

---

Fachausschuss  
**Management der  
Anwendungsentwicklung  
und -wartung (WI-MAW)**  
im FB Wirtschaftsinformatik

Jahrgang 24 Heft 1  
ISSN 1610-5753

April 2018

---

## Inhalt

|                     |    |
|---------------------|----|
| Fachbeiträge .....  | 3  |
| Ankündigungen ..... | 63 |
| Berichte .....      | 67 |
| Organisation .....  | 76 |

## Inhaltsverzeichnis

### Fachbeiträge

|  |    |
|--|----|
| <i>Agile und DevOps treffen auf Regulatorik – Konfrontation oder Kooperation?</i><br>Jens Borchers.....  | 3  |
| <i>Vielfalt in der IT-Projektarbeit – Chancen erkennen und nutzen</i><br>Dorothee Feldmüller .....   | 13 |
| <i>Agile Qualitätssicherung von Big Data Projekten - Bedarf an integrierten Test- und Bewertungsansätzen</i><br>Andreas Schmietendorf, Jan Hentschel ..... | 24 |
| <i>Automating Micro-Service Testing</i><br>Harry M. Sneed, Chris Verhoef .....   | 33 |
| <i>Business Rule Recovery as a Fata Morgana</i><br>Harry M. Sneed .....  | 43 |
| <i>Six-Sigma-getriebene Optimierung des Applikationsmonitorings</i><br>Stephan Tost.....   | 48 |

### Ankündigungen

|   |    |
|---|----|
| Call for Papers / Aufruf zur Einreichung von Beiträgen PVM 2018 (Projektmanagement und Vorgehensmodelle), 15. u. 16. Okt 2018 – Düsseldorf .....  | 63 |
| Buchankündigung: Endstation Wien – Das Ende einer langen Reise durch die deutschsprachige IT-Welt, Gesammelte Erfahrungen von Harry M. Sneed..... | 66 |

### Berichte

|   |    |
|---|----|
| Projektmanagement und Vorgehensmodelle 2017 (PVM217): Die Spannungen zwischen den Menschen und den Prozessen im Projekt (Bericht von Martin Engstler).....  | 67 |
| Workshop “Evaluation of Service-APIs – ESAPI 2017“ Service APIs als Enabler einer erfolgreichen Digitalisierung (detaillierter Bericht von Michael Binzen, Konrad Nadobny, Hendrik Neumann, André Nitze, Andreas Schmietendorf) ..... | 71 |

### Organisation

|   |    |
|---|----|
| Der Fachausschuß „Management der Anwendungsentwicklung und –wartung“ WI-MAW und die Fachgruppen<br>Vorgehensmodelle für die betriebliche Anwendungsentwicklung WI-VM<br>Projektmanagement WI-PM<br>Software Produktmanagement WI-PrdM<br>stellen sich vor ..... | 76 |
|---|----|



# Agile und DevOps treffen auf Regulatorik – Konfrontation oder Kooperation?

Jens Borchers

jens.borchers@soprasteria.com / jensborchers@acm.org

*Sopra Steria Consulting, Hamburg*

**Abstract:** In den letzten 10 Jahren haben sich in vielen Firmen die klassischen Softwareentwicklungsmodelle in Richtung „Agiles Vorgehen“ entwickelt, bei dem im Gegensatz zum Wasserfall-Modell die Softwaresysteme – gerade solche mit nicht vorab voll bestimmbareren oder volatilen Anforderungen – in hochfrequenten Zyklen entwickelt und in Produktion gebracht werden. Prominentester Vertreter ist dabei das Scrum-Projektmanagementmodell mit seinen sog. Sprints. Um die Time-to-Market weiter zu reduzieren, wurde der agile Ansatz in letzter Zeit durch das sogenannte DevOps-Konzept bis zum IT-Produktionsbetrieb verlängert, u.a. soll dabei auch die Inbetriebnahme von Software häufiger und in einer möglichst kurzen Zeit, teilweise in wenigen Minuten oder gar Sekunden, erfolgen. Auf der anderen Seite stehen in vielen Bereichen, insbesondere auch in der Finanzindustrie, interne und externe regulatorische Anforderungen, die sehr rigide organisatorische und prozessuale Vorgaben für eine risikominimierende Softwareentwicklung und IT-Produktion umfassen. Dieser Beitrag beschreibt das Spannungsfeld zwischen diesen zumindest auf den ersten Blick divergierenden Sichtweisen.

Dieser Beitrag ist wie folgt aufgebaut:

1. Agile Software-Entwicklung und DevOps-Konzepte im Überblick
2. Die regulierte Welt
3. IT-Regulatorik im Bankenbereich und ihre Umsetzung
4. Die wesentlichen Reibungspunkte zwischen Agilität und Regulatorik
5. Fazit und Vorschlag zum Vorgehen

## 1 Agile Software-Entwicklung und DevOps-Konzepte

Dieser Beitrag hat nicht das Ziel, die Ziele und Ansätze agiler Software-Entwicklung und der später ergänzten DevOps-Konzepte im Einzelnen darzustellen. Dazu wird auf die mittlerweile umfangreiche Literatur zu diesen Themenkreisen verwiesen. Hier sollen aus Sicht des Autors nur die wichtigsten Konzepte, insbesondere im Hinblick auf die mögliche Konfrontation mit der Regulierung behandelt werden.

### 1.1 Agile Software-Entwicklung

Agile Konzepte haben sich seit dem im Jahr 2001 veröffentlichten „Manifesto for Agile Software Development“ [5], welches seinerzeit von 17 führenden Software-„Gurus“ unterzeichnet wurde, immer weiter durchgesetzt, wobei sich vor allem die Projektmanagementmethode „Scrum“ als wesentlicher Ansatz etabliert hat. Häufig wird Scrum aber fälschlich als Vorgehensmodell zur Herstellung von Software interpretiert, der Unterschied zu z.B. „Extreme Programming“ als einer möglichen agilen Entwicklungsmethodik wird nicht immer klar. Im Übrigen ist der agile Ansatz nicht so neu, wie er scheint: Schon 1991 hat James Martin sein Buch zu „Rapid Application Development“ (RAD) veröffentlicht [6], welches im Kern auf dem iterativen Spiralmodell von Barry Boehm aufsetzte. Laut einer aktuellen Umfrage (VersionOne, 2016) nutzen schon 95% aller Unternehmen „in irgendeiner Form“ agile Prozesse. Nach-

dem sich die agilen Ansätze zunächst bei Firmen im Internetbereich verbreitet haben, versuchen seit ca. 2010 auch immer mehr traditionelle Branchen, wie eben auch Banken, durch diese Ansätze die klassischen Vorgehens- und Projektmanagement-Modelle aus den 1980er- und 90er-Jahren zu ersetzen, die sich in der Praxis häufig als zu schwerfällig und bürokratisch erwiesen haben und nicht immer zum gewünschten Ziel führen.

