur akute Symptome zu behandeln. Deshalb soll bei der Festlegung von Maßnahmen nach den Ursachen („root causes“) für Prozessabweichungen gesucht werden. Die Maßnahmen sind so zu definieren, dass sie effektiv und effizient bearbeitet werden können. Um also eine Wirksamkeit bei der Umsetzung von Maßnahmen zu erreichen, müssen die Abweichungen z.B. aus Prozess-Reviews und Assessments, also kritischen Bewertungen, analysiert, Ursachen identifiziert und nachhaltige Maßnahmen festgelegt werden. Letztere sind dabei konsequent (werkzeuggestützt) zu verfolgen, um Abweichungen wirksam

zu eliminieren. Gemäß VDA [VDA2008] sind Prozessverbesserungsmaßnahmen in (kurzfristige) Sofortmaßnahmen und (langfristige), nachhaltige Korrektur-/Vorbeugungsmaßnahmen zu gliedern. Die Werkzeugunterstützung dient dazu, den wiederholten manuellen Aufwand beispielsweise bei der Verfolgung von Maßnahmen zu reduzieren, und damit Effizienz und Effektivität der Prozesse zu fördern.

Skills

Skills bestehen aus Fähigkeiten und Fertigkeiten. Fähigkeiten sind Kompetenzen, um definierte Aufgaben zu bewältigen. Fertigkeiten sind Kenntnisse und gesammelte Erfahrungen, welche erlernt, erworben oder ausgebildet werden können. Infolgedessen sind Skills in unserem Kontext die erforderlichen Fähigkeiten und Fertigkeiten, die zur Ausfüllung einer Rolle im Projekt relevant sind.

In der Vergangenheit haben Mitarbeiter an einer Vielzahl von Trainings- und Schulungsmaßnahmen teilgenommen; wobei die Effektivität und damit Effizienz der Maßnahme nicht sichergestellt war. Um Effektivität und Effizienz bei diesen Trainings- und Schulungsmaßnahmen zu verbessern, wird auf folgenden Ansatz gesetzt: Mitarbeiter werden gezielt auf eine definierte Rolle (Soll) weiterentwickelt und damit mit den Skills ausgestattet, die sie für ihre jeweilige Rolle benötigen. Die generische projektübergreifend definierte Rolle wird dabei jeweils durch projektspezifische Anforderungen ergänzt.



Organisation

Eine Organisation ist eine Zusammenstellung von Menschen und Ressourcen, die auf geplante Art und Weise zusammenarbeiten, um definierte Ziele zu erreichen. Die Entwicklung und Umsetzung von Prozessen wird durch etablierte und definierte Prozesseigner der Organisation unterstützt. Für das Monitoring von Fortschritten und die Abstimmung von Verbesserungsmaßnahmen können unternehmensintern regelmäßige Key-User-Runden der Prozesseigner dienen. In diesen werden unter anderem Neuerungen von Normen und Standards (z.B. ASPICE 3.0) oder neue OEM-Anforderungen – wie beispielsweise die neue Ausgabe der Volkswagen-Konzern-Grundanforderungen-Software (KGAS) im

Überblick – vorgestellt, Lösungen aus umgesetzten Maßnahmen präsentiert, neue Vorschläge für eine effiziente Prozessverbesserung und deren effektive Umsetzung von Prozessen im Unternehmen besprochen. Eine bereits genannte, doch bedeutende Herausforderung für den Prozesseigner ist, die Prozesse so zu beschreiben, dass diese von den Mitarbeitern einfach zu verstehen und umzusetzen sind. Anmerkungen und Optimierungen hinsichtlich der Wirksamkeit der Maßnahmen für den Prozess sind vom Prozesseigner zu bewerten und in die Prozessentwicklung mit einzubeziehen. Durch den regelmäßigen Erfahrungsaustausch zwischen den Mitarbeitern und Prozesseignern werden die Anregungen und Empfehlungen zur Prozessverbesserung kommuniziert. [>>](#)

Zusammenfassung

Mit Hilfe des kreierten POSMuT-Modells kann die Komplexität der Entwicklungsprozesse besser beherrscht und gleichzeitig die Effektivität und Effizienz der Prozesse verbessert werden. Werkzeugunterstützung bietet den Mitarbeitern eine Hilfestellung bei der Umsetzung von wiederkehrenden Routineaufgaben. Die Automatisierung von manuellem Aufwand macht die Durchführung von manuellen Arbeitsschritten in den Prozessen effizienter und effektiver. Die Essenz in diesem Modell ist, dass eine Balance zwischen den vier Aspekten „Prozesse, Organisation, Skills sowie Methoden und Tools“ gehalten wird und analog Wechselwirkungen zwischen den „Aspekten“ wie bei einem magischen Viereck bestehen. Entscheidend ist die Erkenntnis, dass sich die Aspekte des Modells in ihrer „Abfolge“ komplementär ergänzen: vom unternehmensintern-standardisierten Prozess, über eine werkzeugunterstützte Zwischenlösung bis hin zu einer werkzeuggestützten datenbankbasierten Lösung, wobei adäquate Skills und qualifizierte Mitarbeiter ein wesentlicher Erfolgsfaktor auf dem Weg zu einer höheren Fähigkeitsstufe sind. |



KONTAKT ZUM AUTOR

Horst.Pohlmann@zf.com

Referenzen

- [Hefke2015] Hefke, Max; Pohlmann, H. : Best Practices zur Standardisierung von Testmetriken als Teilaспект zur Erreichung von Automotive SPICE® Reifegradstufe 3; Softwaretechnik-Trends, Band35, Heft 3 (2015).
- [JohM2015] Johanning und Mildner, Car IT kompakt, Springer Fachmedien Wiesbaden, 2015.
- [Loch2008] G. Lochmeier, „www.maschinenmarkt.vogel.de,“ 28. 11. 2008. [Online]. URL: <https://www.maschinenmarkt.vogel.de/wege-zu-fehlerfreier-industrie-software-a-155735/index3.html>
- [SCHM2016] Schmunk, E: „Ein praktischer Ansatz zur Einführung und Umsetzung der Software Prozesse nach Automotive SPICE 3.0“, Masterarbeit, Hochschule Wismar, 2016.
- [VDA2015] VDA QMC Working Group 13 / Automotive SIG, Automotive SPICE, Process Reference Model/ Prozess Assessment Model, Version 3.0, 2015.
- [VDA2008] VDA®, Automotive SPICE® Prozessassessment modell, Frankfurt am Main: Henrich Druck + Medien, 2008.

No risk, no fun(ction), no future!

(Wie) lassen sich software-basierte Systeme in einer komplexer werdenden Welt noch beherrschen?

Michael "Pul" Paulsen

– Principal Consultant, seppmed gmbh –

Florian Prester

– Geschäftsführer, sepp.med gmbh –

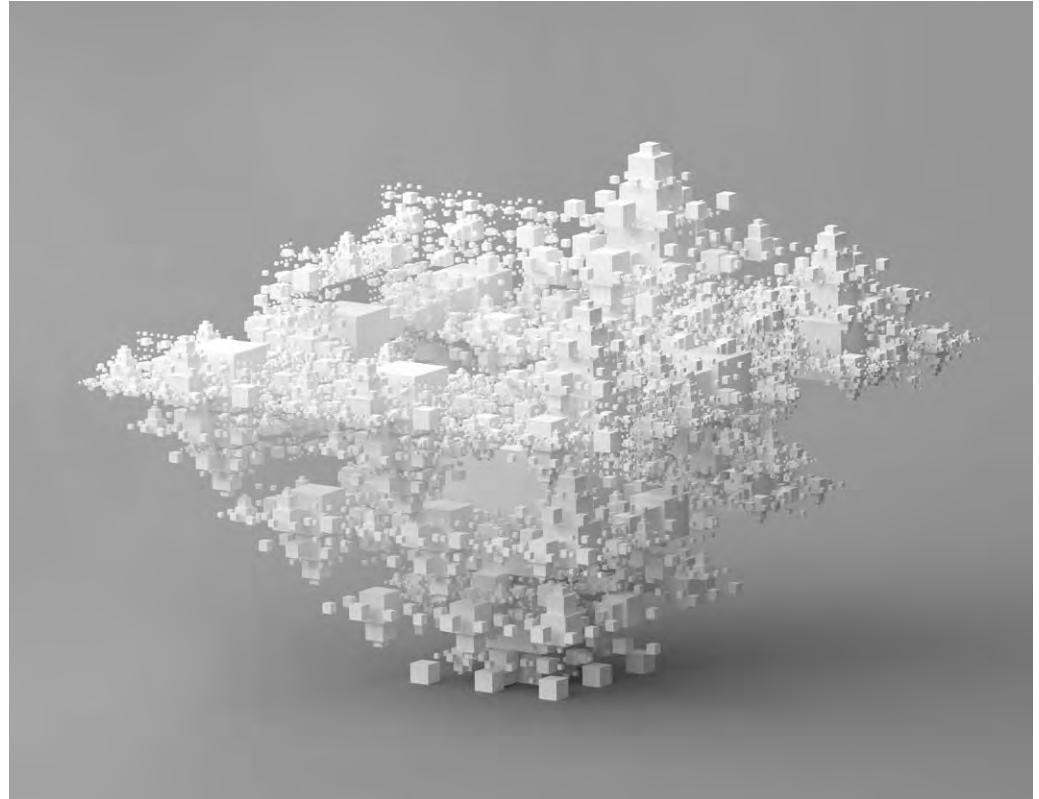
Panta rhei. Alles ist im Fluss, alles verändert sich. Unaufhaltsam, immer schneller, immer unüberschaubarer. Nicht erst heute. Spätestens seit Erfindung des Rades steht die Menschheit vor der Herausforderung, sich in der Veränderung zurechtzufinden. Es wachsen die Möglichkeiten, aber auch die Sorgen. Das meint übrigens die Bibel, wenn Adam und Eva nach dem Essen vom verbotenen Baum der Erkenntnis aus dem Paradies (der animalischen Sorglosigkeit) vertrieben werden.

„Früher war die Zukunft auch besser!“, hat der Münchener Erzkomödiant Karl Valentin humorvoll die Tradition jenes Gefühls beschrieben, das schon der alte Cicero mit „O tempora! O mores!“ zum geflügelten Wort machte und das seinen fatalistischen Gipfel im „No future!“ der Punk-Bewegung erreichte. Aber: That's life. Love it or hate it, but learn to live with it.

Auch in unserer Zunft, der Software-Entwicklung und -Absicherung, merken wir seit einiger Zeit, dass die „Zukunft früher auch besser war“. Grundannahmen ändern sich so massiv, dass ein Festhalten an bewährten Ansätzen alleine nicht mehr trägt, schon weil diese oft an einem nicht mehr zu leugnenden Aspekt scheitern: exponentiell wachsende Komplexität und Dynamik, nicht erst, aber vor allem im Zuge der grassierenden Digitalisierung.

Dieser Artikel soll einige Denkanstöße geben, wie wir mit den Herausforderungen der nächsten Jahre umgehen könnten. Dabei geht es weniger um Patentrezepte, die es per definitionem für eine ungewisse Zukunft nicht geben kann, sondern um eine grundsätzlich andere Haltung, aus der heraus sich dann Handlungstaktiken ableiten lassen. [>>](#)





No future?

Heute haben wir eher zu viel Zukunft¹ – und sie kommt zu schnell und zu mächtig; zumindest wenn man auf Sicherheit steht. Was für Ökosysteme und das Klima schon lange gilt, trifft zunehmend auch auf von Menschen erdachte Systeme zu: Die immer komplexer werdende Welt hält uns fast täglich vor Augen, dass wir sie nicht mehr im hergebrachten Sinne beherrschen.

Als Verfechter von Qualitätssicherung (oder auch nur als ethisch gefestigter Mensch) kann einem natürlich bange werden, wenn man sich gewisse technische Entwicklungen und ihre möglichen Folgen vor Augen hält und extrapoliert. Da werden von Hackern die Bremsen von (amerikanischen) Autos während der Fahrt außer Kraft gesetzt. Da werden (oft gegen den Rat von Experten) „smarte“ Ansätze bei kritischen

Infrastrukturen, wie z.B. Energienetzen, von der Politik forciert, während gleichzeitig der Cyberwar ausgerufen wird. Da werden teure Löcher in Innenstädte gegraben, um Nein, lassen wir diese Baustelle

Kürzlich hat uns ausgerechnet ein Stück künstliche Intelligenz (sagen wir lieber: Deep Machine Learning) gleichsam den Spiegel vorgehalten und die conditio humana aufgezeigt: Im Frühjahr 2016 ließ ein bekannter Software-Hersteller einen Social-Bot auf Facebook los, um zu beweisen, dass er nicht nur Betriebssysteme und Büroanwendungen entwickeln kann. Was als Marketing-Coup gedacht war, wurde von Teilen der Netz-Community so schnell „angelernt“, dass es innerhalb nur eines Tages die vielbeschworene Hasskultur verinnerlicht hatte und mit sexistischen und rassistischen Pöbeleien so auffiel, dass man sich gezwungen sah, das Profil des Bots wieder aus dem Netz zu nehmen. O tempora! O mores!

¹In der asynchronen Programmierung gibt es sogar einen generischen Datentyp namens Future, der ermöglicht, Ketten asynchroner Funktionsaufrufe zu bilden, als wenn sie synchron wären. Ein passendes Beispiel, wie die Menschheit es immer wieder schafft, mit geeigneten Abstraktionen eine eigentlich nicht mehr begreifbare Welt zu beherrschen.

Auch wenn dieses letzte Beispiel eher zum Schmunzeln anregt (nicht zuletzt, weil es einen Konzern getroffen hat, den man eher mit Marketing als mit Qualitätssicherung assoziiert), so bleibt einem das Lachen doch im Halse stecken. Man fragt sich nicht nur „In was für Zeiten leben wir eigentlich?“, sondern vor allem „Beherrschen wir unsere Systeme noch?“ – oder konsequenter: „Lassen sie sich überhaupt noch beherrschen?“.

„Die ich rief, die Geister, werd ich nun nicht los“. Wie Goethes Zauberlehrling mag sich mitunter mancher Ingenieur vorkommen. Auf der Stuttgarter Fachkonferenz „AutoTest“ wurde kürzlich von einem Crash eines autonomen Google-Cam-Autos berichtet. Bei einem zuvor tausende Male erfolgreich vollzogenen Spurwechsel in der Rush Hour traf es das erste Mal auf einen Linienbus auf der Nachbarspur, der nicht bremste, als es vor ihm in eine Lücke scherte. Man hatte versäumt, der Fahrsteuerung beizubringen, dass bestimmte Fahrer qua Berufsbild zur Kompromisslosigkeit neigen...

Soziale Systeme

Das Beispiel des autonomen Fahrens verdeutlicht einen Paradigmenwechsel, den wir in der realen Welt schon vollzogen haben. Während sich soziale Interaktion früher relativ überschaubar darstellte (es soll ja mal berittene Boten als Kommunikationsmedium gegeben haben), spielt sich heute ein Großteil in den sozialen Medien ab. Ohne dies kulturphilosophisch bewerten zu wollen, muss man festhalten, dass dabei völlig andere Regeln und Effekte auftreten, vor allem eine nicht vorhersagbare Dynamik, siehe das eben erwähnte Bot-Beispiel.

Was wir Menschen erleben, gilt aber zunehmend auch für unsere Software-Systeme. Traditionell waren sie, was man früher „Stubenhocker“ genannt hätte. Auch heute gibt es noch jede Menge Monolithen (egal, ob sie in COBOL, Java oder .NET entwickelt wurden), die am liebsten gar nicht mit der Welt reden und wenn überhaupt, dann nur über sehr eingeschränkte, oft grotesk spartanische Interfaces.

Mancher mag einwenden, dass das nicht stimmt, und auf eine strukturierte Modularisierung oder womöglich sogar auf eine SOA² verweisen. Ja, natürlich haben wir spätestens seit den Anfängen der funktionalen Dekomposition in den 70er Jahren gelernt, IT-Systeme aus immer mehr und kleineren Subsystemen und Komponenten zu bauen. Jedoch: Egal, wie groß es war, am Ende gab es doch bislang immer so etwas wie ein System, das nach dem Erklimmen der rechten Flanke des V-Modells einem Systemtest unterzogen und abgenommen werden konnte.

Dieses Paradies der Beherrschbarkeit müssen wir verlassen. Heute ist es häufig ein zentraler Gegenstand von Projekten aller Branchen, Systeme so zu bauen, dass sie eben nicht abgeschlossen sind, sondern offen mit anderen Systemen interagieren (selbst mit solchen, die nicht bekannt sind, ja vielleicht noch nicht einmal gebaut wurden). Egal, ob es dabei um das Verfügbarmachen und/oder Nutzen von offenen Services geht, um Erweiterungskonzepte (Wer hätte vor ein paar Jahren an Apps für Autos gedacht?) oder um eine massive Interaktion als Haupteigenschaft (wie beim autonomen Fahren oder bei IoT/Industry 4.0). Unsere Systeme sind Teil eines größeren Systems geworden, das wir nicht mehr testen können, ja das nicht einmal definiert werden kann! Sie begeben sich in Gesellschaft. >>

² Jener Hype, der grandios gescheitert ist bzw. oft zu einer massiven Verteuerung und Verlangsamung von IT-Projekten geführt hat, obwohl man genau das Gegenteil versprochen hatte. SOA verdeutlicht, was passiert, wenn man zulässt, dass bei etwas im Ansatz richtigen als erstes die Grundprinzipien über Bord geworfen werden (damals war das Ziel die leichtere Einführung, Anpassung und Verknüpfung von Business Services).

Roots to grow and wings to fly

Diese völlig neue Qualität von Systeminteraktion bedeutet immanent das Ende der Vollabsicherung. Natürlich würde eine theoretische Vollabsicherung schon am Ressourcenaufwand scheitern. Aber schon ihre Definition ist utopisch: Experten schätzen, dass selbst mit drei Millionen Kilometer Testfahrten zum Sammeln von Testdaten maximal etwa zwei Prozent aller möglichen Szenarien erfassbar sind, auf die ein Auto im Laufe seines „Lebens“ treffen könnte. Vor allem ist diese Quote durch mehr Testkilometer nicht signifikant zu steigern! Wer will angesichts solcher Zahlen vorhersagen können, ob sich ein System in allen Situationen ordnungsgemäß verhält? Und was ist „ordnungsgemäß“ überhaupt?

Wenn man diesen Gedanken aber konsequent weiterentwickelt, gelangt man zu folgendem Schluss: Wir müssen die Art und Weise, wie wir Systeme entwickeln und ihre Qualität sichern, grundsätzlich überdenken. Wir können sie nicht länger nur wie ein abgeschlossenes Werkstück betrachten, das man in einem definierten handwerklichen Prozess fertigstellen und mit der Schieblehre auf zuvor definierte Qualitätskriterien überprüfen kann.³

Stattdessen sind unsere zukünftigen Systeme eher unseren Kindern gleich: Nachdem wir sie erzogen und lebensfähig gemacht, ihnen also gute Wurzeln gegeben haben, werden sie flügge – oder wir schubsen sie aus dem Nest. Wir entlassen sie nolens volens in eine Zukunft, die wir liebevoll und aufmerksam begleiten mögen, die wir aber weder vorab kennen noch wirklich beeinflussen können. Wir können ihnen nur unsere Erfahrungen und Werte mitgeben.

Wenn wir lernten, Systeme so zu entwickeln wie unseren Nachwuchs – angefangen bei der Liebe und dem Verantwortungsgefühl – hätten wir einen großen Schritt gemacht; nicht nur als Software-Experten, sondern als Menschheit. Denn unverkennbar befinden wir uns an der Schwelle dessen, was vor einigen Jahrzehnten noch (meist düstere) Science-Fiction war: Maschinen nehmen zunehmend eigenmächtig am Leben teil und organisieren diese Teilhabe untereinander. Und damit das nicht in Schreckensszenarien endet, müssen wir sie nach unseren ethischen Maßstäben „erziehen“.