Eigentlicher Nutznießer der agilen Konzepte sind die Anforderer aus den Fachbereichen (in Scrum-Projekten vertreten durch den „Product Owner“), die sehr schnell auf schon umgesetzte Funktionen reagieren können und nicht mehr gezwungen sind, ein vollständiges Fachkonzept zu liefern, bevor die eigentliche Entwicklung beginnt. Für die Entwickler steht im Vordergrund, dass sie in überschaubaren Einheiten weitere Funktionalität umsetzen können, die schnell sichtbare Ergebnisse zeigt. Allerdings darf nicht vergessen werden, dass es auch bei einem agilen Vorgehen vorab definierter und möglichst abgesicherter Architekturvorgaben und auch eines (kleinen) Basis-Fachkonzepts („Frontloading“) bedarf.

## 1.2 DevOps

Auch wenn in vielen Unternehmen bereits agil entwickelt wird, so bleibt doch die strikte organisatorische und technische Trennung von Entwicklung und Wartung („Change the Business“) einerseits und dem IT-Produktionsbetrieb („Run the Business“) weitgehend bestehen. Diese basiert primär auf dem seit Anfang der 1990er Jahre in Großbritannien entwickelten Modell für das sog. „IT Infrastructure Management“ (ITSM), das unter dem Begriff „ITIL“ (IT Infrastructure Library) seit der Mitte der 2000er-Jahre als Framework (und als Basis zur Zertifizierung von IT-Mitarbeitern) populär wurde und seit 2005 als „ISO 20000“ auch als Standard für Organisationen zur Verfügung steht. ITIL bildet seitdem auch die Basis vieler regulatorischer Vorgaben für den Übergang von Software-Artefakten aus der Entwicklungs- in die Produktionsdomäne.

Um den mit den agilen Entwicklungssätzen verbundenen Anspruch an die Änderungsgeschwindigkeit nicht an der Hürde „Übergang in die Produktion“ enden zu lassen, haben

Patrick Debois und Andrew Shafer erstmals 2007 (siehe dazu [7]) einen dynamischen Übergang von Software aus der Entwicklung in die Produktion propagiert, dieser wurde 2009 durch John Allspaw und Paul Hammon in „10+ Deploys a Day: Dev and Ops Cooperation at Flickr“ [1] bekannt gemacht.

Das Hauptziel von DevOps ist eigentlich, die durch das ITIL-Modell verstärkte „mentale Trennung“ von Entwicklung und Betrieb wieder zu reduzieren. Während die Entwickler alle ihre Neuerungen so schnell wie möglich auch in Betrieb nehmen wollen, geht es diesem – nicht ganz zu Unrecht – im Wesentlichen um einen stabilen und reibungslosen Ablauf der IT-Produktion. Auf der anderen Seite werden immer noch – auch in agilen Projekten – die späteren Betriebsaspekte viel zu spät mit in die Entwicklungsvorgaben aufgenommen, was dann spätestens beim ersten Release zu unnötigen Diskussionen und Nacharbeiten führt.

In vielen Unternehmen wird DevOps derzeit nur mit dem Anspruch eingeführt, im Sinne eines „Continuous Deployments“ die technische Strecke zwischen dem Freigeben einer Änderung durch den Entwickler bis zur Aktivierung in der IT-Produktion so zu optimieren, dass diese einer minimalen Zeit erfolgen kann. Die Verbreitung des Vortrags „Amazon Deploys to Production Every 11.6 Seconds“ auf der Velocity 2011 [9] hat auch bei anderen Unternehmen die Motivation geschürt, ihre Produktionsübergabe-Prozesse vom „bürokratisch“ empfundenen „Request for Change“-Verfahren aus ITIL durch flexiblere und vor allem schnellere Abläufe zu ersetzen.

Was allerdings für ein Start-up-Unternehmen in einem nicht regulierten Bereich an DevOps-Konzepten direkt einsetzbar sein mag, ist z.B. bei einem Finanzdienstleister nur bedingt und

auch nur mit erweiterten Rahmenbedingungen möglich. Dieses wird in den folgenden Kapiteln dargestellt.

### 1.3 Agile und DevOps im Zusammenspiel

Die folgende Grafik zeigt plakativ die signifikanten Änderungen der Software-Entwicklung und der Produktionseinführung. Während der agile Ansatz bildlich primär das Ziel hat, die Mauern zwischen Fachbereich und Entwicklung zu öffnen, hat der DevOps-Ansatz dieses Ziel in Bezug auf die – in der Praxis noch viel höhere Mauer – zwischen Entwicklung und IT-Produktion.

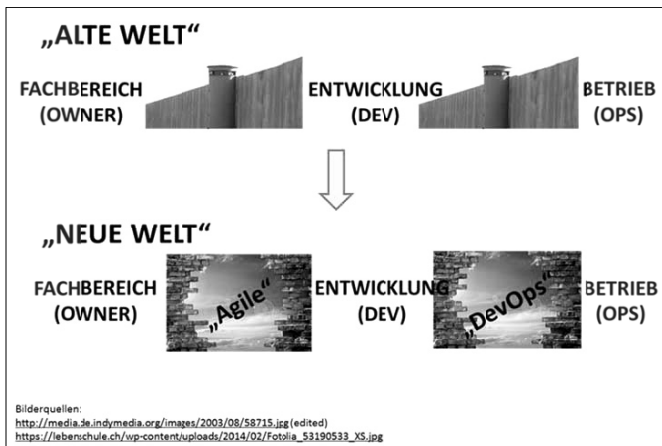


Abbildung 1 - Darstellung der grundlegenden Ziele der agilen und DevOps-Ansätze

## 2 Die regulierte Welt

In vielen Bereichen unterliegt die Entwicklung von Software detaillierten und strengen gesetzlichen und weiteren Regularien wie Normen, de-facto Standards, Prüfungsstandards usw.

In diesem Beitrag soll nur ein kurzer Überblick der einzelnen Industrien gegeben werden, in denen Regulatorik in mehr oder weniger ausgeprägter Form die Entwicklung, Qualitätssicherung und Inbetriebnahme von Softwaresystemen mitbestimmt.

- Medizintechnik (Medizin-Produkte-Gesetz, MPG)

Soweit Software als Teil eines Medizinprodukts, z.B. als Embedded Software eines Geräts eingesetzt wird, unterliegt sie dem MPG, welches als Umsetzung einer entsprechenden EU-Richtlinie 90/385/EWG in Deutschland und Österreich weitreichende Regeln für deren Qualitätssicherung und den Einsatz vorgibt.

Auch Software als eigenständiges Medizinprodukt nimmt immer mehr Raum ein (z.B. für Diagnose-Systeme, KI-Anwendungen) und unterliegt denselben Regulierungen.

- Flugzeugindustrie

Soweit Software in Fluggeräten eingesetzt wird, unterliegt ihre Herstellung und Wartung den weltweit gültigen Richtlinien „DO-178B, DO-178C: Software Considerations in Airborne Systems and Equipment Certification“.

- Automobilindustrie

In der Automobilindustrie gilt für Softwaresysteme in Fahrzeugen z.B. die „ISO 26262 - Sicherheitsrelevante elektronische Systeme in Kraftfahrzeugen“. Hier kommen zunehmend nicht-technische Regelungen, z.B. in Bezug auf die zunehmende Bedeutung für „autonomes Fahren“ hinzu.

- Militärischer Bereich

Der militärische Bereich hat – vor allem in den USA – schon frühzeitig Regularien für die Entwicklung und Wartung von Softwaresystemen entwickelt, hier seien insbesondere der bekannte und schon 1988 veröffentlichte DoD-Standard DOD-STD-2167A und sein Nachfolger MIL-STD-498 genannt. Diese sind mittlerweile durch eine große Anzahl von Einzelregelungen ersetzt worden. Daneben gehörte das amerikanische Militär zu den wesentlichen Sponsoren von CMMI.

- Banken und Finanzdienstleister

Seit Mitte der 2000er Jahre hat eine immer weiter zunehmende Regulierung der Banken und Finanzdienstleister sowohl in Europa als auch in den USA eingesetzt, die in Deutschland erstmalig 2005 durch die sog. „MaRisk“ (Mindestanforderungen an das Risikomanagement) [2], herausgegeben von der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) veröffentlicht wurde. Nach der Bankenkrise wurden die Regularien sukzessive weiter verschärft und auch in Bezug auf Vorgaben für die Entwicklung und den Betrieb von Softwaresystemen bei Banken immer weiter konkretisiert. Im November 2017 wurden die „Bankaufsichtlichen Anforderungen an die IT“ (BAIT) [3] als Konkretisierung zur Umsetzung der MaRisk veröffentlicht und stellen seitdem den rechtlichen Rahmen für die Entwicklung und Betrieb von Software bei Banken und im Finanzdienstleistungsumfeld dar. Im Folgenden werden die Anforderungen an Banken weiter detailliert dargestellt.

### 3 IT-Regulatorik im Bankenbereich und ihre Umsetzung

Ohne hier in diesem Beitrag auch nur annähernd auf die vielfältigen Details der Regulierung von Banken auch im IT-Bereich eingehen zu können, sollen hier die wesentlichen rechtlichen Rahmenbedingungen, unter denen heute bei Banken Software entwickelt (schließt immer Wartung ein) und produktiv betrieben werden kann, dargestellt werden.

Die wesentlichen Vorgaben für die IT-Organisation in Banken basieren heute im Wesentlichen auf den vier folgenden Säulen:

- Die gesetzlichen Regelungen aus dem Handelsgesetzbuch (HGB), Kreditwesengesetz (KWG) und anderen Gesetzen und Verordnungen wie z.B. aus dem BMF die GoBD („Grundsätze zur ordnungsgemäßen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff“), weiterhin natürlich insbesondere die Datenschutzgesetze (ab Mai EU-DSGVO);
- die von der eigenen Revision vorgegebenen Regularien, die allerdings zu einem Großteil die Konkretisierung der von außen vorgegebenen Gesetze und Verordnungen darstellen;
- die von den externen Wirtschaftsprüfern genutzten Prüfungsstandards, in Deutschland insbesondere die vom Institut der Wirtschaftsprüfer (IDW) herausgegebenen IT-Prüfungsstandards PS330, PS850 und weitere;
- last but not least die von der EZB und BaFin vorgegebenen Regelwerke, für die IT im Wesentlichen seit 2005 die MaRisk, die seitdem mehrfach (zuletzt Oktober 2017) fortgeschrieben worden ist und seit November 2017 durch die BaFin in den BAIT (Bankauf-

sichtliche Anforderungen an die IT) konkretisiert wurde. Diese Regelungen konkretisieren dabei §25a des KWG und weiterer Vorgaben wie Basel II und III.

Die BAIT behandeln dabei 8 Themenbereiche:

- IT-Strategie
- IT-Governance
- Informationsrisikomanagement
- Informationssicherheitsmanagement
- Benutzerberechtigungsmanagement
- **IT-Projekte, Anwendungsentwicklung (inkl. durch Endbenutzer in Fachbereichen)**
- **IT-Betrieb (inkl. Datensicherung)**
- Auslagerungen und sonstiger Fremdbezug von IT-Dienstleistungen

Die wesentlichen Aspekte für die weiteren Betrachtungen sind die beiden oben hervorgehobenen Punkte. Weiter verschärfen würden sich die nachfolgend betrachteten Problemfelder beim Einsatz externer Dienstleister, da dieses eine weitere Reihe von Einschränkungen nach sich zieht, vor allem dann, wenn die Dienstleister aus Nicht-EU-Ländern tätig werden.