Das fängt bei der von Isaac Asimov in seinen "Roboter-Gesetzen" vorgeschlagenen Forderung an, in künstlicher Intelligenz ganz tief zu verwurzeln, dass eine Maschine unter keinen Umständen einen Menschen aktiv verletzen oder töten darf. Analog dem Art.1 GG, an den wir als Menschen gebunden sein und in dessen Ethos wir unsere Kinder erziehen sollten. Umgekehrt – so wie wir uns von unseren Kindern mitunter zu Recht anhören müssen „Davon verstehst Du nichts“ – muss es vielleicht sogar bei der Forderung enden, dass Menschen die Entscheidungen von Maschinen nicht „verschlimmbessern“ dürfen, weil sie ihnen auffassungsmäßig nicht gewachsen sind.⁴

Jedem, der am Entwurf von Systemen beteiligt ist, sei vor diesem Hintergrund folgende Tugenden empfohlen: Vertrauen, Respekt und Demut. Besonders dieser letzte, leider etwas aus der Mode gekommene Begriff verdeutlicht, worum es geht: Teil eines Größeren zu sein, das man weder begreifen noch kontrollieren kann, zu dem man aber konstruktiv seinen Teil beizutragen hat.

³ Bitte das nicht falsch verstehen. Handwerkliche Qualität ist nicht nur nicht unwichtig geworden, sie wird zukünftig noch wichtiger (s. folgende Ausführungen). Aber sie kann nur noch ein Teil der Absicherung sein, für die größeren Aspekte braucht es andere Ansätze

⁴ Die Kollision zweier Flugzeuge im Luftraum über dem Bodensee vor einigen Jahren ging auf menschliches Versagen zurück. Zunächst hätten die Flugzeuge gar nicht so dicht aneinander gelost werden dürfen. Das eigentliche Problem war aber folgendes: Die Autopiloten der Flugzeuge hatten untereinander ein Ausweichmanöver verabredet, das die Distanz zwischen ihnen maximieren sollte. Der Pilot der daraufhin absinkenden Maschine hat dieses Manöver, das er womöglich nicht schnell genug als solches verstanden hat, korrigiert und damit genau in die andere Maschine gelenkt!



*„WIR MÜSSEN DIE ART UND WEISE, WIE WIR SYSTEME ENTWICKELN
UND IHRE QUALITÄT SICHERN, GRUNDSÄTZLICH ÜBERDENKEN“*

Michael "Pul" Paulsen / Florian Prester

No risk, no function

Es gibt nichts Tödlicheres als das Leben. Eltern wissen das, weil sie in ständiger Sorge um ihre Kinder leben. Trotzdem hören wir ja nicht auf, Kinder zu erziehen und auch loszulassen. Kontrolliertes Risiko könnte man jenen Erziehungsansatz nennen, der Kindern ermöglicht, eigene Fehler zu machen und daraus zu lernen. Eine endgültige Sicherheit kann es nicht geben, man kann sich nur gegen unnötige Risiken wappnen.

Was bedeutet das, auf unsere Systeme übertragen? Als im Frühjahr der erste Todesfall durch autonomes Fahren gemeldet wurde, bei dem ein begeisterter Tesla-Fahrer unter einen querfahrenden weißen Laster geriet, der von seinem System als Wolke missinterpretiert wurde, war der Aufschrei groß, inklusive jener Stimmen, die es „ja schon immer gesagt“ haben.

Psychologisch ist das verständlich, da ein Tabu gebrochen wurde, indem ein Mensch ums Leben kam, nachdem er dieses einer Software anvertraut hatte. Aber unabhängig davon, wie man persönlich zum autonomen Fahren steht, muss man nüchtern konstatieren, dass weltweit täglich hunderte Menschen bei ähnlichen Unfällen sterben, weil sie oder andere Menschen einen Fehler gemacht haben.

Und jetzt wird es bitter für Freunde einer Vollabsicherung: Wenn (und dieser Trend scheint nicht aufzuhalten zu sein) wir uns durch Maschinen wirklich so massiv entlasten wollen, dass sie sogar unsere Entscheidungen treffen, müssen wir zugestehen, dass diese letztendlich vom Menschen gemachten Maschinen auch fatale Fehler machen werden (und dürfen), so bitter das im Einzelfall auch ist. Entscheidend sind unser verantwortungsvoller Umgang mit der Technologie und die gezielte Rückführung solcher Erfahrungen in diese hinein. Wäre es nicht toll,

wenn Systeme solche Erfahrungen untereinander weitergäben, wie es die Menschen seit Urzeiten am Lagerfeuer gemacht haben?

Jeder vermeidbare Schaden ist einer zu viel, gewiss. Aber eine fehlerfreie Welt ist Utopie. Und deshalb zählt am Ende die ganz einfache statistische Frage, ob in Summe die Maschinen eine geringere Fehlerquote haben als Menschen. Ein Verteufeln aufgrund einzelner Fehler ist weder hilfreich noch sollte es eine sinnvolle Technologie verhindern. Auch durch Gurte sind schon Menschen gestorben, weil sie sich z.B. nicht schnell genug aus einem brennenden Auto befreien konnten. Trotzdem würde niemand den überragenden Sicherheitseffekt von Rückhaltesystemen in Frage stellen. Vielleicht denken wir in einigen Jahrzehnten so auch über autonomes Fahren, wenn wir unser Unbehagen abgelegt haben.

Defensive Systeme

Was allerdings gegeben sein muss: Maschinen dürfen Menschen nicht unnötig in Gefahr bringen – sei es durch eine zu große Fehlerquote (dann darf die Maschine schlichtweg nicht an Menschenstelle Entscheidungen treffen, sondern bestenfalls assistieren, so wie es die Stufen auf dem Weg zum autonomen Fahren ja auch vorsehen), sei es durch fahrlässiges Handeln. Selbstverständlich wäre es in keiner Weise hinnehmbar, wenn z.B. autonome Fahrzeuge so etwas wie jene verbotenen Straßenrennen veranstalten würden, bei denen regelmäßig Menschen zu Schaden kommen und bei denen wir uns fragen, was bei der Erziehung der Fahrer schief gelaufen ist. Was uns wieder zum Kernpunkt führt: Maschinen sollten und können Regeln konsequenter befolgen als Menschen. Und letztendlich dienen ja z.B. Verkehrsregeln einem: der gegenseitigen Rücksichtnahme und Risikominimierung. Wäre das nicht ein lohnendes Entwurfsziel? >>

Fahrschülern werden aus gutem Grund die Prinzipien defensiven und vorausschauenden Fahrens eingeblinst. Und an diesen Prinzipien sollten wir auch unsere Systeme ausrichten. Nicht nur die zum autonomen Fahren, sondern wir sollten es zum allgemeinen Prinzip unseres Entwurfs machen, um übergreifenden Systemverbünden ein möglichst geschmeidiges und fehlerarmes Funktionieren zu ermöglichen.

Das könnte ein defensives System analog zu einem defensiven Fahrer kennzeichnen:

- Es beachtet die zum Gemeinwohl konstruierten Regeln insbesondere anderen Systemen (Mensch und Umwelt sind in diesem Sinne auch Systeme) gegenüber.
- Es rechnet umgekehrt damit, dass andere sich gegebenenfalls nicht an diese Regeln halten oder schlichtweg Fehler machen, und besitzt deshalb Taktiken zur Folgefehlervermeidung (das betrifft zum einen Fehler im eigentlichen Sinne, zum anderen Schutz gegen Schadsoftware).
- Es ist in der Lage, ein nicht ganz konformes Verhalten anderer zu tolerieren, um den gesamten Prozess nicht zu blockieren.⁵
- Es verhält sich auch in Stresssituationen regelkonform (im Extremfall dadurch, dass es eher den Dienst einstellt, als Regeln zu verletzen, deren Einhaltung es nicht mehr garantieren kann.⁶).

Der Transfer des „defensiven Fahrens“ ist also gar nicht so schwer, die konkrete Umsetzung dann vielleicht schon eher – aber dafür sind wir alle ja Experten oder kennen welche. In vielen Umfeldern, gerade von Großprojekten, wird das vor allem eine organisatorische Herausforderung sein, man denke an Conway's Law.

Im Folgenden nur einige Anregungen, wo die digitale Reise hingehen könnte:

- **Microservices** als der State of the Art des „Teile und herrsche“ sind besonders durch die „Shared-Nothing“-Architektur ideal für den Entwurf „sozialer“ Systeme.
- **Konzisere, formale Systembeschreibungen** wie Zustandsmodelle, DSLs und natürlich die bereits existierenden Schnittstellen-Standards, wie z.B. AUTOSAR, müssen konsistenter und konsequenter genutzt werden.
- **Hochintegrierte Entwicklungsverfahren**, wie z.B. Behavior Driven Design, die Medienbrüche vermeiden und früh eine hohe Automatisierungsquote erzielen.
- **Robustheitstests**, deren Ziel es ist, nachzuweisen, dass der Service in allen denkbaren Konstellationen seinen Qualitätsanforderungen genügt.
- **Machine Learning** in der Qualitätssicherung (z.B. in Hinblick auf Planung enger Testressourcen, aber auch auf inhaltliche Qualität).
- **Betriebskäfige**, die (analog zu den Käfigen von Industrierobotern) den eigentlichen Service **zur Laufzeit** einbetten, um Schaden zu verhindern, Annahmen zu verifizieren u.ä..

Diese Liste ließe sich beliebig fortsetzen und vertiefen, was hier aber aus Platzgründen entfallen muss. Wir freuen uns auf Erfahrungs- und Ideentausch, aber natürlich auch auf Zusammenarbeit bei solchen Themen.

⁵ Dieser Punkt bezieht sich im Straßenverkehr z.B. darauf, jemanden einscheren zu lassen, der sich falsch eingeordnet hat, da man schließlich nicht wissen kann, ob er nur einen Fehler gemacht hat oder sich bewusst einen Vorteil verschafft. Bei Systemen wäre darunter z.B. eine liberale Schnittstellenimplementierung zu verstehen, die kleinere Fehler akzeptiert oder gar kompensiert (wenn z.B. nur die PLZ, aber nicht der Ortsname angegeben ist).

⁶ Viele Assistenzsysteme stellen z.B. ihren Service angekündigt ein, wenn man ihn sich gerade besonders wünschen würde: Z.B. wenn bei schwerem Schneefall die Datenqualität von optischen oder Radarsensoren nicht mehr ausreicht. Im Gegensatz zu einem Menschen sollte sich ein System nicht überschätzen.

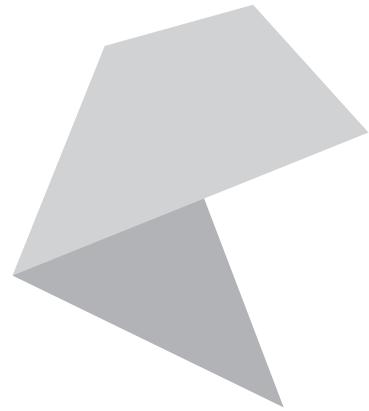


Fazit und Ausblick

Der bewusst etwas kontroverse Artikel sollte aufzeigen, dass wir uns von der Vollabsicherung vergangener Tage verabschieden müssen, weil in Zeiten der Digitalisierung, des autonomen Fahrens und des IoT ein Großteil des Verhaltens zwischen den Systemen stattfindet, wo er kaum zu beschreiben, vorherzusehen und abzusichern ist. Als Experten für Qualitätssicherung sollten wir deshalb aber nicht den Kopf in den Sand stecken, sondern auf ein Umdenken hinwirken.

Je unmöglichter wegen der Komplexität eine Absicherung im Großen wird, desto wichtiger wird sie im Kleinen. Das ist vor allem auch eine Frage des Entwurfs und der Entwicklungsmethodik. Wir plädieren dafür, den Systementwurf in Richtung „defensive Systeme“ zu entwickeln, die sich analog zum defensiven Fahren so verhalten, dass das unüberschaubare Gesamtsystem möglichst reibungs- und fehlerlos funktionieren kann.

In einer Zeit, da die klassische Qualitätssicherung an Komplexitätsgrenzen zu scheitern droht, ist es an uns Experten vom ASQF, unsere Kunden zu neuer Qualität zu führen. Das ist unser Apfelbäumchen in einer kaum noch beherrschbaren Welt. |



KONTAKT ZU DEN AUTOREN

*florian.prester@seppmed.de
michael.paulsen@seppmed.de*

Requirements-Engineering

Ein interaktives, iteratives Schauspiel in vier Akten

Chris Rupp

– General Manager, SOPHIST GmbH –

Karin Schlör

– Marketing, SOPHIST GmbH –

Christian Pikalek

– Beratung & Training, SOPHIST GmbH –

Der ASQF e.V. wird 20 Jahre alt! Dazu erst mal einen herzlichen Glückwunsch von uns SOPHISTen. 20 Jahre für Software-Qualität zu kämpfen, Menschen auszubilden, zusammen zu bringen und zu motivieren, das ist etwas, für das man gar nicht genug Lob aufbringen kann. Sicherlich machte das nicht immer Spaß und es gab bestimmt auch so manchen Moment des Zweifels. Ein ahnungsvoller Blick auf den Stand der Systementwicklung im Jahr 2016 sagt einem aber, dass der Kampf für genau diese Ziele nicht sinnlos ist, schon viel passiert ist und auch noch viele Jahre andauern wird. Somit drücken wir dem ASQF e.V. schon mal die Daumen für die nächsten 20 Jahre.

Zufälligerweise feierte auch die SOPHIST GmbH dieses Jahr ihr 20-jähriges Bestehen. Und auch wir kämpfen seit 20 Jahren für eine bessere IT-Welt. Unsere Frontlinie ist das Requirements-Engineering, für das wir auch nach 20 Jahren noch die Fahne hoch halten. Wir hoffen der folgende Artikel kann dazu beitragen, noch mehr Freunde für Requirements-Engineering zu finden.

Prolog

BER. Was assoziieren Sie mit diesem Kürzel? Moderner Hauptstadtflughafen, komfortables Reisen, professionelles Projektmanagement? Oder doch eher Milliardenkosten, neuer Eröffnungstermin und Brandschutzbestimmungen? Die meisten Menschen tendieren wahrscheinlich zu Variante zwei. Sicher ist die Planung eines Flughafens eine enorm komplexe und sensible Angelegenheit und Probleme im Laufe der Bauphase können nicht gänzlich vermieden werden. Aber ein solches Ausmaß an gravierenden Mängeln, an explodierenden Kosten und einem sich verschiebenden Eröffnungstermin lässt sich nicht so einfach rechtfertigen.

Sie stellen sich jetzt sicher die Frage, was ein Bau-
projekt mit Requirements-Engineering zu tun hat.

Die Antwort: ziemlich viel. Zugegeben, alle Probleme kann auch das beste Requirements-Engineering (kurz: RE) nicht verhindern, aber ein professionelles und konsequentes RE hätte dieses Großprojekt vor einigen unangenehmen Überraschungen bewahrt.

Aber was macht denn nun ein solches professionelles Requirements-Engineering aus?

Requirements-Engineering in einem Satz

Das Requirements-Engineering umfasst die Ermittlung von Anforderungen der Stakeholder an ein System sowie deren sinnvolle Dokumentation, Prüfung und Verwaltung.

Der Requirements-Engineer nimmt die wichtige Rolle des Kommunikators zwischen den Stakeholdern, also allen an einem Projekt-Beteiligten, ein. Somit muss er nicht nur über ausgeprägtes analytisches Denkvermögen und Fachwissen des RE verfügen – ebenso wichtig ist die soziale Intelligenz mit Soft Skills wie Konfliktlösungs- und Kommunikationsfähigkeit sowie Empathie. Diese Fähigkeiten sind essentiell, betrachtet man die vielschichtigen und abwechslungsreichen Aufgaben eines Requirements-Engineers.

Um Ihnen einen kleinen Einblick in die Arbeitsweise und Methoden im Requirements-Engineering zu geben, folgt ein kurzer Abriss zu den vier Akten des Requirements-Engineerings – die Ermittlung, Dokumentation, Prüfung und Verwaltung ihrer Anforderungen.



Glückwunsch ASQF

„ZUR VERBESSERUNG UNSERES GESAMTEN AGILEN SOFTWARE-ENTWICKLUNGS-
PROZESSES UND INSbesondere ALS IMPULSGEBER FÜR NEUE INNOVATIONEN
IN DER QUALITÄTSSICHERUNG NUTZEN WIR DIE VIELFÄLTIGEN ANGEBOTE,
DIE UNS ALS MITGLIED IM ASQF OFFEN STEHEN.“

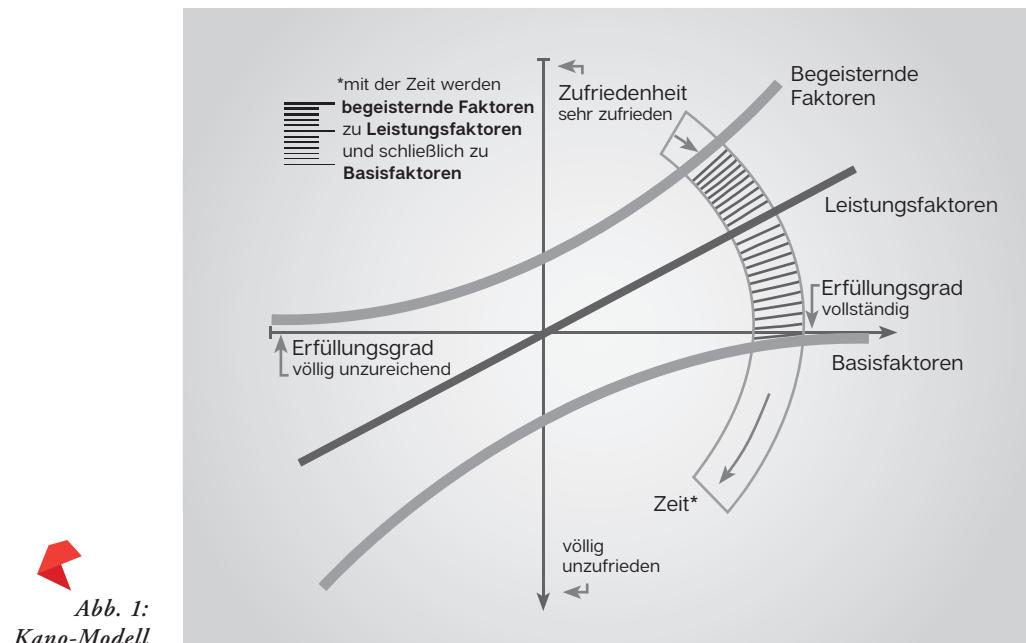
Bernd Jakobs, IWM Software AG

1. Akt: Ermitteln

In der Ermittlungsphase gilt es, sämtliche Anforderungen an ein System herauszufinden und zu sammeln. Die Quellen dieser Anforderungen sind weitreichend und umfassen neben Dokumenten und existierenden Systemen vor allem das Wissen sämtlicher Stakeholder.

Um an das Wissen der Stakeholder zu gelangen, muss der Requirements-Engineer individuell auf die Stakeholder eingehen und situationsspezifisch Ermittlungstechniken wählen. Eine entscheidende

Rolle spielen hierbei die verschiedenen Arten von Wissen und die damit einhergehenden Produktfaktoren. Denn Stakeholder werden nicht alle Anforderungen, die sie an ein System haben, auch explizit erwähnen. Dr. Noriaki Kano unterscheidet in seinem Kano-Modell [Kano84] drei Arten von Produktfaktoren, die unterschiedlichen Wissensebenen entsprechen (siehe Abb. 1). >>



Basisfaktoren entsprechen dem unterbewussten Wissen eines Stakeholders. Es handelt sich also um Anforderungen an ein System, die als so selbstverständlich gesehen werden, dass Sie nicht explizit erwähnt werden.