Im Überblick decken die BAIT die folgenden Bereiche der MaRisk ab:

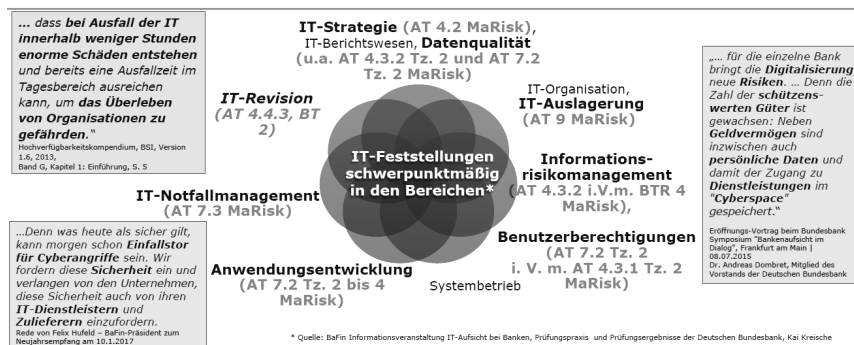


Abbildung 2 - BAIT-Themenbereiche zur Konkretisierung der MaRisk

Quelle: Informationsveranstaltung BaFin zu BAIT am 17.03.2017

[www.bafin.de/SharedDocs/Downloads/DE/Veranstaltung/dl\\_170316\\_it-aufsicht\\_banken\\_vortrag\\_2.pdf](http://www.bafin.de/SharedDocs/Downloads/DE/Veranstaltung/dl_170316_it-aufsicht_banken_vortrag_2.pdf)

Was ist nun das Ziel all dieser vielfältigen Regulierungen durch den Gesetzgeber und die Behörden?

In erster Linie geht es – gerade nach den Erfahrungen aus der weltweiten Finanzkrise am Ende der 2000er Jahre – um eine Risikominimierung gerade im Bereich Kreditvergabe und Investitionen. Dabei spielt die IT einerseits eine umsetzende Rolle (d.h. sie muss alle Funktionen zur Verfügung stellen, um z.B. die vielfältigen Meldewesen-Anforderungen aus dem Kreditgeschäft zu erfüllen), sie ist aber auch selbst Ziel dieser Regulierung, um zu verhindern, dass nicht ausgereifte und hinreichend abgesicherte Software entweder die Bank selbst (z.B. durch ungeplante Ausfälle von Kernsystemen wie Zahlungsverkehr) oder gar ganze Märkte (wie z.B. durch automatisierte Börsen-Handelssysteme) massiv negativ treffen.

Das Ziel der Regulierung durch die EZB und die BaFin ist also – wie der Name „MaRisk“ schon ausdrückt – die Mindestanforderungen an das Risikomanagement sowohl in den Fach-



bereichen als auch in der IT-Organisation konkret vorzugeben und auch periodisch zu prüfen (nach §44 KWG).

Dabei sind bestimmte Regeln der MaRisk „nicht verhandelbar“, wie z.B. eine strikte Trennung der Personen, die Software entwickeln und der Personen, die Software in den Betrieb überführen, die sog. „Segregation of Duties“ (Funktionstrennung). Diese Forderung nach einer Funktionstrennungsforderung gilt dabei nicht nur für die IT, sondern auch für alle anderen Bereiche der Bank, wo die Durchführung einer Prozesskette durch eine Person allein zu schweren Fehlern führen kann.

## 4 Die wesentlichen Reibungspunkte zwischen Agilität und Regulatorik

In den folgenden Tabellen sind – mit der Bewertung aus Sicht des Autors – die wesentlichen Anforderungspunkte der BAIT den wesentlichen Konzepten der agilen Entwicklung und auch der DevOps-Konzepte dargestellt.

Festzuhalten ist dazu, dass es aus Sicht der Regulierung prinzipiell „egal“ ist, mit welchem Projektmanagement- und Vorgehensmodell in einer Bank Software entwickelt und gewartet wird, solange das genutzte Vorgehensmodell inklusiver aller geforderten Artefakte dokumentiert ist und dabei ein paralleles Risikomanagement durchgeführt wird und vor allem die wesentlichen späteren Prüfdokumente (die sog. „Bestandsdokumentation“ jedes Systems) entstehen bzw. mit jeder Änderung an der Software angepasst werden.

### 4.1 Potentielle Problemfelder bei agiler Entwicklung

Im Folgenden sind kurz die zu beachtenden Konfrontationsfelder der Regulierung mit agiler Entwicklung aufgezeigt und aus Sicht des Autors bewertet. Maßgeblich für die Bewertung ist dazu primär der BAIT-Abschnitt „6 - IT-Projekte, Anwendungsentwicklung“:

| BAIT-Nr. | BAIT-Anforderung  | Bewertung  |
|----------|---|--|
| 32       | Die organisatorischen Grundlagen von IT-Projekten (inkl. Qualitätssicherungsmaßnahmen) und die Kriterien für deren Anwendung sind zu regeln.  | Unkritisch, solange die Prozesse dokumentiert sind   |
| 33       | IT-Projekte sind angemessen zu steuern, ... Hierfür sind Vorgehensmodelle festzulegen, deren Einhaltung zu überwachen ist.  | Unkritisch, solange die Prozesse dokumentiert sind und die Einhaltung nachweislich überwacht wird  |
| 36       | Für die Anwendungsentwicklung sind angemessene Prozesse festzulegen, die Vorgaben zur Anforderungsermittlung, zum Entwicklungsziel, zur (technischen) Umsetzung (einschließlich Programmierrichtlinien), zur Qualitätssicherung, sowie zu Test, Abnahme und Freigabe enthalten. | Wichtig!<br>Prozesse sind mit Leitlinien <b>vorab</b> genau festzulegen, insbesondere zur Qualitätssicherung (Übergang DevOps!)                    |
| 37       | Anforderungen an die Funktionalität der Anwendung müssen ebenso erhoben, bewertet und dokumentiert werden wie nichtfunktionale Anforderungen. Die Verantwortung für die Erhebung und Bewertung der Anforderungen liegt in den Fachbereichen.                                    | <b>Kritisch!</b><br>„Software over Documentation“ akzeptiert die Aufsicht nicht für alles! Gefordert sind explizit die späteren Bestandsdokumente. |
|          | Die Anwendung sowie deren Entwicklung sind  | <b>Kritisch!</b>   |

|    |  |  |
|----|--|--|
| 40 | übersichtlich und für sachkundige Dritte nachvollziehbar zu dokumentieren. | Bestandsdokumentation ist „nicht verhandelbar“! Sie muss lückenlos vorliegen! Eine regelmäßige Aktualisierung ist ebenfalls gefordert. |
|----|--|--|

#### 4.2 Potentielle Problemfelder beim Einsatz von DevOps-Konzepten

Im Folgenden werden kurz die zu beachtenden Konfrontationsfelder der Regulierung mit dem Einsatz von DevOps-Konzepten aufgezeigt und aus Sicht des Autors bewertet. Maßgeblich für die Bewertung sind dazu primär die BAIT-Abschnitte „6 - IT-Projekte, Anwendungsentwicklung“ und „7 - IT-Betrieb“:

| BAIT-Nr. | BAIT-Anforderung  | Bewertung   |
|----------|---|---|
| 39       | Im Rahmen der Anwendungsentwicklung müssen Vorkehrungen getroffen werden, die erkennen lassen, ob eine Anwendung versehentlich geändert oder absichtlich manipuliert wurde.   | <b>Kritisch!</b><br>Vier-Augen-Prinzip an allen kritischen Stellen gefordert!   |
| 41       | Es ist eine Methodik für das Testen von Anwendungen vor ihrem erstmaligen Einsatz und nach wesentlichen Änderungen zu definieren und einzuführen. Die Tests haben in ihrem Umfang die Funktionalität der Anwendung, die Sicherheitskontrollen und die Systemleistung unter verschiedenen Stressbelastungsszenarien einzubeziehen. | <b>Kritisch!</b><br>Aber gleichzeitig bester Hebel, die Vorzüge von DevOps aufzuzeigen, wenn z.B. Tests (insbesondere Regression) vollautomatisiert laufen. |
| 46       | Die Komponenten der IT-Systeme sowie deren Beziehungen zueinander sind in geeigneter Weise zu verwalten, und die hierzu erfassten Bestandsangaben regelmäßig sowie anlassbezogen zu aktualisieren   | Wichtig!<br>Im Rahmen von DevOps gut umsetzbar. Ggf. Verbesserungspotential gegenüber aktuellen Konfigurationsmanagement-Systemen!                          |
| 49       | Anträge zur Änderung von IT-Systemen sind in geordneter Art und Weise aufzunehmen, zu dokumentieren, unter Berücksichtigung möglicher Umsetzungsrisiken zu bewerten, zu priorisieren, zu genehmigen sowie koordiniert und sicher umzusetzen.  | <b>Sehr Kritisch!</b><br>Die größte „regulatorische Bedrohung“ bei der Einführung von DevOps-Konzepten! Das bekommt man vollautomatisch nicht hin.          |

## 5 Fazit und Vorschlag zum Vorgehen

Zusammenfassend lässt sich festhalten:

- Der Einsatz von agilen Entwicklungs- und Wartungskonzepten ist der aus Regulatorik-Sicht eher ungefährlichere Teil, da die Regulierung und die Aufsicht grundsätzlich keine Vorgaben machen, in welcher Organisationsform und nach welchem Vorgehensmodell Software erstellt und gewartet wird. Allerdings muss auch bei agiler Entwicklung weiterhin beachtet werden:

- Die geforderte Bestandsdokumentation muss in vorgeschriebenem Umfang immer aktuell vorliegen. Dazu gehören insbesondere die Systembeschreibungen, Benutzerhandbücher, Berechtigungskonzepte (!) und Betriebsführungshandbücher.
- Die Traceability jeder einzelnen Fachanforderung („User Story“) bis in die implementierte Software muss durchgehend nachweisbar sein. Hier haben neu gestartete agile Projekte sicher einen Vorteil gegenüber den schon länger bestehenden Systemen, bei denen die Traceability kaum noch herzustellen ist. Sie müssen allerdings dafür entsprechende technische Prozesse im Vorgehensmodell vorsehen.
- Lang lebende Kernsysteme mit primär von außen vorgegebenen (gesetzlichen) Anforderungen sind eher Kandidaten für klassische Release-Zyklen mit dem V-Modell als Vorgehen.
- Der Einsatz von DevOps-Konzepten (vor allem nach der „reinen Lehre“) ist aus Sicht der Regulatorik und Aufsicht nach heutigem Erkenntnisstand als „illegal“ einzustufen, wenn nicht zusätzliche Kontroll-Schritte eingefügt werden:
  - Die Entwickler dürfen auf keinen Fall die IT-Produktion direkt mit Software-Komponenten versorgen, ohne dass eine Betriebs-Person eine entsprechende Freigabe durchführt („Funktionstrennung“). Nach derzeitiger Lesart sind Entwicklung, Test, fachliche Abnahme, betriebliche Abnahme und Deployment in der Produktion von jeweils unterschiedlichen Personen durchzuführen.
  - Außerdem ist zu diesem Zeitpunkt sicherzustellen, dass die gesamte Dokumentation aktualisiert wurde. Auch dieses ist automatisiert nicht möglich.
  - Die von der Aufsicht geforderte Trennung zwischen Entwicklung und Betrieb muss in jedem Fall organisatorisch erhalten bleiben. Hier gibt es derzeit gegensätzliche Trends: Während in einigen Banken eine noch deutlichere Trennung von Entwicklung und Betrieb forciert wird, versuchen andere Banken, wieder tendenziell zu übergreifenden Teams zu kommen, soweit dieses regulatorisch möglich ist.
  - Die von der Regulatorik geforderten und an ITIL orientierten „Request for Change“-Prozesse lassen sich nur in Teilen weiter automatisieren. Eine Vollautomatisierung eines gesamten und vom Entwickler allein initiierten Deployment-Prozesses, wie im „reinen“ DevOps-Modell angestrebt, wird der Regulator nach derzeitiger Einschätzung nicht akzeptieren.

Wie kann man also die Ziele zum Einsatz agiler Entwicklungskonzepte und weiterführender DevOps-Konzepte bis in den Betrieb erreichen, ohne Gefahr zu laufen, nach der nächsten Prüfung nach §44 KWG durch die Aufsicht einem „Berg von Moniten (Beanstandungen)“ gegenüber zu stehen?