Leistungsfaktoren sind jene explizit formulierten Anforderungen an ein System. Diese entsprechen dem bewussten Wissen eines Stakeholders, das heißt, der Stakeholder verlangt bei der Konzeption des neuen Systems explizit nach einer bestimmten Funktion.

Begeisterungsfaktoren sind Systemmerkmale, die dem Stakeholder noch unbekannt sind. Er kann sie daher nicht explizit formulieren, sondern er entdeckt sie erst als angenehme Überraschungen, wenn er die Faktoren genannt bekommt beziehungsweise sieht. Diese Faktoren entsprechen dem unbewussten Wissen eines Stakeholders.

Die Aufgabe des Requirements-Engineers ist es, diese drei Faktoren zu beachten und entsprechend Ermittlungsmethoden zu wählen, um an die unterschiedlichen Wissensebenen zu gelangen. Die Ermittlungsmethoden reichen von klassischen Befragungstechniken (vor allem für bewusstes Wissen), über Beobachtungs- und artefaktbasierte Techniken (vor allem für unterbewusstes Wissen) bis hin zu Kreativitätstechniken (vor allem für unbewusstes Wissen).

2. Akt: Dokumentieren

Natürlich gilt es dann die ermittelten Anforderungen sinnvoll zu dokumentieren. Auch hier muss der Requirements-Engineer situationsspezifisch Entscheidungen treffen – unter anderem über die Art der Dokumentation. Man unterscheidet zwischen natürlichsprachlicher und modellbasierte Dokumentation. Natürlichsprachliche Dokumentation, also Prosa-Anforderungen, bieten den Vorteil, dass sie in erster Linie für jedermann verständlich sind. Man muss sich also nicht erst, wie zum Beispiel bei der modellbasierten Dokumentation, die nötige Theorie aneignen, um Prosa-Anforderungen zu verstehen und damit arbeiten zu können. Gegen eine natürlichsprachliche Dokumentation und für ein modellbasiertes Vorgehen spricht andererseits,

dass Prosa-Anforderungen oft mehrdeutig und missverständlich sind. Demgegenüber kann die modellbasierte Dokumentation, wird sie korrekt angewendet, in kompakter Form eine eindeutig verständliche Darstellung der ermittelten Anforderungen bieten.

Der Requirements-Engineer ist gefragt, die für das jeweilige Projekt sinnigste Dokumentationsart zu wählen und entsprechend anzuwenden.

Zwar hilft die Kombination von natürlichsprachlicher und modellbasierte Dokumentation dabei, eine höhere Qualität der Anforderungen zu erreichen, trotzdem benötigen Sie im dritten Akt eine Qualitätssicherung.

3. Akt: Prüfen und Abstimmen

Die Anforderungen sind zusammengetragen – die Spezifikation steht. Damit ist der Job des Requirements-Engineers allerdings noch nicht getan. Nun gilt es, die Qualität der Spezifikation zu prüfen und gegebenenfalls abzustimmen. Denn: Je später Fehler oder Unklarheiten festgestellt werden, umso teurer wird die Behebung der Fehler. Es empfiehlt sich daher bereits im Vorfeld der Anforderungsdokumentation Qualitätskriterien zu definieren und anhand dieser, die Anforderungen zu prüfen. Für die zum Vorschein gekommenen Mängel müssen Lösungen gesucht und gefunden werden. Dabei ist es allerdings nicht auszuschließen, dass mehrere valide Anforderungen erhoben wurden, diese sich aber gegenseitig blockieren, so dass nicht alle „guten“ Anforderungen gleichzeitig erfüllt werden können und ein Konflikt entsteht. Das heißt, Ziel des Prozesses ist es ebenfalls, dass alle Stakeholder im Konfliktlösungsprozess einbezogen wurden und eine Lösung gefunden wird, die für alle Parteien vertretbar ist. Bleibt noch der letzte Akt.

4. Akt: Verwalten

Die Verwaltung der Anforderungen, auch Anforderungsmanagement genannt, ist ein nicht zu unterschätzender Punkt in der Arbeit eines Requirements-Engineers. Anforderungen müssen in ihrem gesamten Lebenszyklus, von der Ermittlung und Dokumentation, über die Systementwicklung bis hin



zu Test und Wartungen, verwaltet werden. Dabei ist es ganz natürlich, dass sich Anforderungen ändern, dass etwas ergänzt oder gelöscht werden muss. Um eine konsistente Änderung zu gewährleisten, muss ein definierter Änderungsprozess existieren.

Zusätzlich muss die Struktur der Spezifikation so gestaltet sein, dass alle Stakeholder sie lesen, verstehen und damit arbeiten können. Außerdem muss die Spezifikation immer auf dem neuesten Stand sein und darf im Laufe der Zeit nicht an Qualität und Übersichtlichkeit einbüßen.

Epilog

Es bleibt spannend, wann der Hauptstadtflughafen nun tatsächlich seinen regulären Betrieb aufnehmen kann. Sollten Sie einmal in die Verlegenheit kommen, einen Flughafen zu planen... oder auch ein etwas kleineres Projekt anzugehen, seien Sie sich der Bedeutung von Requirements-Engineering bewusst – Sie sparen sich jede Menge Geld und Nerven.

Eine ausführliche Betrachtung der Arbeitsweise und Methoden im Requirements-Engineering inklusive zahlreicher Praxisbeispiele finden Sie in der druckfrischen Neuauflage des Standardwerks des Requirements-Engineerings – „Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil.“ von Chris Rupp und den SOPHISTen [Rupp14]. |

Referenzen

[Kano84] Kano, Noriaki; Tsuji, Shinichi; Seraku, Nobuhiko; Takahashi, Fumio: „Attractive Quality and Must-be Quality.“ in: Quality – The Journal of the Japanese Society for Quality Control.“ Vol. 14, Nr. 2, S. 39-44, 1984.

[MASTER] Die SOPHISTen (2013): „MASTER. Schablonen für alle Fälle.“ Online verfügbar unter: https://www.sophist.de/fileadmin/SOPHIST/Publikationen/Broschueren/SOPHIST_Broschueren_MASTeR.pdf

[Rupp14] Rupp, Chris; die SOPHISTen: „Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil.“ Nürnberg: Hanser, 2014.

[UML2] Rupp, Chris; Dr. Qeins, Stefan; die SOPHISTen: „UML 2 glasklar.“ Nürnberg: Han-ser, 2012.



**KONTAKT
ZU DEN AUTOREN**

heureka@sophist.de



Software-Qualität in Zeiten von Digitalisierung und Internet der Dinge

Matthias Schneider

– Leiter Test & Integration Center, T-Systems Multimedia Solutions –

Stephan Oswald

– Test & Integration Center, Principal PreSales Specialist,
T-Systems Multimedia Solutions –

DevOps wird zu DevQOPs

Die Zeiten, als ein neues Release über einen längeren Zeitraum erst umfangreich getestet werden konnte, gehen dem Ende entgegen. In Zeiten der Digitalisierung kommt es für Unternehmen darauf an, ihr Business flexibel an die immer schneller wechselnden Erfordernisse des Marktes anzupassen. Agilität und Individualität werden immer wichtiger. Die Anzahl der Deployments pro Woche steigt rasant. Daher müssen Änderungen an den IT-Lösungen schneller und in kleineren Teilschritten umgesetzt werden.

Um diese neuen Anforderungen zu bewältigen, ist ein technisches und methodisches Herangehen von der Entwicklung bis in den Betrieb notwendig. DevOps bietet hier den lösenden Ansatz zur Prozessverbesserung. Durch DevOps arbeiten die einzelnen Disziplinen Entwicklung, Test, Betrieb enger zusammen. Dies funktioniert nur durch immer stärkere Automatisierung. Doch welche Herausforderungen ergeben sich bei schnellen Release-Zyklen? Neben der Einhaltung von Qualitätsanforderungen bei erhöhtem Fehlerrisiko gilt es Testprozesse aufgrund der vollautomatisierten Arbeitsabläufe von der Entwicklung bis in die Produktion anzupassen. Ebenso ist es notwendig, technisch interpretierbare Ergebnisse für automatisierte Entscheidungsprozesse zu erzeugen.

Um diesen Herausforderungen zu begegnen, eignet sich die automatisierte Quality Service Lane (QS-Lane). Sie integriert alle entscheidenden Testverfahren, wie u.a.:

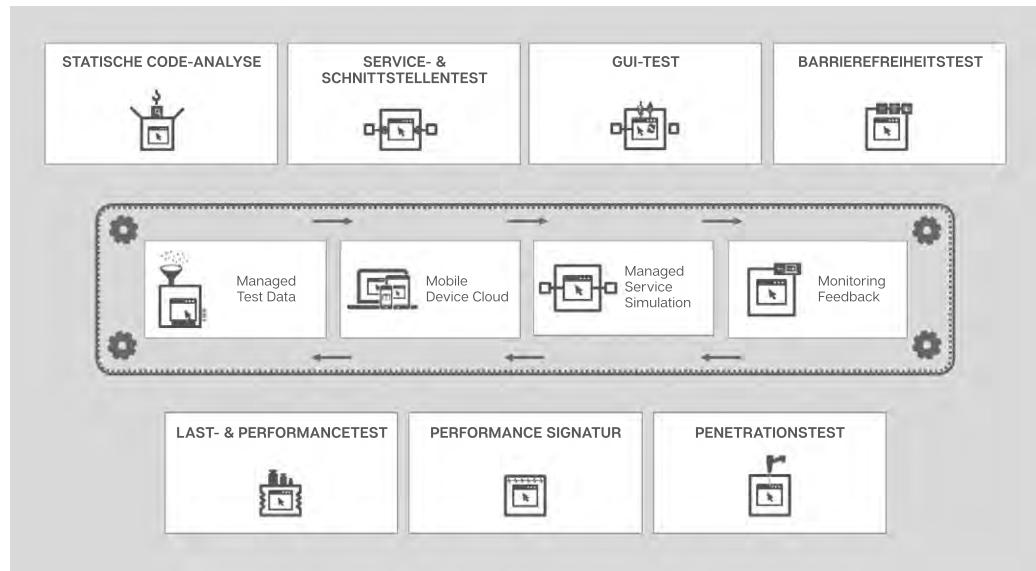
- Source-Code-Vermessung,
- automatisierter Funktionstest,
- Schnittstellentest,
- Penetrationstest,
- Performance Signatur & Last- und Performancetest,
- Barrierefreiheitstest,
- Continuous Application Monitoring.

Dadurch stehen Testergebnisse in Echtzeit zur Verfügung und ermöglichen durch sofortige Rückkopplung eine schnelle Weiterentwicklung und Optimierung der Applikation bzw. der Testobjekte. Die QS-Lane bildet somit die Basis für automatisierte Freigabeentscheidungen und sichert eine konstant hohe Software-Qualität & Performance der Applikation – was schlussendlich zu Zeit- und Kostenersparnis führt. Qualitätssicherung im DevOps-Prozess: DevQOps.

Quality @ IOT

Alles wird smarter – immer mehr Endgeräte werden miteinander vernetzt: angefangen bei heimischen Geräten, wie dem Kühlschrank, über Autos bis hin zu Infrastruktur- und Industrieanlagen. In vielen Bereichen soll durch die Vernetzung über das Internet das Leben einfacher und die Produktions- sowie Steuerungsprozesse effizienter und flexibler werden.

Dabei muss die Qualität der Systeme in Form von Verlässlichkeit und Robustheit gegen Ausfälle und Angriffe stets von zentraler Bedeutung sein, denn



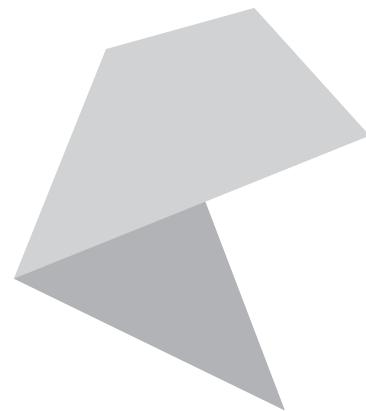
mit der zunehmenden Vernetzung steigt auch das Bedrohungspotential. Durch die neu entstehenden vernetzten Strukturen, können Ausfälle und Störungen einzelner Komponenten weitreichende Auswirkungen für das Gesamtsystem haben. Angriffe wie Stuxnet, Shamoon, Flame oder DuQu sind dabei nur die Spitze des Eisbergs. In vielen Fällen werden dabei nicht nur die IT-Software-Systeme beeinträchtigt, sondern es entsteht sogar ein physischer Schaden an den Endgeräten.

Je mehr Strukturen zu Smart Grids, Connected Car Infrastrukturen, Industrial Control Systems etc. vernetzt werden, desto höher ist die Gefahr, dass Technologien, die nicht für einen Betrieb über öffentliche Netze vorgesehen waren, nun plötzlich über das Internet erreichbar sind. Physische Abschottung reicht dann nicht mehr aus.

Daraus ergeben sich auch direkte Anforderungen an die vernetzten Endgeräte und deren Applikationen. Was geschieht im Falle einer Schwachstelle, die viele Endgeräte betrifft? Sind Hersteller, Betreiber, Entwickler in der Lage die Firmware Ihrer Endgeräte zu aktualisieren, wenn ein neuer (Open-) SSL Bug publik wird? Wurde eine entsprechende Möglichkeit bei der Entwicklung der Endgeräte und deren Software mit berücksichtigt?

Qualitätssicherung im Bereich IoT und Industrie 4.0 bringt eine Vielzahl neuer Aspekte mit sich, so dass die Expertise der IT und den branchenspezifischen Spezialisten gebündelt werden muss.

Umso essentieller werden Spezialisten, die sowohl IT-Fachwissen, als auch technisches Verständnis für die auf den Endgeräten eingesetzten Technologien besitzen. Nur so kann ein funktionierender Qualitätssicherungsprozess etabliert werden, der das komplette System vom Endgerät bis zur etwaigen Cloud-Plattform betrachtet. Eine durchgängige Qualitätssicherung End-to-End (von Endgeräten bis in die Cloud) sind für IoT-Infrastrukturen essentiell. |



KONTAKT ZU DEN AUTOREN

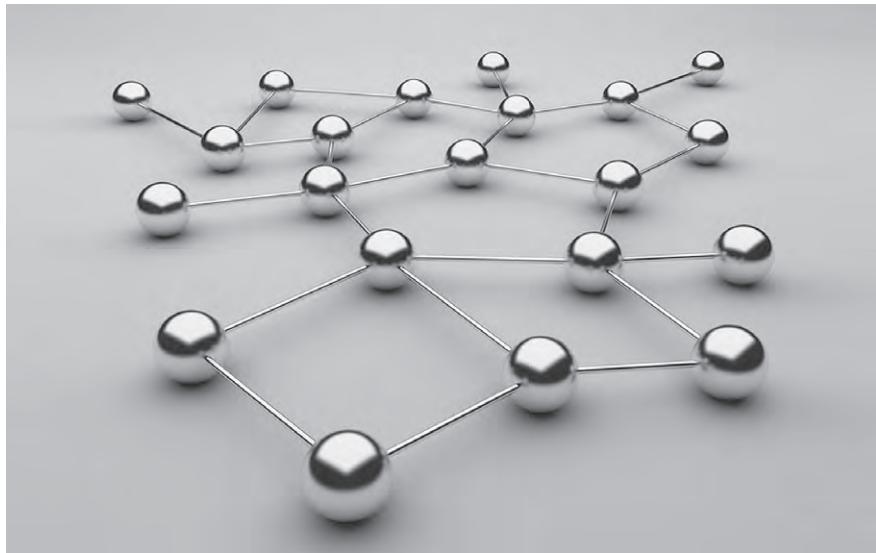
*Matthias.Schneider01@t-systems.com
Stephan.Oswald@t-systems.com*

Networking – die zeitgemäße Währung

**Warum lokales Netzwerken und
Expertenrunden so wichtig sind**

Alexander Weichselberger

– Vizepräsident des Austrian Testing Boards / Mitglied der SEQIS-Geschäftsleitung –



Das Austrian Testing Board, kurz ATB, ist die Vertretung des ISTQB® in Österreich. Neben der Qualitätssicherung von ISTQB®-Inhalten, der Erstellung von Lehrplänen und Zertifizierung von Schulungsanbietern sowie der Auswahl von geeigneten Prüfungs- und Zertifizierungsstellen, verfolgt das ATB das Ziel, Software-Testerinnen und -Tester durch Vernetzung zu fördern und selbst Treiber für bestimmte Themeninhalte zu sein. Die beiden letztgenannten Ziele stellen zugleich wesentliche Be rührungs punkte in der jahrelangen Zusammenarbeit zwischen dem ASQF und dem ATB dar.

Wachstum und Professionalisierung

Seit seiner Gründung vor 14 Jahren ist das ATB schnell gewachsen. Entscheidend war dafür, ne-

ben engagierten Einzelpersonen, auch der Zulauf der verschiedenen Beratungsfirmen und Trainingsanbieter. Wesentlich beim Aufbau der Prüfungsinfrastruktur war die Unterstützung durch das International Software Quality Institute, deren alleiniger Gesellschafter der ASQF ist. Vor allem in der Anfangsphase hat das Institut mit wertvoller Erfahrung und konkreten Maßnahmen zur Professionalisierung des ATB beigetragen. Diese enge Zusammenarbeit ist auch heute noch sehr erfolgreich. Darüber hinaus hat der intensive Kontakt zwischen dem ASQF und dem ATB den Austausch von innovativen Ideen, Kenntnissen und Erfahrungen auf den Gebieten der Software-Entwicklung und des Qualitätsmanagements wesentlich vorangetrieben. Gemeinsames Networking zählte und zählt hierbei zu den besonderen Stärken der Partnerschaft.



„... ABER: IT PAYS OFF!“

Alexander Weichselberger

In zahlreichen Veranstaltungen und Kooperationen haben sich das ATB und der ASQF bereits gegenseitig erfolgreich unterstützt. Einmal mehr hat sich dabei gezeigt, wie wichtig eine ausgeprägte Netzwerk-Kultur ist. Diesbezüglich haben insbesondere die so genannten ATB-Expertentreffen an Bedeutung gewonnen. Fünfmal pro Jahr trifft sich die Community und lässt sich durch neue Themen, spezifische Projektberichte oder grundsätzliche Empfehlungen inspirieren. Aktuell ist die Teilnahme an den Expertentreffen für alle Interessierten kostenfrei.

Networking ist die neue Währung

Drehen wir etwas am Rad der Zeit und betrachten den Grund für eine Vielzahl der heutigen Errungenschaften. Historisch gesehen hat sich mit dem Geld ein Zahlungs- bzw. Verbindungsmittel zwischen dem Adel, dem Bürgertum und den Bauern etabliert. Dieser Umstand ermöglichte eine Vernetzung zwischen sehr ungleichen Marktteilnehmern, auch über die jeweiligen Landesgrenzen hinweg. Mit dem Einsatz von Geld wurden Leistungen, Produkte und Liegenschaften vergleichbar gemacht. Es gab eine akzeptierte und gemeinsame Referenzgröße, die von allen Seiten für den Austausch genutzt wurde und auch heute noch genutzt wird.

Zusammengefasst lag und liegt der Vorteil von Geld also nicht (nur) in seiner inhärenten Wertschöpfung, sondern auch in der geschaffenen Vergleichbarkeit, durch die die Akteure der unterschiedlichsten Lager miteinander vernetzt werden. Folgt man dieser Argumentation, wurde wohl durch diese Vernetzung das „global village“ begründet.