Für den agilen Entwicklungsteil bieten sich folgende Maßnahmen an:

- Alle Anwendungssysteme müssen nach Risikoklassen eingeteilt sein, das fordert die Aufsicht schon heute. Es bietet sich an, das agile Entwicklungsvorgehen vorrangig für die risikoärmeren Systeme einzusetzen. Eine Ausweitung auf andere Systeme – auch in der Wartung für bestehende Systeme – kann dann sukzessive erfolgen.
- Insbesondere die These „Software over Documentation“ aus dem Agilen Manifest muss mit der Regulatorik in Einklang gebracht werden. Dies gilt insbesondere für die aus aufsichtsrechtlicher Sicht unverzichtbaren Dokumente und Nachweise des Risikomanagements. Reine Projektdokumente sind sicher entbehrlicher, da sie am Ende des Projekts im Gegensatz zur System- und Betriebsdokumentation ohnehin nicht mehr relevant sind.
- Es ist vor Beginn an Wert auf die durchgängige Traceability jeder Anforderung bis zur Umsetzung in der Software zu legen. Diese Regelung wird in Zukunft sicher schärfer bewertet als bisher, vor allem auch aus Gründen der IT-Sicherheit.
- Es kann sich anbieten, im agilen Projekt spezialisierte Teams für die Abdeckung der Betriebs- und Compliance-Absicherung einzusetzen, also einen „Operations Tribe“ und ei-

nen „Compliance Tribe“, ggf. sogar in einem Team zusammengefasst. Damit ist auch ein Schritt zum Kulturwandel in Richtung DevOps vorbereitet.

Weiterhin sei hier auf die Arbeit von Timo Weinrich et al. [8] verwiesen, in der anhand einer Fallstudie die Einführung von agilen Konzepten bei einem Finanzdienstleister dargestellt wird.

Für das DevOps-Konzept sind folgende Maßnahmen zu empfehlen:

- Auch hier bietet sich an, das DevOps-Konzept zunächst vorrangig für die risikoärmeren Systeme einzusetzen. Eine Ausweitung auf andere Systeme – auch in der Wartung für bestehende Systeme – kann dann ebenfalls später sukzessive erfolgen.
- Man sollte das technische DevOps-Konzept (also den reinen Tool-Chain-Teil) an einem unkritischen System erproben und parallel festlegen, an welchen Stellen der Werkzeugkette bewusste Unterbrechungen („Hick-ups“) einzubauen sind, um explizite Freigaben durch eine andere Person als den Entwickler durchführen zu lassen.
- Vor Einführung muss eine saubere und durchgängige Dokumentation des gesamten Prozesses inklusiver aller eingesetzten Werkzeuge sowie eine Risikobetrachtung durchgeführt werden, die der Aufsicht vorgelegt werden kann.
- Wenn möglich, sollte man vor einer flächendeckenden Einführung auch für kritische Systeme eine Abstimmung mit der Aufsicht oder zumindest dem Wirtschaftsprüfer durchführen, ob das angestrebte Konzept so akzeptiert werden würde. Eine offizielle Stellungnahme der BaFin zum Einsatz von DevOps-Konzepten ist dem Autor nicht bekannt.
- Unabhängig von den technischen Aspekten sollte man den mit dem DevOps-Konzept angestrebten Abbau kultureller Barrieren zwischen Entwicklung und Betrieb forcieren, ohne jedoch die notwendige Rigidität in Bezug auf die Regulierung außer Acht zu lassen. Es kann nach Erfahrungen des Autors durchaus bereichernd für Entwickler sein, auch einmal die Betriebsseite ihrer Systeme kennenzulernen, in der Gegenrichtung können Betriebsmitarbeiter Teil des Entwicklerteams werden (wenngleich nicht unbedingt Programmierer).

Hier sei auf eine Arbeit von Joachim Baumann [4] verwiesen, die sich mit der Einführung von DevOps-Konzepten in regulierten Umgebungen befasst.

Abschließend lässt sich festhalten, dass wie bei anderen neuen Ansätzen (die im Kern dann doch meist nicht so neu sind, wie sie erscheinen oder zumindest „verkauft“ werden) eine dogmatische Verfolgung von Maximalzielen nicht praxistauglich ist.

Auch bei den agilen und DevOps-Philosophien gilt: Sie sind in bestimmten Projekten sicherlich hilfreich und führen schneller zu noch passgenaueren Ergebnissen als die klassische Wasserfall-basierten Entwicklung. Auf der anderen Seite können sie aber kein Freibrief sein, gängige und über Jahre bewährte Regelwerke wie z.B. ITIL zu ignorieren. Wie immer liegt auch in diesem Fall die Wahrheit in der Mitte: So viel Agilität und Geschwindigkeit wie möglich für Systeme und Projekte, wo die Ansätze passen, auf der einen Seite – und so viel Strenge und Stabilität wie nötig, um den Betrieb der Bank nicht zu gefährden. So kann man der nächsten Prüfung durch die Aufsicht gelassener entgegensehen.

## Literatur

- [1] John Allspaw and Paul Hammond: 10+ Deploys per Day - Dev and Ops Cooperation at Flickr, Velocity 09 – O'Reilly Conferences, 22.-24.06.2009, Download am 23.02.2028, <https://conferences.oreilly.com/velocity/velocity2009/public/schedule/detail/7641>
- [2] BaFin: Rundschreiben 09/2017 (BA) – Mindestanforderungen an das Risikomanagement – MaRisk, 27.10.2017, Download am 23.02.2018

- [https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/2017/rs\\_1709\\_mari\\_sk\\_ba.html](https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/2017/rs_1709_mari_sk_ba.html)
- [3] BaFin: Rundschreiben 10/2017 (BA) - Bankaufsichtliche Anforderungen an die IT (BAIT), 06.11.2017, Download am 23.02.2018, [https://www.bafin.de/SharedDocs/Downloads/DE/Rundschreiben/dl\\_rs\\_1710\\_ba\\_BAIT.html](https://www.bafin.de/SharedDocs/Downloads/DE/Rundschreiben/dl_rs_1710_ba_BAIT.html)
- [4] Joachim Baumann: DevOps light – DevOps in regulierten Umgebungen, OBJEKTSpektrum, 02/2015, Sigs Datacom, Download am 23.02.2018, [https://www.sigs-datacom.de/uploads/tx\\_dmjournals/baumann\\_OS\\_02\\_15\\_N8e4.pdf](https://www.sigs-datacom.de/uploads/tx_dmjournals/baumann_OS_02_15_N8e4.pdf)
- [5] Kent Beck et al.: Manifesto for Agile Software Development, Download am 23.02.2018, <http://agilemanifesto.org/> <https://agilemanifesto.org>
- [6] James Martin: Rapid Application Development, Macmillan Publishing Co., Indianapolis, IN, USA, 1991, ISBN 0-02-376775-8
- [7] Fredric Paul: The Incredible True Story of How DevOps Got Its Name, New Relic Blog, Download am 23.02.2018, <https://blog.newrelic.com/2014/05/16/devops-name/>
- [8] Timo Weinrich et al.: Herausforderungen bei der Einführung agiler Vorgehensmodelle für Finanzdienstleister – eine Fallstudie, in: Martin Engstler et al.: Projektmanagement und Vorgehensmodelle 2016, LNI, Ges. für Informatik, Bonn, 2016
- [9] Velocity 2011: Jon Jenkins, "Velocity Culture", Download am 23.02.2018, <https://www.youtube.com/watch?v=dxk8b9rSKOo>

# Vielfalt in der IT-Projektarbeit – Chancen erkennen und nutzen

Dorothee Feldmüller  
Hochschule Bochum, Campus Velbert/Heiligenhaus  
dorothee.feldmueller@hs-bochum.de

**Abstract:** Projekte werden heute häufig in international und/oder interdisziplinär gemischten Teams bearbeitet und sollen Ergebnisse für eine vielfältige Menge von Nutzern bereitstellen. Welche Auswirkungen hat Vielfalt in der Projektarbeit, welcher Nutzen kann dadurch erzielt werden, welche Schwierigkeiten sind dabei zu überwinden – Fragen, denen sich Projektverantwortliche auf allen Ebenen stellen müssen. In dem Artikel werden aktuelle Erkenntnisse zum Diversity Management zusammengestellt und an Projektbeispielen aus der IT-Projektwelt reflektiert. Insbesondere geht es um die Frage nach den Chancen, die in dem bewussten und gekonnten Umgang mit Vielfalt liegen – nicht zuletzt für die Projektarbeit in der Informatik, die immer noch mit dem Problem der „Nerd“-Mono-Kultur belastet ist.

## 1 Einführung

In Projekten arbeiten Teams daran, einzigartige und komplexe Aufgabenstellungen mit begrenzten Ressourcen zu lösen. Die Komplexität bringt es mit sich, dass in der Regel unterschiedliche Bereiche oder Disziplinen beteiligt werden müssen, die Globalisierung der Wirtschaft führt dazu, dass immer mehr Nationen und Kulturen beteiligt werden.

Die Vielfalt in einem Projektteam ist oft gar nicht bewusst ausgesucht, sondern ergibt sich aus der Aufgabenstellung und ist damit für das Projektmanagement gesetzt. Auch die Vielfalt der Nutzer des Projektergebnisses und der Stakeholder eines Projekts ist in der Regel vorgegeben.

Bewusster Umgang mit Vielfalt heißt Diversity Management und wird in vielen großen Organisationen mit eigenen Verantwortlichkeiten betrieben, in der Regel ohne spezielle Fokussierung auf Projektarbeit. Umgekehrt wird im Projektmanagement in der Regel über Teamarbeit und Teambildung reflektiert, und die Bedeutung einer guten Teambildung ist im Projektmanagement unbestritten - aber in der Regel wird in diesem Zusammenhang keine systematische Betrachtung von Diversity Management betrieben. In diesem Artikel werden Vielfalt und Projektarbeit miteinander in Verbindung gebracht, und die Chancen eines bewussten Umgangs mit Vielfalt in der Projektarbeit herausgestellt – ohne die Schwierigkeiten, die dabei auftreten können, außer Acht zu lassen.

Nach einem Überblick über aktuelle Forschungsergebnisse in diesem Zusammenhang werden auch Beispiele aus der IT-Projektwelt reflektiert.

## 2 Vielfalt und Projektarbeit

Unter Vielfalt bzw. Diversity verstehen wir die Unterschiedlichkeit von Menschen. Ein grundlegendes Modell, Diversity zu verstehen, stammt von Gardenswartz und Rowe ([GaRo95], zur deutschen Übersetzung siehe z.B. [ChdV17]). Unterschieden werden in dem Modell verschiedene Dimensionen der Vielfalt:

- Persönlichkeit
- Innere Merkmale wie Alter, Geschlecht, sexuelle Orientierung und Identität, ethnische Herkunft und Nationalität, Behinderung, Religion und Weltanschauung – diese sind in der Regel durch den Menschen nicht beeinflussbar
- Äußere Merkmale wie geografische Lage, Einkommen, Gewohnheiten, Freizeitverhalten, Berufserfahrung, Ausbildung u.a.m. – diese sind eher beeinflussbare Merkmale
- Organisationale Merkmale wie Funktion/Einstufung, Arbeitsinhalte/-feld, Abteilung/Einheit/Gruppe, Dauer der Zugehörigkeit u.a.m.

Bewusster Umgang mit oder mehr noch die konstruktive Nutzung von Vielfalt ist das Ziel von Diversity Management. Diese Aufgabe ist heute von vielen Organisationen erkannt und eine dafür verantwortliche Person in der Organisation benannt.

Die Bedeutung von Diversity Management ist in Deutschland auch gestiegen durch das Allgemeine Gleichbehandlungsgesetz (AGG), das in 2006 verabschiedet worden ist und die Benachteiligung aufgrund von ethnischer Herkunft oder Rasse, Geschlecht, Religion oder Weltanschauung, Behinderung, Alter oder sexueller Identität verhindern soll.

In der Projektarbeit spielt die Zusammenarbeit von Menschen mit unterschiedlichen Hintergründen schon immer eine Rolle: die Beteiligung unterschiedlicher Disziplinen und Erfahrungen an einem Projekt ist in vielen Fällen erforderlich und wird seit langem praktiziert.

In internationalen Projekten kommen Menschen mit unterschiedlichen kulturellen Hintergründen zusammen. Die Arbeiten von Hofstede, Trompenars und Hall zu Dimensionen kultureller Unterschiede (vgl. [Hofs91] bzw. die umfassende Darstellung in [HSF04]) haben ihren Einzug in das internationale Projektmanagement bereits vor Jahren gehalten und sind Gegenstand von Lehre, Forschung und Beratung.

Anders verhält es sich mit anderen der oben genannten Diversitäts-Merkmale: die Zugehörigkeit zu bestimmten Berufsgruppen oder das Geschlecht wird in der Projektarbeit selten offen und konstruktiv thematisiert – geschweige denn noch „sensiblere“ Merkmale wie Behinderung oder sexuelle Identität.