Geld ist als Basis für den gemeinsamen Austausch etabliert, doch zur weiteren Optimierung unseres Lebens beschäftigen wir uns immer intensiver mit unserem Umfeld und vernetzen uns direkt vor Ort. Dadurch sind wir schneller, informierter und vielseitiger. Wir erhalten Hilfestellungen und helfen anderen – ein tolles Gefühl! Abgrenzung und Abschottung reduzieren unsere Möglichkeiten. Durch das Miteinander im Networking erhalten wir einen echten zusätzlichen Mehrwert.

Netzwerken in Krisenzeiten

Politische, wirtschaftliche und auch gesellschaftliche Krisen stellen Unternehmen wie Institutionen vor zunehmende Herausforderungen. Mehr als je zuvor ist es deshalb wichtig, das eigene Netzwerk aktuell zu halten und gut aufzustellen. Funktionierende Netzwerke erleichtern uns den Kontakt zu anderen. Dies macht es einfacher, Geschäfte abzuwickeln und Herausforderungen gemeinsam zu stemmen, die allein undenkbar wären. >>

Echte Erlebnisse verbinden

Sicherlich, es gibt sie, die Netzwerkgruppen im Sozialen Web. Aber im Gegensatz zur globalen „Vernetzung“ via Facebook, LinkedIn und XING beziehen sich lokale Networking-Veranstaltungen wie der ATB-Expertentreff explizit auf Treffen vor Ort. Diese Differenzierung ist wesentlich!

Sich zu sehen, zu treffen und gemeinsam etwas zu erleben, verbindet deutlich stärker als die virtuelle Verknüpfung. Es trägt dazu bei, Vertrauen rasch aufzubauen und Erfahrungen auszutauschen.

Auf diese Weise werden neben dem reinen Informationsaustausch auch Körpersprache und Stimme übertragen. Das ist deutlich besser, weil persönlicher, und hat mehr Substanz. Teilnehmer eines Netzwerk treffens haben untereinander eine stärkere Verbindung!

Netzwerken leicht gemacht

Erfolgreiches Networking vor Ort ist gar nicht so schwierig - wenn man einige wenige Grundregeln beachtet:

1. Organisation

Um lokales Networking effizient und effektiv betreiben zu können, benötigen Sie eine Person, die sich ernsthaft dem Vernetzungsgedanken verschrieben hat und in diesem Sinne ihre Zeit und Energie gerne in eine Community investieren will. Abgesehen von ausreichend Manpower, ist die Auswahl des Veranstaltungsortes von großer Bedeutung für erfolgreiches Networking. Achten Sie insbesondere darauf, dass die technischen Voraussetzungen für Ihre Veranstaltungen erfüllt sind. Zudem sollte die Location verkehrstechnisch gut erschlossen sein (Möglichkeit der Anreise mit öffentlichen Verkehrsmitteln, ausreichend Parkplätze, etc.).

Wichtig ist weiterhin ein Ablaufkonzept, das eine kurzweilige Veranstaltung garantiert und ausreichend Platz für die Vernetzung zwischen den Teilnehmern erlaubt.

2. Themen

Welches Thema Sie im Rahmen Ihres Networking bespielen, hängt nicht zuletzt von den Interessen ihrer Mitstreiter ab. Seien Sie offen gegenüber neuen Fragestellungen! Das ATB setzt beispielsweise bei seinen Networking-Veranstaltungen (ATB-Expertentreffs) auf testrelevante Themen wie agil vs. traditionell, Mobile, Crowdsourcing, Marketing für SW Test, Usability, Virtualisierung, IoT, usw.. Wir haben aber auch schon über Soft Facts im Software-Testen, über Soziokratie und über Prozesse (Scrum, Kanban) gesprochen.

3. Referenten

Wählen Sie zu den jeweiligen Themen bekannte Persönlichkeiten aus. Setzen Sie auf Experten, die bereit sind, Einblicke in ihr themenspezifisches Nähkästchen zu geben und die Community mit Tipps und Tricks zu versorgen. Idealerweise stellen Sie Material zu Themen und Speakern für den Download auf eigenen Portalen bereit.

4. Sponsoren

Sie können die Veranstaltung nur schwer unter freiem Himmel machen – und da Sie idealerweise die Veranstaltung auf Randtermine am Tagesende setzen, müssen Sie sich über die Verpflegung Ihrer Teilnehmer Gedanken machen. An sich ist dies kein großes Problem, es bedingt jedoch die Großzügigkeit von Sponsoren oder die Bereitschaft der Teilnehmer, für die Veranstaltung auch zu zahlen. Letzteres ist gewiss eine Hürde. Leider ist in vielen Unternehmen kein Budget für Networking vorgesehen – absurd, wenn Sie sich die Vorteile von Networking nochmals in Erinnerung rufen.



Teilnehmer bestimmen den Networking-Charakter

Erfolgreiches Networking ist nur so gut wie seine Teilnehmer. Sie tragen im Wesentlichen zum Erfolg bei – und zwar mit dem Wichtigsten, das sie geben können: Zeit. Angesichtsgesichts einer hohen Taktung und vieler Arbeitsstunden in Unternehmen ist das nicht selbstverständlich. Networking kostet Zeit und erfordert die Bereitschaft, zu geben (z.B. Kontakte, Tipps, Feedback, ...), aber: It pays off! |



**KONTAKT
ZUM AUTOR**

weichselberger@austriantestingboard.at

Sichere Software im Umfeld von Industrie 4.0

Best Practices für sichere Entwicklung, Einsatz und Wartung von Maschinensoftware

Prof. Dr. Sachar Paulus

– Fakultät für Informatik, Hochschule Mannheim –



Software wird zunehmend in allen Lebensbereichen eingesetzt – von der industriellen Produktion über die Logistik bis hin zur Gesundheitsbranche. Autonome und intelligente Systeme helfen uns dabei, schneller, effizienter und oft auch kundenorientierter zu sein und handeln zu können. Intelligente Geräte werden vernetzt und damit steuerbar aus der Ferne – das „Internet of Things“ verspricht auch große Gewinne an Lebensqualität.

Doch diese Entwicklung ist nicht risikofrei: Neben der für die Dienstqualität erforderliche massenhafte Bereitstellung von Daten (was in diesem Artikel nicht weiter betrachtet wird) ist die Angreifbarkeit von Software ein großes Problem.

Um die Vorteile der schönen neuen Welt auch zu angemessenen Risiken durchsetzen zu können, müssen wir lernen, möglichst sichere Software zu bauen (dies wendet sich an die Software- und Maschinenhersteller), und diese auch sicher einzusetzen. Wir betrachten exemplarisch den Bereich der industriellen Fertigung – die Erkenntnisse treffen aber genauso gut auch auf andere Bereiche zu, etwa den digitalisierten Gesundheitsbereich.

Dieser Artikel beleuchtet die Sicherheitsrisiken von Industrie 4.0, beschreibt Sicherheitsanforderungen für den Einsatz von Industrie-Software, formuliert Anforderungen an den Hersteller und geht auf die Erfordernisse bei der sicheren Software-Entwicklung ein.

Sicherheitsrisiken bei Industrie 4.0

Die Risiken durch die Computerisierung der Maschinen sind vielfältig: Das Ausspähen von Maschinen-Konfigurationsdaten (für den Konkurrenten interessant), die Manipulation von Steuerungsinformationen (zur bewussten Fehlsteuerung, also Sabotage) oder auch nur eine Einflussnahme auf die Qualitätssicherung (etwa indem Prüfroutinen übergangen werden) sind auch ohne die standardisierte Vernetzung durch die Programmierbarkeit der Komponenten möglich. Dabei reicht die Kenntnis über Befehlssätze, Schnittstellen und Zugangspunkte für Spionage und Sabotage schon aus. Allerdings ist es dafür immer noch notwendig, physikalisch in die (Nähe der) Produktion zu gelangen.

Durch die Standardisierung der Protokolle, die Vereinheitlichung der Steuerungen und die damit verbundenen Vernetzungseffekte vergrößert sich das Risiko nun erheblich, denn sowohl das Wissen über die Steuerbarkeit einer Anlage oder eines Geräts, als auch die Angriffspunkte, um mit einer Maschine zu kommunizieren sind nun verbreitet verfügbar.

Sabotage

Durch die Vernetzung ist die Steuerung von Industrieanlagen prinzipiell von überall möglich. Saboteure können bei mangelnder Sicherheit die Anlagen leicht fernsteuern und manipulieren – das dafür erforderliche Wissen ist frei zugänglich. Es gibt sogar Suchmaschinen für aus dem Internet erreichbare Industrieanlagensteuerungen – damit sind diese schnell auffindbar. Die Schäden können erheblich sein: Wird nur sukzessive und nur durch kaum merkliche, kleine Änderungen gesteuert, kann das zu produzierende Industriegut nicht mehr oder nicht mehr mit der erforderlichen Qualität hergestellt werden, ohne dass dies sofort entdeckt wird.

Spionage

Ein weiteres Risiko stellt die Industriespionage dar. Wissen über neue Produkte, neue Produktionsverfahren der Konkurrenz bis hin zu Konfigurationsdaten von Maschinen können über die Vernetzung zugänglich gemacht werden. Durch die standardisierten Protokolle bedürfen individuelle Angriffe zum unbemerkten „Abhören“ und „Absaugen“ keiner be-

sonderen Kenntnisse mehr – damit ist der Spionage durch Konkurrenten Tür und Tor geöffnet. Manche Staaten betreiben dies professionell (dann nennt man das Wirtschaftsspionage) und versorgen die eigenen Unternehmen mit strategisch interessanten Informationen.

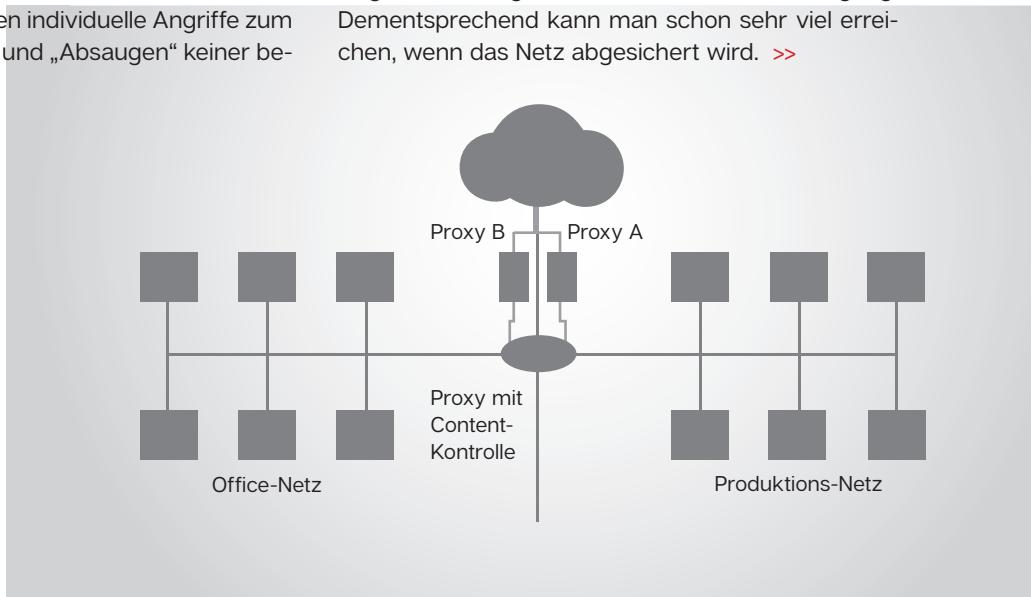
Qualitätsmängel

Durch die Vernetzung von Maschinen und Anlagen entsteht aber auch ein neues Qualitätsrisiko: Wie kann nun bei steigender Vernetzung nachgewiesen werden, dass die Produkte den Produktsicherheitsanforderungen und sonstigen Qualitätsanforderungen genügen? Mehr Standardisierung und Vernetzung bedeutet auch, dass a priori mehr legitimierte Mitarbeiter und Kooperationspartner auf die Produktionswerkzeuge zugreifen können (eventuell sogar sollen, um die Produktivität zu steigern) – und somit in die Produktion eingreifen können. Dies wird aber in den meisten Fällen gar nicht kontrolliert. Ein Qualitätsnachweis ist damit deutlich aufwändiger geworden.

Status quo – Wie ist Industrie-Software heute zu sichern?

Es sind einige grundlegende technische Vorkehrungen für einen Basisschutz erforderlich. Die meisten Angriffe benötigen einen netzbasierten Zugang. Dementsprechend kann man schon sehr viel erreichen, wenn das Netz abgesichert wird. >>

↖
Idealtypischer
Netzwerkaufbau.





„VIELE UNTERNEHMEN KÖNNEN IN DER REGEL KEINE ANGRIFFE SELBST ERKENNEN“

Prof. Dr. Sachar Paulus

Zuallererst muss das Netz, in dem die Produktionsmaschinen laufen, von anderen Computernetzen getrennt sein. Viel Schadsoftware wird über die Arbeitsrechner der Mitarbeiter eingeschleppt, daher darf insbesondere das Büro-Netzwerk keine beliebige Verbindung zum Produktionsnetzwerk haben. Die Maschinen dürfen niemals „direkt“ im Internet stehen. Damit ist gemeint, dass sie „von außen“ nicht direkt ansprechbar bzw. sichtbar sind. Anfragen an die Maschinen müssen immer über Proxies laufen, damit die Anfragen und die entsprechenden Antworten gepuffert und so auf schadhafte Inhalte kontrolliert werden können. Zudem ist es damit für Angreifer deutlich schwerer, eine Rückmeldung von eingeschleustem Schadcode oder auf durchgeführte Angriffe zu bekommen.

Fernwartung sollte idealerweise aus dem Inneren des Betriebsnetzes gestartet (oder zumindest bestätigt) und optimaler Weise durch einen Mitarbeiter durch Sichtkontrolle begleitet werden. Alternativ können entsprechende vertragliche Regelungen getroffen werden, um das betriebswirtschaftliche Risiko zu minimieren, wenn man beim Betrieb selbst keine Mitarbeiter stellen will oder kann.

Viele Unternehmen können in der Regel keine Angriffe selbst erkennen – die dafür erforderliche Fachkenntnis bindet oft zu viele Ressourcen. Um dennoch von Angriffserkennung zu profitieren und geeignete Alarne zu bekommen, sollte ein Netzwerkdienstleister mit der Angriffserkennung beauftragt werden – alternativ eine spezialisierte Firma, die entsprechende Sensoren im Netzwerk platzieren und überwachen kann.

Zudem sollte eine Verwaltung der Maschinen und der zugehörigen IT-Systeme eingeführt („Configuration Management Data Base – CMDB“) und regelmäßig geprüft werden, ob die Systeme im Netzwerk auch dazu passen (und keine fremden, auch keine virtuellen, Systeme auftauchen). Die Software der

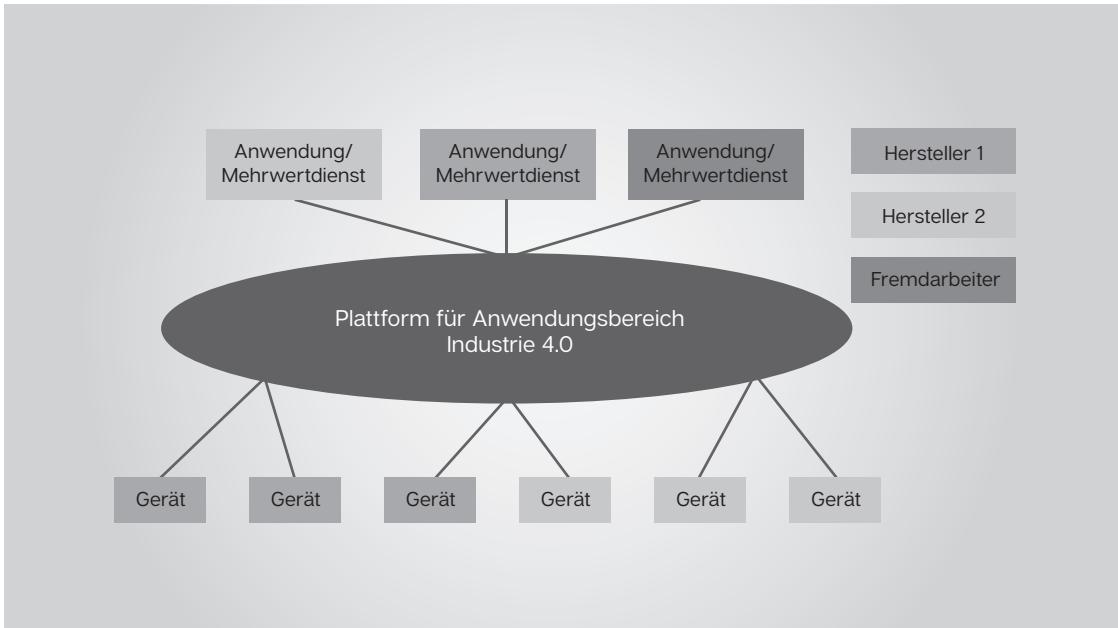
Maschinen muss auch stets möglichst aktuell sein, daher müssen – sofern verfügbar – für Software-Updates entsprechende Wartungsfenster eingeplant werden, in denen idealerweise die Updates auch getestet werden. Ist das nicht möglich, so ist die Absicherung des Netzwerks umso wichtiger.

Schließlich sollten alle Verbindungen zwischen Maschinen sowie zwischen Maschinen und anderen Computern verschlüsselt und authentifiziert ablaufen. Nur in dokumentierten und genehmigten Ausnahmefällen darf davon abgewichen werden, etwa wenn die Echtzeitfähigkeit dadurch gefährdet wäre (das ist aber nur selten wirklich der Fall). Die dafür erforderlichen Identitäten der Maschinen (Kennungen, Passwörter, Zertifikate) werden am besten mit einem entsprechenden Identitätsmanagement-System verwaltet.

Dies alles ist natürlich nur dann sinnvoll und zu angemessenen Kosten realisierbar, wenn die Maschinen-Software auf diese Aspekte vorbereitet ist – alleine die Update-Fähigkeit, idealerweise mit Integritätsschutz, ist für viele Hersteller schon eine große Herausforderung.

Anforderungen an den Hersteller

Die Berücksichtigung von Sicherheitsanforderungen ist nicht auf die Entwicklung von Software beschränkt – oft wird Software „versteckt“ eingesetzt (z.B. Pumpenhersteller, die Pumpen einbauen, welche Wartungsinformationen direkt via GSM „nach Hause“ versenden), oder Software wird für Zwecke eingesetzt, für die sie nicht geschrieben wurde. Man kann sich also nicht darauf verlassen, dass die Software in den Maschinen die eigenen Sicherheitsanforderungen berücksichtigt – egal, ob man die Maschinen betreibt, oder diese im Rahmen einer eigenen, größeren Lösung einsetzt.



Industrie 4.0-Plattformen werden die zentralen Einfallstore für Angriffe auf Maschinen sein.

In den folgenden Fällen sollte daher auch auf die Sicherheit von Maschinen, Systemen und Daten geachtet werden:

- beim Kauf von Maschinen (inzwischen werden kaum noch Maschinen ohne Software-Unterstützung ausgeliefert),
- bei der Beauftragung von Software-Entwicklung oder -Anpassung rund um Maschinen,
- bei der Nutzung von Industrie 4.0-Plattformen und -Diensten.

Dabei ist es von Vorteil, wenn man die Anforderungen an den Hersteller der Maschine (bzw. der Software) vertraglich regelt – denn es geht nicht nur darum, die Software in der Maschine sicher zu bekommen, sondern auch alle Einfallstore darum herum abzudichten. Eine solche vertragliche Vereinbarung über Sicherheitsaspekte sollte mindestens die folgenden Aspekte abdecken:

- Der Hersteller hält alle sicherheitsrelevanten Systeme (Firewalls, Windows Domänen Controller etc.) sowie alle Systeme, welche mit „ihren“ Maschinen verbunden sind, möglichst auf aktuellem Softwarestand.
- Der Hersteller verwendet ausschließlich sichere Kommunikation (Verschlüsselung der Verbindung mit Standard-Protokollen wie TLS oder IPSEC, Authentifizierung mit sicheren Passwörtern mit mindestens 12 Zeichen) – wenn er sich mit den Systemen des Unternehmens verbindet, aber auch etwa, wenn er selbst Fernwartung für die Administration seiner Systeme einsetzt.
- Der Hersteller setzt nur geeignetes qualifiziertes Personal ein.
- Der Hersteller informiert das Unternehmen unverzüglich über Veränderungen der Sicherheit, etwa, wenn Sicherheitslücken in der Maschinen-Software bekannt werden oder wenn eigene Systeme gehackt wurden. >>

Zudem ist es erforderlich, dass auch Lieferanten des Herstellers analoge Sicherheitszusagen machen, wenn sie die Sicherheit ihrer Maschinen direkt oder indirekt beeinflussen können.

Schließlich sind die Systeme dann leichter zu sichern, wenn sie Standard-Komponenten mit etablierten Sicherheitsprotokollen und -architekturen verwenden, statt selbst entwickelte Protokolle und eventuell sogar Programmiersprachen. So sollte der Hersteller etwa die „ISA 99“-Sicherheitsarchitektur erfüllen – das ist ein branchenübergreifender Sicherheitsstandard für Industrie 4.0.

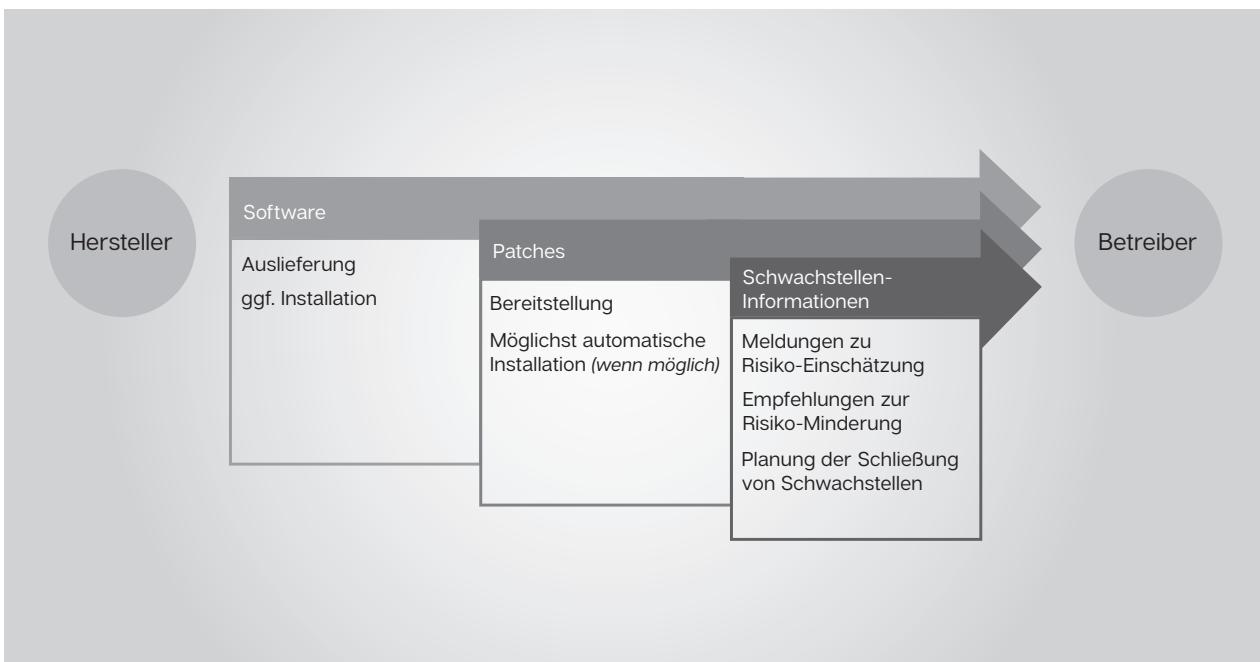
Best Practices bei der Entwicklung – von der IT lernen

Für Software-Hersteller – auch von Maschinen-Software – sollte die sichere Kommunikation zum Kunden, und der Umgang mit Sicherheitsvorfällen und Schwachstellen in der Software wie im letzten Abschnitt beschrieben zur Standardvorgehensweise gehören.

Für Software-Entwickler im Maschinenumfeld sind die folgenden Aspekte wichtig. Vieles davon kann man von der „Standard-IT“ übernehmen und sich von den Erfahrungen der letzten 15 Jahre einiges anschauen:

- Sicherheitsanforderungen sollten explizit formuliert und dokumentiert werden.
- Von vornherein sollte vorgesehen sein, dass Maschinen-Software auch gepatcht werden können muss, d.h. dass es einen vertrauenswürdigen Prozess geben muss, mit dem neue, überarbeitete oder auch von Schwachstellen bereinigte Software zeitnah und sicher eingespielt werden kann.
- Selbst entwickelte Sicherheitskonzepte sind sehr komplex. Die Wahrscheinlichkeit, dass man dabei Fehler macht, ist sehr hoch. Wo immer möglich, sollten unabhängige geprüfte, etablierte Sicherheitsprotokolle, und idealerweise sogar gut getestete Programmbibliotheken verwendet werden. Dies gilt auch für Passwort-Prüfung und -Verwaltung.

Sicherheitsrelevante Kommunikation und Dienstleistungen zum Kunden.



- Die eigenen Entwicklungsprozesse müssen abgesichert sein, damit Hacker sich nicht schon frühzeitig über das Einbringen von Hintertüren in die Software-Entwicklung einschleichen können (sichere Ablage des Quellcodes, gesicherter Netzzugang, Berechtigungskonzepte).
- Die Software sollte aus Sicherheitsgründen ausschließlich die für die Ausführung ihrer Aufgaben erforderlichen Berechtigungen haben (keine Admin-Rechte), damit im Falle eines Angriffs die Berechtigung nicht für einen Angriff auf weitere Anwendungen oder Rechner genutzt werden kann.
- Bei der Maschine-zu-Maschine-Kommunikation sollten möglichst keine Benutzer-Passwort-Kombinationen zur sicheren Authentifizierung der Komponenten verwendet werden, da deren Verwaltung schnell komplex wird und nicht mehr beherrscht werden kann. Stattdessen sollten Zertifikate für die Authentifizierung und verschlüsselte Kommunikation zwischen den einzelnen Komponenten verwendet werden.
- Die Software-Entwicklung sollte nach einer Methodik des „Secure Software Engineering“ durchgeführt werden – es gibt einige Vorgehensmodelle, wie etwa Microsoft's „Secure Development Lifecycle“, oder „OWASP OpenSAMM“, oder die (zertifizierbaren) Common Criteria (ISO 15408).
- Die Sicherheitsanforderungen sollten regelmäßig auf Umsetzung geprüft werden, dazu sind Black-Box- und White-Box-Tests erforderlich, die die Software im Rahmen von Sicherheitstests („White Hat Hacking“) auf Herz und Nieren prüfen. Je konkreter die Sicherheitsanforderungen formuliert sind (z.B. mit Akzeptanztests) und je besser die Testfälle entworfen wurden, desto wertvoller ist das Ergebnis der Sicherheitsüberprüfung.
- Entwickler der Software für Maschinen und Anlagen sollten regelmäßig auf Sicherheit geschult werden, und dies nach dem neuesten Stand.

Eine Reihe dieser Aspekte kann auch der Anwender bzw. Systemintegrator durchführen; so sind dafür etwa die Formulierung der Sicherheitsanforderungen oder auch die Sicherheitstests gut geeignet.

Ausblick

Wie die Argumentation in diesem Artikel zeigt, geht es nur zu einem Bruchteil um die Vermeidung von „schlechtem“ Code. Vielmehr benötigen wir einen hohen Reifegrad, eine Professionalisierung der Software-Entwicklung für Maschinen, um die Sicherheitsaspekte erfolgreich adressieren zu können. Nur dann werden wir in der Lage sein, die Vorteile der schönen neuen Welt mit angemessenem Risiko nutzen zu können – bei der industriellen Produktion, beim Internet of Things oder in der Medizintechnik. |



KONTAKT ZUM AUTOR

s.paulus@hs-mannheim.de

Veränderte Anforderungen an die Qualitätssicherung

von Data-Science-Lösungen

Prof. Dr. Andreas Schmietendorf

– Hochschule für Wirtschaft und Recht Berlin –

Jan Hentschel

– Ultra Tendency UG –

Die Verwendung qualitativ hochwertiger Daten und der korrekte bzw. nachvollziehbare Einsatz von Analyse-Algorithmen werden als Schlüssel für aussagefähige Datenanalysen angesehen. Dennoch kann die Reife der Qualitätssicherung konkreter Data-Science- bzw. Big-Data-Ansätze nicht überzeugen. Da benötigte Maßnahmen zur Qualitätssicherung unmittelbar mit den durchzuführenden Datenanalysen zusammenfallen, ergeben sich dramatische Veränderungen für das Software-Qualitätsmanagement. Vielfach fehlt es dabei an methodischen Vorgehensweisen einer inhärent benötigten Qualitätssicherung. Typische Beispiele beziehen sich auf unzureichend qualitätsgesicherte Rohdaten, die theoretische Fundierung der eingesetzten Algorithmen zur Datenauswertung, aber auch auf den Umgang mit konkreten Analyseergebnissen.

1. Motivation und Ziele

Die agilen Vorgehensweisen und Prinzipien haben die Art und Weise, wie Software-Qualitätssicherung durchzuführen ist, dramatisch verändert. Die Gründe dafür liegen in einer immer stärkeren Be-

rücksichtigung von konstruktiven Maßnahmen zur Sicherung der Software-Qualität, aber auch in kontinuierlich und automatisiert durchgeführten Entwickler- und Integrationstests. Auf dieser Grundlage lässt sich ein unmittelbares Feedback zum Qualitätsverhalten der Software erzielen. Die Sinnfälligkeit einer von der eigentlichen Entwicklung losgelösten Testdurchführung wird dabei zunehmend in Frage gestellt. Besonders deutlich zeigt sich dieser Effekt bei Lösungsansätzen im Umfeld analytisch genutzter Datenbanksysteme (Data Science bzw. Big Data). Hier zerfällt der Bezugsbereich einer Qualitätssicherung in mindestens zwei Betrachtungsweisen. Auf der einen Seite gilt es, die für eine Datenanalyse benötigten Infrastrukturen (ggf. auch aus der Cloud) bereitzustellen und auf der anderen Seite die operative Nutzung in Bezug auf durchzuführende Analysen sicherzustellen. Während bei der Infrastruktur noch klassische Tests zum Nachweis von Verfügbarkeiten, Performance, Sicherheitsmechanismen und angebotenen Funktionen zur Administration greifen, können „ad hoc“ Datenanalysen mit den zumeist szenarioorientierten Testansätzen der Software-Entwicklung nicht mehr erfasst werden.



Glückwunsch ASQF

„FÜR EIN UNTERNEHMEN WIE ATOS, MIT EINER ÜBER 9.000 MITARBEITER STARKEN GLOBALEN TEST UNIT, IST DIE ASQF-FACHGRUPPENARBEIT EIN ABSOLUTES MUSS, UM GERADE IN DEUTSCHLAND DIE AUSBILDUNG UND ZERTIFIZIERUNG IM TEST-BEREICH MITZUGESTALTEN.“

Ralf Mack, ATOS ORIGIN Deutschland

Dennoch bedarf es auch hier der Berücksichtigung vielfältiger Qualitätsaspekte bezogen auf die aus verschiedenen Quellen übernommenen Daten, die über diese ausgeführten Analyse-Algorithmen und ihre Ergebnisdarstellung und -interpretation.

2. Qualitätssicherung der Datenübernahme

Data-Science-Lösungen nutzen eine Vielzahl von Datenquellen, wie Sensordaten, Daten von Fertigungsstrecken, mobil verarbeitete Daten oder auch Systeme der Informationsverarbeitung bzw. soziale Netzwerke. Entscheidend für valide Analyseergebnisse ist die Qualität der eingesetzten Daten. Klassische Qualitätskriterien beziehen sich z.B. auf die Korrektheit, die Vertrauenswürdigkeit, die Fehlerfreiheit, die Aktualität oder auch die Konsistenz der Daten. Bei Lösungen aus dem Data Warehouse wird die Datenqualität zumeist während der ETL-orientierten Datenübernahme berücksichtigt. Im Zusammenhang mit Data-Science-Lösungen wird dieser Ansatz kontrovers diskutiert. Besonders radikal bzw. provozierend sind die unter [Henschen 2012] eingenommenen Positionen für (pro) bzw. gegen (contra) den Einsatz von ETL-Ansätzen:

- **Pro:** Für den Fortbestand von ETL-Ansätzen sprechen u.a. operativ genutzte IT-Systeme (Altsysteme) und die dabei verwendeten Datenformate bzw. genutzten Persistenz-Mechanismen. Im Mittelpunkt dieser Systeme steht die Gewährleistung der Datenkonsistenz und Transaktionssicherheit.
- **Contra:** Gegen den Fortbestand sprechen die Skalierbarkeit, Performance, Verfügbarkeit und Kosteneffektivität potentieller Hadoop-Installations. Eine redundante Speicherung der Daten kann durch die unmittelbare Ablage in Hadoop entfallen.

Eine detaillierte bzw. sachliche Auseinandersetzung wird wohl in beiden Positionen Ansätze für notwendige Veränderungen im Kontext von Data-Science-Lösungen identifizieren, was sich auch im folgenden Zitat niederschlägt:

„Faktoren, wie hohe Datenvolumina und -raten oder unvollständige Schemata, führen insgesamt zu einer Neudeinition der Daten-Integration: Weg vom Vorgehen des »Extract-Transform-Load«, hin zu einem »Extract-Load-Transform«.“ (Quelle: [Bitkom 2014]) >>

Durch den bei Data-Science-Lösungen vermeintlich einfacheren Umgang mit vielfältig strukturierten Daten wird auf Transformationsaufgaben bei der Datenübernahme von Quell- zu Zielsystemen auch aus Performance-Gründen zunächst verzichtet.

Klassisch beziehen sich diese z.B. auf Formattanpassungen, den Umgang mit unbelegten Datenattributen, die Verdichtung bzw. Zusammenführung von Daten, die Gewährleistung von Konsistenz-Bedingungen oder aber auf Aufgaben der Datenbereinigung. Diese Aufgaben fallen mit der Datenauswertung zusammen, wobei es entsprechend [Dong 2013] folgende Aspekte zu berücksichtigen gibt:

- **Schema Mapping** – Zusammenführung der ggf. schemaarmen Daten, so dass Beziehungen zwischen Datenobjekten erkannt werden können.
- **Record Linkage** – Identifizierung von Daten verschiedenster Quellsysteme, die zur gleichen logischen Datenquelle gehören.
- **Data Fusion** – Auflösen von Konflikten zwischen Datenquellen, so dass verfälschte bzw. unkorrekte Daten im Sinne des „veracity“ erkannt werden.

Zur Qualitätssicherung werden automatisierte Tests zum Vergleich der Ausgangs- bzw. Rohdaten mit den im Analyse-Cluster (Data Lake) abgelegten Daten benötigt. Aktuelle Ansätze nutzen z.B. Stichproben (Sampling) oder aber Differenzanalysen (Minus Queries) über den Datenbeständen.

Neben der Übernahme von Daten aus verschiedenen Quellsystemen bedarf es kontextbezogener bzw. semantischer Informationen (Metadaten) zur analytischen Verarbeitung. Diese werden benötigt, um gegebenenfalls vorhandene Muster und Zusammenhänge zwischen den Daten eindeutig zuordnen zu können bzw. eine empirische Interpretation überhaupt erst zu ermöglichen. Beispiele für derartige Daten beziehen sich unter anderem auf Zeitstempel, einsetzbare Schlüsselattribute, bekannte Beziehungen zwischen unterschiedlichen Fachdomänen oder auch auf die Abbildung/Mapping heterogener Datenstrukturen.

3. Qualitätssicherung eingesetzter Analyse-Algorithmen

Eingesetzte Algorithmen können grob in vier Kategorien eingeteilt werden: Verwendung externer Service-APIs, Query-basierte Algorithmen, individuell entwickelte Algorithmen und mathematisch-orientierte Algorithmen. Für alle Kategorien lassen sich ähnliche Prinzipien für die Qualitätssicherung ableiten, die primär aus der agilen Entwicklung stammen. Dazu gehören ein hoher Automatisierungsgrad, durchgehende Testabdeckung, insbesondere durch Entwicklertests und Dokumentation.

3.1 Verwendung von Data Science APIs

Zunehmend werden Algorithmen des maschinellen Lernens (kurz ML) als einfach zu integrierende Web-APIs zur Verfügung gestellt. Entsprechende Services beziehen sich z.B. auf Text- und Sentiment-Analysen, die Klassifizierung von Dokumenten oder auch auf die Bilderkennung. Konkrete Angebote finden sich mit IBM Bluemix¹ und den Watson Service APIs, dem Azure ML Studio² von Microsoft oder auch dem Marktplatz Algorithmia³. Vor der Verwendung stellen sich vielfältige Fragen nach der Qualität dieser Angebote in Bezug auf die hinter den Algorithmen stehenden Theorien, die Offenlegung der Implementierung, die Vertrauenswürdigkeit der Datenablage, verfügbare Praxiserfahrungen, Musteranalysen, betriebliche Qualitätsattribute oder auch hinsichtlich des Gerichtsstands.

In Abhängigkeit der Kritikalität durchzuführender Datenanalysen werden eigene Tests benötigt. Exemplarisch sei hier Bezug auf die automatische Klassifikation von Dokumenten durch einen ML-Services genommen. Ein Test könnte folgende Ziele berücksichtigen:

- Vergleich von Services unterschiedlicher Theorien wie z.B. Naive Bayes, Support Vector Machines, Taxonomien, Clustering.
- Voraussetzung in Bezug auf die zu klassifizierenden Dokumente wie z.B. enthaltene Wörter, Terme, Fehler, Leerstellen oder Rauschen.
- Aufwand zum Trainieren des Klassifikators im Sinne dafür einzusetzender Dokumente und erreichbarer Performance.
- Güte einer Klassifikation in Bezug auf korrekt bzw. fehlerhaft zugeordnete Dokumente.

Neben den funktionalen Eigenschaften gilt es auch, Compliance-Anforderungen zu berücksichtigen. In diesem Zusammenhang empfiehlt sich der Einsatz von Zertifikaten, die durch „vertrauenswürdige Dritte“ mit Hilfe eines offen gelegten Prüfprozesses zur Verfügung gestellt werden.

3.2 Query-basierte Analyse-Algorithmen

Die Implementierung von komplexen Analyse-Algorithmen, z.B. anhand von MapReduce in Hadoop, wird durch die Bereitstellung einer deklarativ einsetzbaren Abfragesprache in Anlehnung an die bekannte SQL-Notation deutlich vereinfacht. Im Diskurs des Hadoop-Ökosystems stehen mit Hive⁴ und Impala⁵ entsprechende Ansätze zur Verfügung. Beide basieren auf SQL-ähnlichen Anfragesprachen, die auch die grundlegenden Aspekte einer Data Definition Language (DDL) und einer Data Manipulation Language (DML) inkludieren. Somit müssen zwei Aspekte aus Sicht der Teststellung bedacht werden:

- Schemata- und Tabellenaufbau gemäß Spezifikation,
- Überprüfung der Ergebnisse von Anfragen.

Bezogen auf Hive bietet das Projekt HiveRunner⁶ gute Möglichkeiten, Test Suites in Form von JUnit- oder Scalatest-Tests, automatisiert über Maven oder SBT (Werkzeuge zum Management des Build Prozesses), für HiveQL-Skripte zu definieren und dabei durch Annotationen Setup-Skripte und Beispieldatensätze zu nutzen. DDL-Skripte werden hinsichtlich ihrer Spezifikation getestet. Dies bedeutet, sowohl Feldnamen und -datentypen als auch weiterführende Tabelleninformationen wie Partitionierung. >>

¹ <https://console.ng.bluemix.net/catalog/> | ² <https://studio.azureml.net/> | ³ <https://algorithmia.com>

⁴ <https://hive.apache.org/> | ⁵ <https://impala.apache.org/> | ⁶ <https://github.com/klarna/HiveRunner>

Um aus diesen Tests eine für Außenstehende nachvollziehbare Dokumentation zu erstellen, empfiehlt sich der Einsatz von Frameworks aus dem Bereich Behaviour-Driven Development. Prominentestes Beispiel für ein solches Framework ist Cucumber⁷.

Neben Hive und Impala ist HBase eine NoSQL-Datenbanktechnologie, die in Query-basierten Analysen eine wichtige Rolle spielt. Im Gegensatz zu Hive oder Impala liefert HBase sein eigenes Testframework (genannt HBase Mini-Cluster) mit, welches direkt in das Repository eingebettet ist⁸.

Unabhängig von den vorgenannten Werkzeugen der Open Source Community ist die Reife des in die Entwicklung integrierten Testprozesses von entscheidender Bedeutung. Die Güte datengetriebener Testansätze wird maßgeblich durch berücksichtigte Testdatensätze, die Einfachheit der Ausführung, die Art der Automation bzw. Wiederhol- und Reproduzierbarkeit, aber auch durch die Nachvollziehbarkeit und Dokumentation bestimmt.

3.3 Individuell entwickelte Analyse-Algorithmen

Der traditionelle Ansatz des im Data Science bestimmenden Hadoop-Ökosystems ist die Entwicklung individueller Analyse-Algorithmen. Als erstes ist hier MapReduce zu nennen. Entsprechende Algorithmen

werden typischerweise in Java oder auch in Python implementiert. Vergleichbar zum HBase Mini-Cluster bietet auch Hadoop sein eigenes Testframework an⁹. Alternativ liefert HBase Mini-Cluster ebenfalls verschiedene Clustervarianten mit. Verglichen mit den Query-basierten Analyse-Algorithmen ist es in dieser Kategorie noch wichtiger, kleine testbare Einheiten zu bilden.

Ein zunehmend an Bedeutung gewinnendes Projekt der Hadoop-Community findet sich mit Apache Spark¹⁰, welches sich aktuell in fünf Komponenten (Spark Core, Spark SQL, Spark Streaming, MLlib und GraphX) unterteilt. Damit unterstützt Spark eine effiziente Datenübernahme, vielfältige Algorithmen zur Datenanalyse (u.a. Textanalyse) und die Präsentation von Analyseergebnissen [Zaharia 2010]. Die Implementierung paralleler Analyse-Algorithmen kann unter Verwendung der MLlib (Machine Learning) mit Programmiersprachen wie Java, Spark, Python und R erfolgen. Die Zielstellung korrespondierender Entwicklertests wird u.a. durch das „spark-testing-base“-Projekt verfolgt¹¹.

Auch bei individuell entwickelten Analyse-Algorithmen ist eine Automatisierung zwingend. Dies schafft ein agiles Umfeld, in den Tests und Integrationaufgaben ständig (kontinuierlich) ausgeführt werden können.

⁷ <https://cucumber.io/> | ⁸ <https://github.com/apache/hbase/blob/master/hbase-server/src/test/java/org/apache/hadoop/hbase/HBaseTestingUtility.java> | ⁹ <https://github.com/apache/hadoop> | ¹⁰ <http://spark.apache.org/> | ¹¹ <https://github.com/holdenk/spark-testing-base>

3.4 Mathematisch-orientierte Analysen

Die Qualitätssicherung bei dieser Analyseart bezieht sich auf die Sicherstellung der messtheoretischen Grundlagen, wie z.B. Skaleneigenschaften der eingehenden Messgrößen und daraus folgend die Anwendbarkeit statistischer Methoden. Zumeist erfolgt der originäre Einsatz statistischer Verfahren (u.a. multivariate Datenanalysen) unter Verwendung einer dafür spezialisierten Programmiersprache wie z.B. R. Mit Hilfe von strukturprüfenden Verfahren (z.B. der Regressionsanalyse) und strukturentdeckenden Verfahren (z.B. der Clusteranalyse) erfolgt eine experimentelle Untersuchung der entsprechenden Datensätze. Im Bereich des „Predictive Maintenance“ hat sich die Durchführung von Analyse-Workshops mit einem interdisziplinär zusammengesetzten Teilnehmerkreis bewährt, so dass eine unmittelbare Validierung der Analyseergebnisse vorgenommen werden kann.

Aus Sicht qualitätssichernder Maßnahmen und aus der Sicht der klassischen Software-Qualitätssicherung ist diese Kategorie am wenigsten ausgeprägt, sollten die gewählten Technologien nicht in eine der drei vorhergehenden Kategorien eingeordnet

werden können. Aus diesem Grunde ist Sampling aus gewonnenen Rohdaten in dieser Kategorie eine wichtige Technik. Die Erstellung von repräsentativen Testdatensätzen, die wiederholbare Ergebnisse ermöglichen, stellt die grundlegende Basis dar, um Automatisierung zu schaffen.

4. Qualitätssicherung der Analyseergebnisse

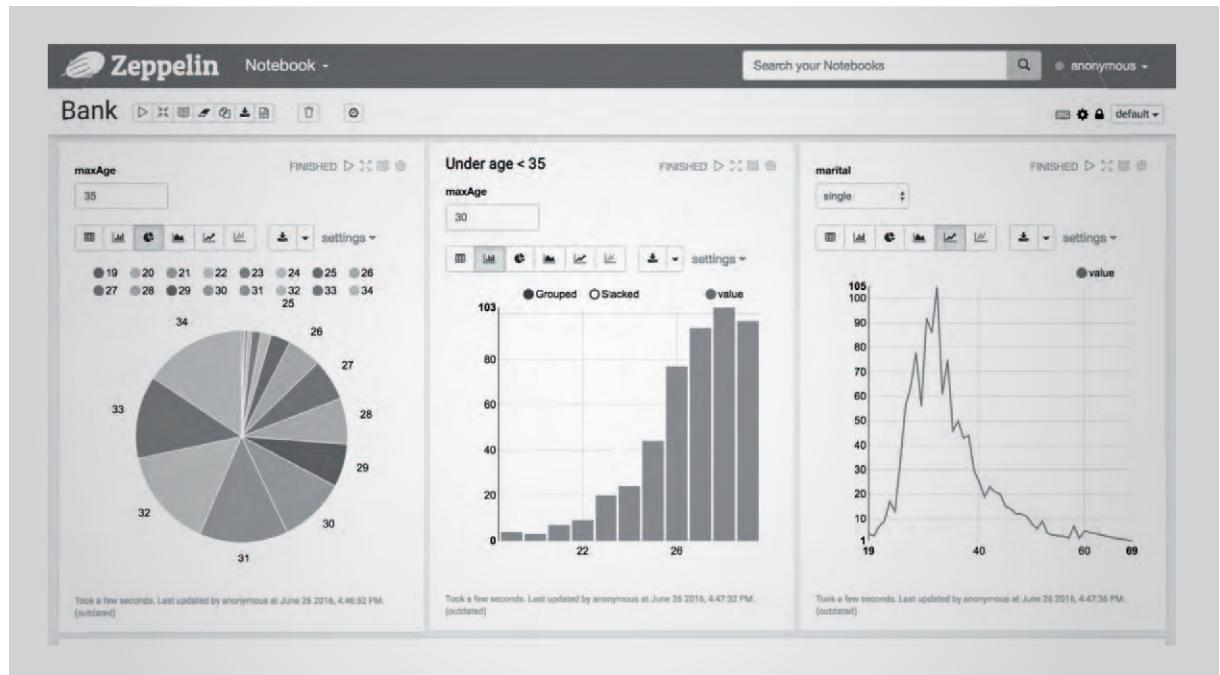
Ein Problem von Tests in datengetriebenen und analytischen Einsatzgebieten ist deren Verlässlichkeit. Insbesondere in mathematisch-orientierten Analysen können Testdaten perfekt zu den Tests passen, sich aber als Ausreißer in der Realität präsentieren. Sie können beispielsweise das Verständnis schaffen, dass die Geschlechterverteilung auf der Welt 90:10 beträgt. Tests auf dieser Datenbasis mindern schnell das Vertrauen in Lösungen des Data Science. Daraus ist es von entscheidender Bedeutung, Personen mit domänenpezifischem Wissen in den Test- und Qualitätssicherungsprozess mit einzubinden, bspw. durch deren Einbeziehung in aus der Software-Entwicklung bekannte Review-Verfahren. >>



Als zweite Komponente qualitätssichernder Maßnahmen ist das Verfahren für die Dokumentation zu wählen. Zu berücksichtigen ist, dass Erkenntnisgewinn im datengetriebenen Ansatz primär aus geleitetem Versuch und Irrtum besteht. Diesen Prozess gilt es zu dokumentieren. Der klassische Ansatz würde zu einer text- oder präsentationslastigen Dokumentation führen. Im Data-Science-Umfeld ergeben sich neben den bereits erwähnten Behaviour Driven Development die Chancen, Big Data orien-

tierte Datenanalyse-Tools für die Dokumentation der Tests einzusetzen. Ein für den Data Scientist passendes Werkzeug ist z.B. Apache Zeppelin¹². Zeppelin bietet eine Art agiler Dokumentation, in der der Data Scientist neben der Dokumentation auch funktionierende Beispiele und Visualisierungen einbetten kann. Anhand der in Zeppelin erstellten Notebooks können Test Suites zusammengestellt werden, die wiederholbar ausgeführt werden können. Abbildung 1 stellt ein entsprechendes Beispiel.

Abbildung 1: Beispielhafte Dokumentation in Apache Zeppelin (Quelle: [Zeppelin 2016])



¹² <https://zeppelin.apache.org/>

5. Zusammenfassung

Algorithmische Fähigkeiten sind für das agile Testen von Data-Science-Lösungen unerlässlich! Das in diesem Zusammenhang benötigte „Know-how“ erstreckt sich über das Beherrschene objektorientierter bzw. funktionaler Programmiersprachen wie Java, Scala und Python, über deklarativ orientierten Abfragesprachen wie HiveQL bis hin zur Implementierung mathematisch/statistischer Algorithmen mit Hilfe von R. Nicht immer ist das Beherrschene einer Programmiersprache wie Java und Python mit zusätzlichen Kenntnissen in deklarativen oder mathematisch-orientierten Programmiersprachen bei einer Person gegeben. Hier ist die interdisziplinäre Zusammenstellung der Teams mit sich ergänzendem Wissen (auch domänenspezifischem) ein Erfolgskriterium. Durch die Breite der eingesetzten Technologien und das benötigte Wissen um diese, ist eine traditionelle Aufteilung zwischen Entwicklung und Test eher schwierig. |



KONTAKT ZU DEN AUTOREN

*andreas.schmietendorf@hwr-berlin.de
jan.hentschel@ultratendency.com*

Quellenverzeichnis

- [Bitkom 2014] Bitkom Leitfaden Big-Data-Technologien - Wissen für Entscheider, Februar 2014, http://www.bitkom.org/files/documents/BITKOM_Leitfaden_Big-Data-Technologien-Wissen_fuer_Entscheider_Febr_2014.pdf
- [Dong 2013] Dong, X. L.; Srivastava, D.: Big Data Integration, in Proc. Of the VLDB Endowment, Vol. 6, No. 11, August 2013
- [Henschen 2012] Henschen, D.: Big Data Debate: End Near For ETL?, information-week.com, March 2012
- [Zaharia 2010] Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I.: Spark: cluster computing with working sets, HotCloud 10, pp. 10-10, 2010
- [Zeppelin 2016] Apache Zeppelin - A web-based notebook that enables interactive data analytics., <https://zeppelin.apache.org> (Abruf: 14.Oktober 2016)

Wanted: Testspezialisten dringend gesucht!

Graham Bath

– Senior Consultant, T-Systems GEI GmbH –



Blickt man zurück auf die letzten 10 bis 15 Jahre, so fällt auf, dass sich die Fähigkeiten und Kompetenzen auf dem Gebiet des Software-Testens erheblich verbessert haben. Durch die Einführung bestimmter Testrollen und Karrierewege wurde eine deutliche Verbesserung in den Fähigkeiten und im allgemeinen Ansehen von Software-Testern erreicht. Tester müssen sich in ihren Projekten nicht mehr als unqualifizierte Teammitglieder zweiter Klasse fühlen, denn sie sind nun viel besser auf die technischen und organisatorischen Herausforderungen vorbereitet.

Dies war und ist ein großer Fortschritt, nicht nur für die Betroffenen im Bereich Software-Test, sondern auch für alle anderen Beteiligten. Es ist einfach: Projekte und Organisationen profitieren von der verbesserten Software-Qualität, und die Verantwortlichen im Management können besser fundierte Entscheidungen über die erzielte Qualität treffen.

Es ist der Verdienst von Non-Profit-Organisationen wie dem German Testing Board (GTB), der Gesellschaft für Informatik (GI) und dem International Software Testing Qualifications Board (ISTQB®), die die Verbesserung der Fähigkeiten auf diesem

Gebiet möglich gemacht haben (obwohl es natürlich vermessen wäre zu behaupten, dass sie die Einzigsten sind, die daran beteiligt waren). Diese Organisationen sollten sich nun jedoch mehr auf die in Zukunft benötigten Fähigkeiten im Bereich Software-Test fokussieren. Wie der Titel dieses Artikels bereits andeutet, tut sich ein „Loch“ auf in der Weiterentwicklung der Fähigkeiten von Software-Testern. Um es auf den Punkt zu bringen: Wir brauchen mehr Spezialisten!

Der Artikel hinterfragt die Entwicklung der benötigten Fähigkeiten:

1. Was sind in Bezug auf die Fähigkeiten die Herausforderungen, mit denen unsere Industrie konfrontiert ist?
2. Sind die derzeit vorhandenen Möglichkeiten zur Verbesserung der Fähigkeiten ausreichend?
3. Welche Maßnahmen müssen ergriffen werden, damit der Bereich des Software-Testens mit den Entwicklungen in der IT-Industrie mithalten kann?

Die vom ISTQB® gewählte Strategie wird als Input für diese Diskussion herangezogen.



„ES BESTEHT EIN UNGLEICHGEWICHT ZWISCHEN GENERALISTEN UND SPEZIALISTEN.“

Graham Bath

1. Welche Fähigkeiten benötigen professionelle Software-Tester für die Zukunft?

1.1 Wo stehen wir?

Ausbildungsprogramme haben weltweit wesentlich zur Entwicklung der Testerfähigkeiten beigetragen, und Organisationen wie das ISTQB® können stolz darauf verweisen, dass in nicht allzu ferner Zukunft eine Million Tester-Zertifizierungen erreicht werden. Das ist ein beachtlicher Erfolg, keine Frage. Wenn wir jedoch genauer hinschauen, dann zeigt sich, dass die Pyramide der Fähigkeiten dieser Gruppe ziemlich flach ist. Anders ausgedrückt: Es besteht ein Ungleichgewicht zwischen Generalisten und Spezialisten.

Mit einer derart flachen Pyramide ist zu rechnen, da die Anzahl der Personen, die mit dem Karriereweg des Software-Testers beginnen, die Zahl der Personen weit übertrifft, die die höheren Qualifikationsstufen erreichen. Aber sollte eine derart flache Form der Pyramide unser Ziel sein?

1.2 Wo werden mehr Testfähigkeiten benötigt?

Betrachten wir zunächst einige spezialisierte Fähigkeiten, die Tester in der Zukunft berücksichtigen sollten:

- Domänen spezifische Fähigkeiten, wie z.B.
 - » Testen im Bankensektor,
 - » Testen in der Automobilindustrie,
 - » Testen in der Gesundheitsbranche,
 - » Testen in Nischenindustrien, wie der Spiele- und Unterhaltungsindustrie.
- Technologie-spezifisch, wie z.B.
 - » Testen des Internets der Dinge und von Cloud-basierten Systemen,
 - » Testen in DevOps,
 - » Testen von mobile Applikationen,
 - » Testautomatisierung.
- Testen von bestimmten Qualitätsmerkmalen, wie z.B.
 - » Performance,
 - » Sicherheit,
 - » Usability.
- Management-spezifische Fähigkeiten, wie z.B.
 - » Testdatenmanagement,
 - » Management von Testumgebungen,
 - » Testprozessverbesserung,
 - » Testmanagement in großen und organisatorisch komplexen Projekten. >>

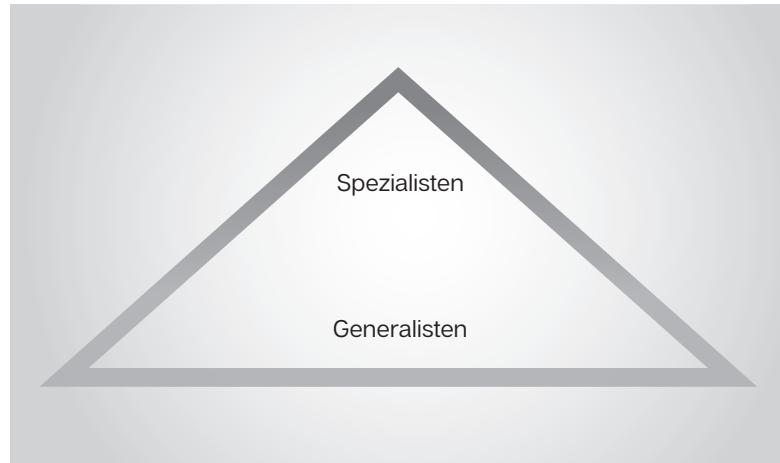
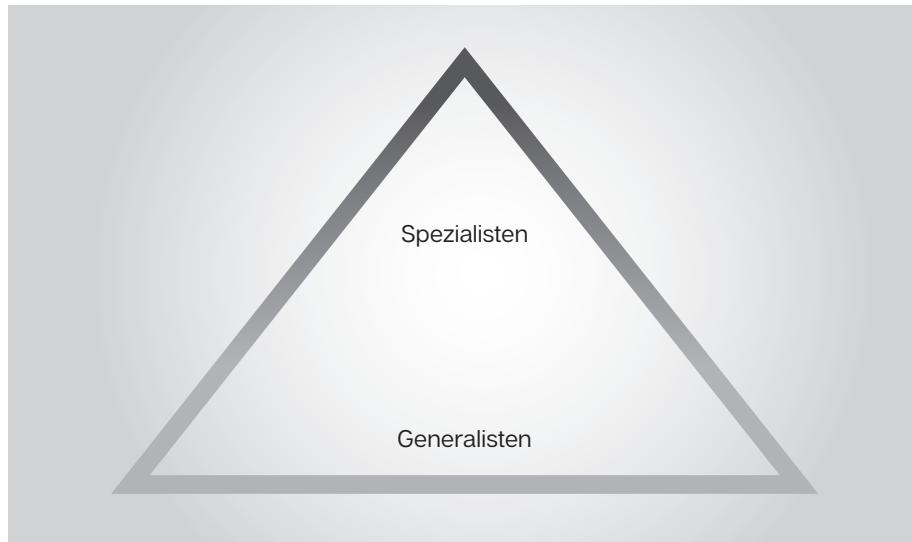




Abb. 2: Zukünftige Pyramide der Fähigkeiten



Legt man diese große Vielfalt an Spezialkompetenzen (die vorstehende Liste ist längst nicht vollständig!) zu Grunde, die im Software-Testing-Markt benötigt werden, dann wird deutlich, dass wir eine Pyramide der Fähigkeiten anstreben sollten, die eher der Pyramide in Abbildung 2 entspricht.

1.3 Bedarf an Prozessfähigkeiten

In Verbindung mit den oben erwähnten testspezifischen Fähigkeiten benötigen professionelle Software-Tester in zunehmendem Maße auch ein fundiertes Verständnis für andere Prozesse, die Schnittstellen mit dem Testen haben, u.a. mit Requirements Engineering, Software-Entwicklung, Usability-Design und -Engineering.

1.4 Bedarf an Führungsqualitäten

Die Software-Industrie hat zunehmend Bedarf an professionellen Software-Testern mit Spezialisierungen, die in den Projekten und Organisationen eine führende Rolle übernehmen können. Diese Personen zeichnen sich besonders durch exzellente Soft Skills aus und arbeiten mit dem höheren Management auf der Führungsebene des Unternehmens zusammen, um die unternehmensinternen Richtlinien in Bezug auf bestimmte Testthemen mitzugestalten. Sie sind die Katalysatoren,

die die spezialisierten Fähigkeiten fördern, die die Organisation benötigt, damit alle IT-Projekte davon profitieren können. Dabei ist völlig egal, ob diese Personen in formellen Kompetenzzentren oder in weniger formellen Pools organisiert sind. Was wirklich zählt, ist deren Kommunikationsfähigkeit sowie die Fähigkeit, bestimmte spezialisierte Aufgaben durchführen zu können.

Die allgemeine Schlussfolgerung ist klar und deutlich: Wir benötigen mehr Spezialisten mit Spezialfähigkeiten. Hier in Deutschland ist der Bedarf besonders akut. Unsere Industrie befindet sich in der Transformation zur IT 4.0, und es werden viele spezialisierte Fähigkeiten benötigt werden.

2. Möglichkeiten für die Entwicklung von Fähigkeiten

Die eigentliche Herausforderung, vor der wir in der Software-Industrie stehen, ist die Entwicklung der benötigten spezialisierten Fähigkeiten, die den künftigen Bedürfnissen der IT-Industrie gerecht werden. Es müssen Möglichkeiten geschaffen werden, die in Abb. 2 dargestellte Pyramide der Fähigkeiten zu erzielen. Das ISTQB® hat 2015 diesen Bedarf erkannt und ein neues Portfolio aufgesetzt, um diese Entwicklung zu unterstützen (siehe Abbildung 3).

In dieser Abbildung ist zu beachten, dass Spezialisten und Experten eigene, strategische Bereiche im Portfolio haben. Die Graustufen entsprechen den Graustufen in der in Abb. 2 dargestellten Pyramide der Fähigkeiten. Obwohl die Größe der in Abbildung 3 dargestellten Blöcke nicht entsprechend der tatsächlichen Anzahl von Zertifizierungen skaliert ist (die überwältigende Mehrheit der zertifizierten Tester sind im Bereich Core des Foundation Level zu finden), so zeigt die Abbildung doch deutlich das Engagement des ISTQB® zur Entwicklung der Fähigkeiten von Spezialisten und Experten in den kommenden Jahren.

3. Wie kann die Softwaretest-Branche mit den Entwicklungen in der IT Schritt halten?

Das in Abb. 3 dargestellte ISTQB®-Portfolio ist ein großer Entwicklungsschritt und wird von zahlreichen Ausbildungsmodulen von verschiedenen anderen Organisationen flankiert. Dennoch bleibt noch viel zu tun, wenn wir die in Abbildung 2 dargestellte zukünftige Pyramide der Fähigkeiten erreichen wollen. In diesem Abschnitt werden einige der Aufgaben und Herausforderungen betrachtet, die in diesem Zusammenhang bewältigt werden müssen.

3.1 Die Ausbildungsportfolios müssen um spezifische Module im Bereich Specialist erweitert werden

Ausbildungsportfolios wie das ISTQB®-Produktportfolio in Abbildung 3 müssen mit der richtigen Mischung an spezialisierten Modulen ausgestattet werden, um den Bedarf der Industrie abzudecken. Es wird erwartet, dass die meisten der in Punkt 1 aufgelisteten Themen in solchen Portfolios zu finden sind, insbesondere die „großen fünf“: Sicherheit, Usability, Performanz, mobile Applikationen und Testautomatisierung. Diese Module befinden sich in der Entwicklung. Zum gegenwärtigen Zeitpunkt enthält das ISTQB® Certified Tester-Portfolio nur zwei der „großen fünf“ Spezialmodule im Bereich Specialist (Sicherheit und Testautomatisierung).

3.2 Professionelle Tester brauchen mehr Auswahlmöglichkeiten

Mit der zunehmenden Anzahl an Möglichkeiten zur Spezialisierung in der Aus- und Weiterbildung erweitern sich auch die möglichen Karrierewege für Software-Tester um ein Vielfaches. >>

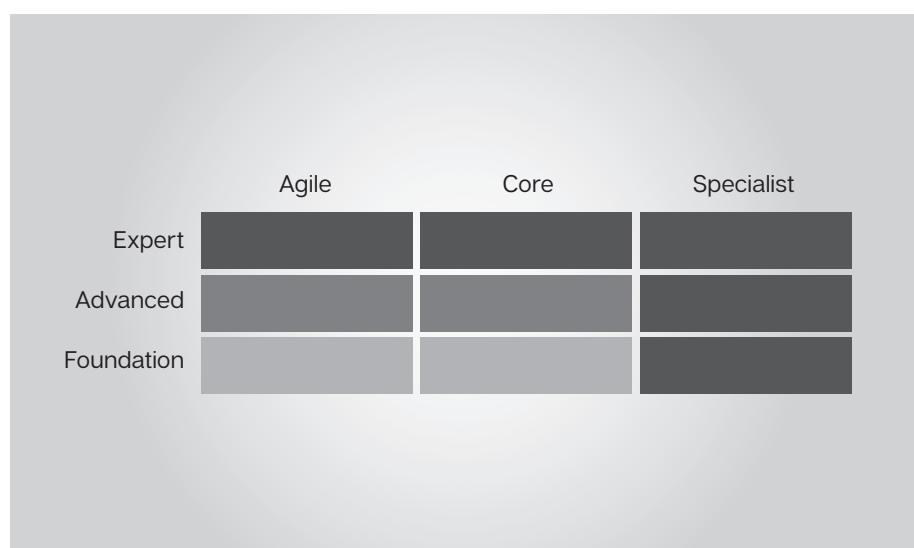


Abb. 3: Aufbau des ISTQB® Certified Tester-Portfolios

Die derzeitige Praxis definierter Vorbedingungen (z.B. dass Tester sich nur dann für ein bestimmtes Spezialthema X entscheiden können, wenn sie in Thema Y zertifiziert sind) hat sich für Organisationen (wie z.B. das ISTQB®) bislang gut bewährt. Für die Spezialthemen werden jedoch mehr Wahlmöglichkeiten, mehr Flexibilität und ein weniger restriktives Aus- und Weiterbildungsschema benötigt. Der Erwerb spezialisierter Testkenntnisse sollte für Fachleute aus der gesamten IT-Branche so offen wie nur möglich gestaltet sein. Diese Leute wollen einfach nicht, dass ihnen durch unflexible Vorbedingungen eine zusätzliche Last aufgebürdet wird.

3.3 Karrierewege in der IT-Industrie müssen synchronisiert werden

Organisationen wie das ISTQB® haben maßgeblich dazu beigetragen, einen Karriereweg für Software-Tester zu definieren. Parallel dazu wurden in der IT-Industrie von Fachorganisationen andere Karrierewege entwickelt, wie z.B. im Requirements Engineering und im Usability-Design-/Engineering.

Es ist nunmehr ausgesprochen wichtig, die Synergien, die zwischen einzelnen Karrierewegen in der IT bestehen, zu identifizieren und es Spezialisten zu ermöglichen, gegenseitig von den Trainingsangeboten zu profitieren. Beispielsweise sollte das ISTQB® dafür sorgen, dass professionelle Usability-Designer den Usability Tester-Kurs des ISTQB® belegen können, ohne dass ihnen vorgeschrieben wird, dass sie dafür einen Karriereweg im Software-Testen einschlagen müssen. Dies bedeutet, dass zwischen verschiedenen Karrierewegen in der IT Synchronisationspunkte bestimmt werden müssen, und es eine Politik der offenen Tür gibt, die sicherstellt, dass die jeweiligen Aus- und Weiterbildungssangebote gegenseitig leicht zugänglich sind. Geschieht dies nicht, dann wird das die Entwicklung von spezialisierten Fähigkeiten in der IT-Industrie insgesamt behindern.

3.4 Eine stärkere Gewichtung von praktischen Fähigkeiten ist erforderlich

Spezialisierte Fähigkeiten können nur voll entwickelt werden, wenn die erworbenen theoretischen Grundlagen anschließend in realistischen Situationen angewendet werden. Es gibt ein reichhaltiges Angebot an Schulungen, die den Kursteilnehmern praktische „hands on“-Erfahrungen liefern. Organisationen wie das ISTQB® müssen für derartige Angebote offen sein und diese an das eigene Aus- und Weiterbildungsschema anpassen. Genau wie die oben erwähnte Synchronisation von Karrierewegen bedeutet auch dies eine kontrollierte Öffnung des bestehenden Schemas.

4. Schlussfolgerung

Im Laufe der letzten zehn bis 15 Jahre ist es gelungen, eine solide Grundlage von Testerfähigkeiten zu entwickeln. Um die Herausforderungen zu bewältigen, mit denen die IT-Industrie konfrontiert ist, werden professionelle Software-Tester in zunehmendem Maße breit gefächerte, spezialisierte Fähigkeiten benötigen. Wenn Schulungsorganisationen relevant bleiben wollen, dann werden sie nicht darum herumkommen, mehr spezialisierte Schulungsmodule anzubieten und ihre Tür für andere Karrierewege in der IT und für praktische Schulungen zu öffnen. |



KONTAKT ZUM AUTOR

Graham.Bath@t-systems.com

Das Qualitäts-Tripel: Design by Contract, Clean Code, Lean Testing

89

Prof. Dr. Andreas Spillner

– Professor für Informatik an der Hochschule Bremen, Mitglied im ASQF Beirat –

Prof. Dr. Ulrich Breymann

– ehem. Professor für Informatik an der Hochschule Bremen –



In den letzten Jahren gab es eine ganze Reihe von Maßnahmen und Vorschlägen, um die Qualität von Software zu erhöhen. Auch hat das Qualitätsbewusstsein der Entwickler¹ bedeutend zugenommen. Der Beitrag fokussiert auf den Bereich, für den diese Gruppe zuständig ist. Es werden drei Ansätze – daher Qualitäts-Tripel – vorgestellt und es wird gezeigt, wie deren Zusammenspiel zur Qualitätssteigerung beiträgt. Design by Contract, Clean Code und Lean Testing sind diese drei »Handwerkszeuge« für den Entwickler, die näher betrachtet werden.

Design by Contract (DbC)

Design by Contract hat zum Ziel, das reibungslose Zusammenspiel einzelner Programmmodulen (oder Klassen, Methoden, Funktionen) sicherzustellen. Um

dies zu erreichen, werden »Verträge« vereinbart, die festlegen, wie die Schnittstellen zu verwenden sind. Die Vereinbarungen gehen über die statische Definition (Signatur der Schnittstelle) hinaus. Entwickelt und eingeführt wurde das Vorgehen von Bertrand Meyer [Mey09].

Zu jeder Schnittstelle wird ein Vertrag geschlossen, der bei der Verwendung der Schnittstelle einzuhalten ist. Vertragsbestandteile sind:

- Vorbedingungen (pre-conditions) – Zusicherungen, die vom Aufrufer einzuhalten sind,
- Nachbedingungen (post-conditions) – Zusicherungen, die der Aufgerufene garantiert, und
- Invarianten (invariants) – Bedingungen, die durch den Aufruf unverändert bleiben. [>>](#)

¹ Geschlechtsbezogene Formen meinen hier und im Folgenden stets Frauen, Männer und alle anderen.



Im Vertrag ist festgelegt, wofür der Aufrufer – der Dienstnehmer – verantwortlich ist und mit welchen Ergebnissen er vom Dienstanbieter nach dem Aufruf rechnen kann. Der Dienstanbieter kann eine größere Komponente, aber auch nur eine einfache Funktion oder Methode sein. Vom Aufrufer sind die vereinbarten Vorbedingungen einzuhalten, der Dienstanbieter garantiert dann die Einhaltung der Nachbedingung und keine Änderung der Invarianten. Durch DbC werden die Verantwortlichkeiten bei der Nutzung von Schnittstellen auf beiden Seiten festgelegt.

DbC geht über TDD (Test-driven Development) hinaus. Bei TDD wird die Schnittstelle (wie z.B. Typ und Reihenfolge der Parameter) festlegt, Vor- und Nachbedingungen sowie Invarianten und damit die Verantwortlichkeiten werden nicht spezifiziert.

Clean Code (CC)

Clean Code steht für eine ganze Reihe von Prinzipien und Praktiken, die alle ein Hauptziel verfolgen: Der Programmtext soll einfach und intuitiv sein und mit wenig Aufwand in kurzer Zeit in eindeutiger Weise verstanden werden können. Er soll einfach strukturiert sein – z.B. keine tiefen Verschachtelungen – und keinerlei Programmtricks enthalten, da diese nur schwer nachvollziehbar sind. Geprägt wurde der Begriff von Robert C. Martin [Mar08].

Prinzipien und Praktiken sind beispielsweise (um nur einige wenige zu nennen):

- KISS – Keep it simple, stupid,
- DRY – Don't repeat yourself,
- Information Hiding,
- Versionskontrollsystem nutzen,
- Reviews und
- automatisierte Unit-Tests durchführen.

CC ist auch als mentale Einstellung der Programmierer zu sehen, die dafür sorgt, dass im Programmieralltag die Prinzipien und Praktiken bewusst angewendet werden (Clean Code Development Initiative für mehr Professionalität in der Software-Entwicklung).

Das Ergebnis bei Einhaltung der Clean-Code-Prinzipien und Praktiken ist ein leicht verständlicher Programmtext, der einfach zu ändern und gut zu testen ist.

Refactoring gehört ebenso zu den CC-Praktiken und hat in den letzten Jahren Einzug in die – nicht nur agile – Software-Entwicklung gehalten. CC steht für einen „ganzheitlichen Ansatz“, wo hingegen Refactoring anstrebt, den Programmtext einfacher und verständlicher zu gestalten, ohne seine Funktionalität zu verändern.



Glückwunsch ASQF

„FACHLICHER AUSTAUSCH, QUALIFIKATION UND PRODUKT-QUALITÄT SIND
IN UNSERER TÄGLICHEN ARBEIT IM BEREICH SOFTWARE-ENTWICKLUNG ALS
DYNAMISCHES ARBEITSFELD HOCHRELEVANT. DAHER ENGAGIEREN WIR UNS ALS
MITGLIED UND PROFITIEREN SOMIT KONTINUIERLICH VON DEN DURCH DIE
ASQF BEHANDELN THEMEN IN UNSERER GESCHÄFTSPRAXIS“

Carsten Menzel, rocket-media GmbH & Co KG

Lean Testing (LT)

„Lean Testing“ unterstützt Entwickler dabei, einen angemessenen und nicht zu aufwendigen Test durchzuführen. Beim Testen besteht folgendes Dilemma: Einerseits soll möglichst wenig getestet werden, um die Kosten niedrig zu halten, andererseits möglichst viel, um dem Ziel der Fehlerfreiheit (oder zumindest nur noch wenige unkritische Fehler im Programm zu haben) nahezukommen. Letztlich geht es darum, einen vernünftigen Kompromiss zwischen diesen beiden Extremen zu finden. Der aus dem agilen Umfeld bekannte Begriff „lean“ bedeutet hier, sich beim Testen auf das Wichtige zu konzentrieren, um diesen Kompromiss zu erreichen.

Manche Software-Entwickler meinen, sie müssten ein Programm „austesten“ oder gar nicht testen, im Vertrauen auf ihre sehr guten Programmierungsfähigkeiten. Jeff Langr schreibt beispielsweise in seinem Buch zu TDD „Using a testing technique, you would seek to exhaustively analyze the specification in question (and possibly the code) and devise tests that exhaustively cover the behavior.“ [Lan13]. Keine Motivation für Entwickler, sich mit dem Testen zu beschäftigen.

Lean Testing steht dagegen für einen Ansatz, der auf der einen Seite alle wichtigen Testfälle zur Prüfung der Software berücksichtigt, auf der anderen Seite aber den Testaufwand in einem überschaubaren Rahmen hält. Beim LT werden die Testfälle strukturiert und systematisch hergeleitet. Dadurch wird erreicht, dass keine wichtigen Testfälle übersehen werden. Gleichzeitig wird ein nachvollziehbares Kriterium an die Hand gegeben, wann der Test als ausreichend angesehen und somit beendet werden kann.

Das Vorgehen soll an einem Beispiel verdeutlicht werden. Angenommen eine Schnittstelle besteht aus zehn booleschen Parametern, die alle unabhängig voneinander mit den beiden Werten *true* und *false* belegt werden können. Es ergeben sich insgesamt $2^{10} = 1024$ Kombinationen. Müssen nun 1024 Testfälle durchgeführt werden? Nein, es sind einfach zu viele! Aber welche und wie viele Testfälle sollen ausgewählt werden? Eine Frage, die sich beim Testen immer wieder stellt! >>



Wird angenommen, dass nicht alle zehn Parameter beim Test frei kombiniert werden müssen, sondern dass eine Beschränkung, z.B. auf drei beliebige (aus den zehn möglichen) Parametern ausreicht, um Fehler aufzudecken, dann kann die Anzahl der benötigten Testfälle drastisch (auf 1,3 Prozent) verringert werden. Wenn drei (beliebige) Parameter kombiniert werden sollen, sind jeweils acht Kombinationen ($2^3 = 8$) mit Testfällen zu überprüfen. Nun ergeben sich insgesamt nur 13 Testfälle (s. Abbildung 1, drei 8-er Kombinationen sind farblich hervorgehoben. Aber auch bei jeder anderen Auswahl einer Kombination von drei beliebigen Parametern werden alle acht Kombinationen durch die 13 Testfälle überprüft, z.B. A,D,H oder C,G,J).

Zur Auswahl der Testfälle wurde ein Testverfahren zum kombinatorischen Testen verwendet, dessen Grundlage sogenannte Covering Arrays sind. Durch die systematische Herleitung der Testfälle kann deren Anzahl erheblich verringert werden und ein Kriterium zum Beenden der Tests ist ebenso gegeben – in diesem Fall wird die Ausführung der 13 Testfälle als ausreichend angesehen. Eine umfassende Darstellung der Herleitung der Testfälle sowie ein weiteres Beispiel zum Lean Testing ist in [Bre16] zu finden. Diverse weitere Lean-Testing-Ansätze werden in [Spi16] ausführlich beschrieben und jeweils an konkreten C++-Beispielen erläutert. Die LT-Ansätze decken alle für den Entwicklertest relevanten Testverfahren ab, die im aktuellen ISO-Standard zum Software-Test definiert sind [URL: ISO29119].



	A	B	C	D	E	F	G	H	I	J
1	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	0	1	0	0	0	0	1
4	1	0	1	1	0	1	0	1	0	0
5	1	0	0	0	1	1	1	0	0	0
6	0	1	1	0	0	1	0	0	1	0
7	0	0	1	0	1	0	1	1	1	0
8	1	1	0	1	0	0	1	0	1	0
9	0	0	0	1	1	1	0	0	1	1
10	0	0	1	1	0	0	1	0	0	1
11	0	1	0	1	1	0	0	1	0	0
12	1	0	0	0	0	0	0	1	1	1
13	0	1	0	0	0	1	1	1	0	1

Abb. 1: 3er-Kombinationen mit zehn booleschen Parametern (aus [Bre16])

Wie hängen DbC, CC und LT zusammen?

Im obigen Beispiel wurden Annahmen getroffen, die es ermöglichen, die geringe Anzahl an Testfällen als ausreichend anzusehen:

- Aus dem Vertrag zur Schnittstelle geht hervor, dass nur boolesche Werte übergeben werden. Ein Test auf ungültige Parameterwerte kann somit entfallen – hierfür ist der Aufrufer verantwortlich (und es muss dort entsprechend getestet werden). Es ist auch festgehalten, dass zwischen den zehn Parametern keine Abhängigkeiten existieren, an der Schnittstelle also alle Kombinationen als Eingaben möglich sind.
- Bei sauberer Programmierung wird davon ausgängen, dass wenn Fehler auftreten, diese nicht erst bei der vollständigen Kombination aller Parameter (1024 Testfälle) zum Tragen kommen, sondern bei einer geringeren Zahl – hier war die Annahme bei der vollständigen Kombinationen von jeweils drei (und nicht zehn) Parametern. Ein Code-Review kann Klarheit schaffen, ob diese Annahme berechtigt ist.

DbC & LT

Durch die im Vertrag festgelegten Zuständigkeiten lässt sich der Testaufwand unter Verwendung von Testverfahren angemessen – „lean“ – gestalten.

Beispiel: Wenn als Vorbedingung festgelegt ist, dass nur positive Zahlen als Parameter übergeben werden, dann kann der Test mit negativen Zahlen, Sonderzeichen usw. entfallen – jedenfalls an dieser Schnittstelle. Der Aufrufer muss sicherstellen, dass nur positive Zahlen weitergegeben werden. Dies muss dann beim Aufrufer getestet werden.

DbC & CC

DbC ist Voraussetzung für CC, da von den vereinbarten Vorbedingungen ausgegangen werden kann und somit die Funktionalität im Fokus der Umsetzung steht (Sicherheitsaspekte usw. sind dann vom Aufrufer zu gewährleisten). Verantwortlichkeiten sind klar geregelt und den jeweiligen Programmteilen zugeordnet.

CC & LT

Die Beachtung der Clean-Code-Prinzipien schafft eine wichtige Voraussetzung, den Test angemessen gestalten zu können. Erst durch eine einfache Programmstruktur ist eine einfache Testbarkeit gegeben. Die einfache Testbarkeit garantiert, dass der Test mit einfachen Testverfahren durchgeführt werden kann und damit „lean“ ist. Clean Code kann einfach getestet werden! Es kann in aller Regel von einem gleichartigen Verhalten bei gleichartigen Eingaben ausgegangen werden. Der Eingabebereich der einzelnen Parameter kann z.B. in Äquivalenzklassen aufgeteilt werden, um die Zahl der Testfälle gering zu halten (s. [Spi16], Kap. 4.3). [>>](#)

 *Tabelle 1: Auswertung der Umfrage 2015/16 zur Testintensität*

Die Testintensität ist übertrieben	... genau richtig	... zu gering
Tester	4,8 %	51,2 %	44,0 %
Entwickler	1,2 %	32,9 %	65,9 %
Gesamtergebnis (alle Rollen)	3,8 %	43,8 %	52,4 %

Fazit

Dass Entwickler mehr Systematik bei ihren Tests anstreben und mit der derzeitigen Situation eher unzufrieden sind, kann durch die Ergebnisse der aktuellen Umfrage „Softwaretest in Praxis und Forschung“ [URL: Softwaretest] belegt werden. Zur Frage „Wie schätzen Sie die Intensität des Testens ein?“ gab es folgende Antworten der befragten unterschiedlichen Rollen: siehe Tabelle 1

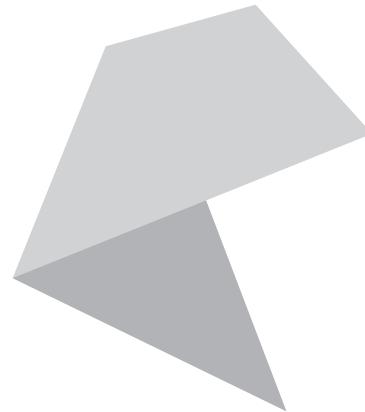
Knappe zwei Drittel der Entwickler sehen den Test als zu gering an. Sie sind also verunsichert, was die Testintensität betrifft. Im Gegensatz dazu sehen die

Hälften der Tester die Testintensität als genau richtig an. Vermutlich liegt es daran, dass Tester die passenden Testverfahren kennen, diese zur Testfallerstellung heranziehen und damit gut einschätzen können, wann ausreichend genug getestet wurde.

Design by Contract, Clean Code und Lean Testing sind jede für sich genommen bereits Verbesserungen im Bezug auf die Qualität der zu erstellenden Software. Ein großer Schritt in Richtung Qualität gelingt, wenn alle drei umgesetzt werden – also das Tripel erreicht ist! |

Literatur

- [Bre16] Ulrich Breymann, Andreas Spillner: Lean Testing, OBJEKTspektrum Online Themenspecial Testing, Ausgabe September 2016 (s.u. <http://www.sigs-datacom.de/fachzeitschriften/objektspektrum/archiv/artikelansicht/artikel-titel/lean-testing.html>)
- [Lan13] Jeff Langr: Modern C++ Programming with Test-Driven Development, O'Reilly, 2013
- [Mar08] Robert C. Martin: Clean Code - A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008
- [Mey09] Bertrand Meyer, Touch of Class, Springer Verlag, Berlin, 2009
- [Spi16] Andreas Spillner, Ulrich Breymann: Lean Testing für C++-Programmierer, dpunkt.verlag, Heidelberg, 2016 (siehe auch <http://www.leantesting.de/>)
- [URL: ISO29119] ISO/IEC/IEEE 29119-4: Software Testing, Part 4: Test Techniques (s.u. <http://www.softwaretestingstandard.org/part4.php>)
- [URL: Softwaretest] Ergebnisse der Umfrage »Softwaretest in Praxis und Forschung« (s.u. <http://softwaretest-umfrage.de/>)



KONTAKT ZU DEN AUTOREN

*Andreas.Spillner@hs-bremen.de
Ulrich.Breymann@hs-bremen.de*

Gemeinsam sind wir stärker:

GTB und ASQF gestern, heute und morgen

Dr. Frank Simon

– Mitglied der BLUECARAT AG Geschäftsführung, Leiter des BITKOM-Lenkungsausschusses Software und Vorstand des German Testing Boards –



Der Mensch und durch ihn gegründete Organisationen sind schon wundersame Gebilde: Drohen unruhige Zeiten, so tut man sich ohne weiteres zusammen, ignoriert gegebenenfalls existierende Unterschiede und versucht gemeinsam, Wege aus der Unruhe zu finden. Wird es dann aber wieder ruhiger und macht sich in Summe eine gewisse Sättigung in der Zusammenarbeit breit, dann fokussiert man sich gerne wieder auf die jeweils eigenen Distinktionsmerkmale. Die Zusammenschlüsse fragmentieren sich wieder und jeder geht seine eigenen Wege, ohne dem anderes etwas Böses zu wollen.

Schwierige Zeiten am Anfang des Jahrtausends

So war es z.B. Anfang dieses Jahrtausends: Das Testen wurde für die zunehmend größer und wichtiger werdenden IT-Systeme immer existentieller. Und trotzdem war das Testen noch eine Individualfähigkeit, meist mit einem eigenen Wording, einem eigenen Vorgehen und eigenen Ergebnissen. Aber die Zeiten wurden unruhiger, der Markt verlangte dringend etwas, um das Testen und die damit beschäftigten Menschen zu professionalisieren, damit alle das-selbe unter dem Begriff Testen verstanden, dieselben Möglichkeiten des Testens kannten und ähnliche Vor-

gehen für ein professionelles Testen vorschlugen. Aus dieser Grundstimmung heraus wurde 2001 eine ASQF-Arbeitsgruppe mit dem Ziel gegründet, ein firmenunabhängiges Trainingsschema zum Thema Software-Testen für den deutschen Markt zu schaffen. Dieser neuen Arbeitsgruppe haben sich schnell einige Experten angeschlossen. Dazu gesellte sich – wie gesagt, die Zeiten waren unruhig – noch die Gesellschaft für Informatik mit ihrer Arbeitsgruppe TAV (Test-Automatisierung und Verifikation), die ein Glossar in der ersten Version zur Verfügung stellte. Weitere Sympathisanten fanden sich schnell unter den fränkischen Mittelständlern, denn damals stand das „F“ im ASQF noch für Franken. Diese Gruppe von Testexperten etablierte dann das Certified-Tester-Schema in Deutschland. Die ersten Trainings fanden dann auch als Joint Venture der Provider aus der Arbeitsgruppe unter dem Dach des ASQF statt. Die Prüfungen nahm der ASQF selbst ab. 2002 wurde dann das German Testing Board gegründet – immer noch als Organisation des ASQF – um die Internationalisierung in einem entstehenden International Software Testing Qualifications Board (kurz ISTQB®) zu etablieren. Dieses Schema war von Anfang an sehr erfolgreich. Die folgenden ruhigeren Jahre führten dann dazu, dass sich 2005 das „erwachsene“ German Testing Board ausgründet hat. >>

Ruhigere Zeiten: Die Erfolgsstory rund um das GTB und das ISTQB®

In den Folgejahren hat sich das GTB prächtig entwickelt, relativ losgelöst vom ASQF (siehe ruhigere Zeiten). Der Advanced Level wurde mit seinen drei Ausprägungen „Test Manager“, „Test Analyst“ und „Technical Tester“ ausgerollt und fand in Deutschland regen Zuspruch. Vor einigen Jahren wurde dann auch der bereits lange angekündigte Expert-Level operativ ins Leben gerufen. Es gehört seitdem ebenfalls zum etablierten Zertifizierungskatalog.

Immer mehr Trainingsanbieter begannen, eigene Certified-Tester-Kurse anzubieten, nicht ohne jeweils sehr sorgfältig vom GTB überprüft zu werden. Denn dafür steht das GTB seit Jahren: für die Entwicklung von Trainingsschemata von höchster Qualität und die Sicherstellung, dass sämtliche Trainingsmaterialien der Trainingsanbieter das Trainingsschema effektiv abdecken und didaktisch erfolgsversprechend den Schulungsteilnehmern vermitteln. Das GTB hat hierfür zügig effiziente Prozesse zur Absicherung der Qualität im Markt mit Akkreditierungsprozessen für Trainingsprovider und Prozessen der Qualitätskontrolle für die Zertifizierungen herausgegeben, die direkt fruchten und heute international ihresgleichen suchen.

Die Aufgaben des GTB umfassen aber nicht nur das Curriculum als Kern dieser Schemata. Auch das Thema der Prüfungsfragen ist entscheidend für die Qualität und den Erfolg eines Schemas. Für eine saubere Aufgabenteilung (Segregation of Duties, SoD) wurde die Zertifizierung schnell in entsprechend professionelle Hände gelegt. So greift das GTB heute sehr erfolgreich auf drei Zertifizierungsstellen zurück. Sie sind für das German Test-

ting Board ein wesentlicher Bestandteil des Ökosystems Certified Tester in Deutschland.

So kann das Certified-Tester-Schema heute weltweit auf fast 500.000 herausgegebene Zertifikate schauen, davon fast 50.000 alleine in Deutschland. Es ist damit heute das weltweit dritt wichtigste unabhängige IT-Zertifikat! Das alles aus der ursprünglichen Erkenntnis heraus, dass es in schwierigen Zeiten sinnvoll ist, zu kooperieren, um gemeinsam mehr in weniger Zeit zu schaffen. Herausgekommen ist heute ein de-facto-Standard im Testen, auf den alle mit Stolz schauen sollten.

Die Zeiten werden wieder rauher

Ende gut, alles gut – könnte man meinen. Aber die Entwicklung der IT und der sie erzeugenden und modifizierenden Prozesse, Methoden und Werkzeuge ging und geht aktuell mit enormer Geschwindigkeit weiter. Einige Beispiele mit einem besonderen Bezug zum Testen sind:

- Neue Vorgehen brachten agile Prinzipien in die Projekte, in denen ein sehr viel restriktiveres Time-To-Market-Denken herrschte, in denen die einzelnen Teams sehr viel autonomer handeln konnten und in denen häufig kein Platz für dedizierte Nur-Tester war. Hier waren neue Konzepte gefragt, wie die Testdisziplin Einzug in agile Teams halten könnte.
- Neue Programmiersprachen brachten neue Konzepte, die ein sehr frühes Testen ermöglichten. Dieses White-Box-Testen mittels eigener Frameworks (X-Unit) erforderte zusätzliche Skills vom Tester.

- Neue Architekturkonzepte brachten neue Testobjekte hervor, die vollständig ohne GUI auskommen und die nur Services über entsprechende Formalismen (WSDL, SOAP, etc.) anboten. Deren Test war ohne detaillierte Technikkenntnisse und Testwerkzeuge nicht möglich, was wiederum zusätzliche Testskills erzeugte.
- Es kamen modellbasierte Software-Entwicklungsmethoden auf, denen sehr schnell methodisch ähnliche, modellbasierte Testmethoden beigestellten wurden. Auch dieses Wissen musste aufbereitet, standardisiert und zertifizierbar aufbereitet werden.
- Die Funktionalität als Kern-Qualitätsattribut wird immer weiter durch andere Qualitätsattribute abgelöst. So rücken heute die Usability, die Performance und die Sicherheit immer weiter in den Vordergrund und fragen nach erweiterten/alternativen Testkonzepten.
- Die Darreichungsformen von Weiterbildungsan geboten haben sich in den letzten Jahren ebenso gewandelt. Effiziente Konzepte wie E-Learning oder Blended Learning finden Einzug in unsere Industriewelt, nicht exklusiv, aber so doch in einer gesunden Mischung (blended) mit klassischen Angeboten. Dies hat natürlich auch Auswirkungen auf das Material, das die Trainingsanbieter nutzen und damit auf das GTB, das auch dieses Material qualitätssichern muss.

Wenn die Zeiten unruhiger werden, beginnen zugleich wieder die Kooperationen, um gemeinsam diese Zeit zu meistern. So auch das GTB:

- So hat das German Testing Board in einer Kooperation mit dem Usability und User Experience Qualification Board (UXQB) in 2015 sehr erfolgreich den ersten Usability-Testing-Kurs erarbeitet, der in 2016 durch die Alpha und Beta-Reviews ging und seit Ende 2016 in Deutschland final ausgerollt wird. Für 2017 steht dann der internationale Roll-Out an. Das UXQB ist übrigens ein Board, welches in enger Kooperation mit dem ASQF und dem International Software Quality Institute (iSQI) steht, dessen Gesellschafter der ASQF ist.
- Ganz aktuell hat das GTB mit dem ASQF eine Kooperationsvereinbarung geschlossen, mit dem Ziel, einen Lehrplan für das Thema Quality Engineering für das Internet der Dinge zu entwickeln und auf dem deutschsprachigen Markt zu etablieren. Auch hier geht es um dieselben Ziele wie bei dem Certified-Tester-Schema. Der Markt ist – wieder – verunsichert ob des Themas IoT. Es gibt keine einheitliche Sichtweise auf dieses Thema, erst recht fehlt eine einheitliche Sprechweise. Und es fehlt vor allem eine Basis an Wissen, mit dem die Teams in dieses Thema einsteigen können. Dies ist gerade beim Thema IoT so besonders dramatisch, da es hier auch um cyberphysische Systeme geht, die aufgrund der Möglichkeit, physischen Schaden anzurichten, besonderen Qualitätsansprüchen genügen müssen. >>

Und ein „es wird schon wieder ruhiger“ ist nicht abzusehen. Aktuell arbeitet das GTB aktiv an einer Vielzahl weiterer Lehrpläne: hier sind z.B. Performance-Testing, Testdatenmanagement und Security-Testen zu nennen. Diese neuen Herausforderungen treffen alle Organisationen, die sich dem Thema Qualität und Testen verschrieben haben. Von daher scheinen aktuell wieder alle Zeichen auf Kooperation zu stehen. Das GTB hat dies erkannt und kooperiert zunehmend mit Experten externer Organisationen. Insbesondere verbindet dies das German Testing Board und den ASQF, da beide Gruppen an exakt den gleichen Zielen arbeiten: Das Thema der Qualität für Software und Systeme zu etablieren und dafür zu sensibilisieren. Vieles wurde in den letzten Jahrzehnten bereits erreicht. Jedoch, die Zeiten sind aktuell unruhig und so ist es gut, mittels tatkräftiger Kooperationen gemeinsam Lösungen zu erarbeiten. Es gilt, Antworten zu liefern und handfeste Werkzeuge bereitzustellen, um den Anforderungen an die Kompetenz der IT-Teams auch in Zukunft gerecht zu werden. Dies ist ein wichtiger Faktor für den Wirtschaftsstandort Deutschland. Das German Testing Board freut sich daher weiterhin auf viele nachhaltige und fruchtbare Kooperationen, insbesondere natürlich mit dem ASQF. |



KONTAKT ZUM AUTOR

Frank.Simon@german-testing-board.info

Impressum

HERAUSGEBER

ASQF e.V.
Friedrich-Engels-Str. 24, 14473 Potsdam
Tel +49 331 231810-29
Fax +49 331 231810-10
info@asqf.de, www.asqf.de

November 2016

REDAKTION

V.i.S.d.P.:
Stephan Goericke (ASQF-Hauptgeschäftsführer)

Christin Senftleben, Anja Schreinert, Isabel von Gustedt, Julia Schirmer, Dr. Armin Metzger

GESTALTUNG

Frenkelson Werbeagentur, Potsdam
www.frenkelson.de

Dieses Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtes bedarf der Genehmigung durch den Herausgeber.

ISSN 2367-3516

Wir danken allen Autoren ganz herzlich für ihre Unterstützung und Mitwirkung an dieser Sonderedition. Hierfür konnten wir allesamt hochkarätige Persönlichkeiten gewinnen, die mit ihrem Erfahrungsschatz zum Gelingen dieser Ausgabe beigetragen haben. Ohne ihr Wissen und Engagement hätte sie nicht entstehen können. Ein besonderer Dank gilt zugleich unseren Partnern für ihre Unterstützung und Kooperation.

Unter der Verwendung von Grafiken und Bildern von:
©shutterstock.com: 3DDock, a-image, agsandrew, alice-photo, Anneka, cgi_4_you, Dmitry Kalinovsky, ESB, JKII4, Leigh Prather, Lisa S., Liu zishan, MarcelClemens, Matej Kastelic, Roman Tsubin, Sashkin, SFIO CRACHO, Svetlana Lukienko, tsyhun, TypoArt BS, vs148, optimarc / ©Unsplash_pixabay.com / ©Lom_Fotolia.com
Foto Prof. Dr. Ina Schieferdecker: Matthias Heyde/ Fraunhofer FOKUS / Foto Stephan Goericke: Karoline Wolf



bedankt sich bei den Partnern ...



Exam Bodies:



Danke!

Software. Testing. Excellence.

FUNDAMENTALS OF TESTING

Für Einsteiger und Profis:
ISTQB® Certification Product Portfolio

GTB Premium Partner

alTRAN

BERNER & MATTNER
AN ASSISTAN COMPANY

CGI



Cognizant
SetCon



D&H
Diaz & Hilterschmid



EXCO



ISARTALakademie

knowledge
department

loyal team
for software quality

Lysant

methodpark



sepp.med
Qualität sichert Erfolg

software
quality lab

SOGETI

SRC
security consulting

T-Systems

T-Systems
T-Systems Multimedia Solutions GmbH