Die Projektmanagement-Community ist – wie die Informatik auch – bislang eine Männer-Domäne: der Anteil der weiblichen Mitglieder der GPM Deutsche Gesellschaft für Projektmanagement e.V. wird mit 23% angegeben ([GPM17]), an der fünften Gehaltsstudie im Projektmanagement 2015 waren 18% der Teilnehmer weiblich.

Sowohl Projektmanagement als auch Branchen mit hohem Projektanteil wie Informatik und andere technische Disziplinen versuchen, mehr Frauen für eine Tätigkeit in diesen Bereichen zu gewinnen, um mehr Vielfalt zu erreichen. Bislang ist es auch bei vielen Maßnahmen für Mädchen und Frauen in Mathematik, Informatik, Naturwissenschaft und Technik nicht gelungen, den Frauenanteil in diesen sogenannten MINT-Fächern so weit zu steigern, dass diese nicht (mehr) in einer Minderheitensituation arbeiten müssen.

Die besondere Problematik der Informatik einschließlich ihrer historischen Entwicklung stellt unter anderen Birgitta Böckeler dar ([Böck16a], [Böck16b], nach [Ensm10]): für die Software-Entwicklung wurden über einen längeren Zeitraum systematisch Personen mit bestimmten Persönlichkeitsmerkmalen rekrutiert. Zum Stereotyp des „Nerd“ gehört, „dafür geboren“ zu sein, sich schon sehr lange und fast ausschließlich mit Software-Entwicklung – und weniger mit Menschen – befasst zu haben. Dieses in der Realität vielfach vorhandene Profil wirkt auf anders geartete Personen nahezu abschreckend, vor allem aber auf die Frauen, die tendenziell gesehen auf die gesellschaftliche und soziale Relevanz ihrer Tätigkeit mehr Wert legen und mehr an kommunikativen Prozessen interessiert sind. Dass Informatik erlernbar ist und viele kommunikative Prozesse beinhaltet, ist nach außen nicht so bekannt

wie das „Nerd“-Stereotyp verbreitet ist, es bedarf einer besseren Außendarstellung der Berufsbilder (vgl. [RGDK17]). Aber es bedarf auch innerhalb der Informatik bewusster Anstrengungen, aufgeschlossen mit „Andersdenkenden“ umzugehen, diese für die Beteiligung an der Informatik einzuladen und zu integrieren.

Der derzeitige Frauenanteil von ca. 20% in der Informatik ist nicht hoch genug. Für die Arbeit in politischen Parlamenten wird eine Beteiligung von Frauen in einer Minderheit von mindestens 30% Frauenanteil angestrebt (z.B. [UN13]), um eine „kritische Masse“ an Frauen zu erreichen, die sich auf die Politik und Gesetzgebung spürbar auswirkt (Beispiele z.B. in [UN17]). Auf die Dauer sind 50% zu erreichen.

In Bezug auf das Geschlecht ist in der IT-Projektarbeit mehr Vielfalt gewünscht, und dieses Anliegen in Kombination mit den o.g. Anteilsgrenzen betrachten wir im Folgenden auch als ein Anliegen von Diversity Management in IT-Projekten.

Im Folgenden wollen wir am Beispiel der drei Dimensionen Geschlecht, Berufsgruppe und Nationalität bzw. kultureller Hintergrund Auswirkungen von Vielfalt bzw. Nicht-Vielfalt betrachten, die Chancen, die darin liegen, aber auch die Herausforderungen, um die Vielfalt zu managen. Durch die Betrachtung mehrerer Dimensionen gleichzeitig und die Betrachtung der Chancen soll der Blick im positiven Sinn geöffnet werden. Diversity Management hat immer wieder die Tendenz, als Förderung von Minderheiten, die ein Defizit haben, verstanden zu werden – nicht als eine wichtige Aktivität, um Defizite, die bei der herrschenden Mehrheit liegen, bewusst zu vermindern.

### **3 Auswirkungen von (Nicht-)Vielfalt in der Projektarbeit**

Die Notwendigkeit, aber auch die Schwierigkeiten, unterschiedliche Persönlichkeiten in der Teamarbeit zusammenzubringen, ist im Projektmanagement ein bekanntes Thema: Ansätze von Belbin ([Belb93]), sowie die Systematiken des Myers-Briggs-Type Indicator (MBTI, siehe z.B. [BeBl92]) und Team Management System nach Margerison/McCann (TMS, siehe z.B. [TSWa08]) sind in der Projektwelt bekannt und diskutiert (siehe z.B. [PrGo09]).

Aus der Arbeit in internationalen Projekten mit vielfältig besetzten Teams bekannt ist, dass typische Situationen in Projekten wie Zeitplanung, Durchführung von Sitzungen, Entscheidungsfindung, Umgang mit Risiken je nach kulturellem Hintergrund unterschiedlich gehandhabt werden und dadurch Potential für Missverständnisse oder gar Konflikte gegeben ist.

Zum Beispiel sind viele asiatische Kulturen eher durch eine große Machtdistanz zwischen Vorgesetzten und Mitarbeitern geprägt – im Vergleich zu Deutschen. Bei einer großen Machtdistanz erwarten die Mitarbeiter eher Anweisungen, bei einer kleineren erwarten sie, bei Entscheidungen beteiligt zu werden. Die geschulte deutsche Projektleitung weiß das zu berücksichtigen, indem sie z.B. Meinungen von asiatischen Team-Mitgliedern im direkten persönlichen Gespräch mit viel Umsicht einholt und in Sitzungen mit mehreren Beteiligten auf schwierige Diskussionen möglichst verzichtet. Auch im direkten persönlichen Gespräch ist das asiatische Bedürfnis nach Harmonie zu wahren und schwierige Themen sind eher indirekt anzusprechen. Offensichtlich werden die Kommunikationsprozesse deutlich komplexer durch die interkulturelle Komponente.

Will man ein internationales Projekt erfolgreich durchführen, so muss man diese Aspekte in der Projektarbeit berücksichtigen. Für eine ausführliche Darstellung der interkulturellen Aspekte im Projektmanagement verweisen wir auf die bereits genannte Literatur ([Hofs91] bzw. [HSF04]).



Erfolgreiches Projektmanagement ist darauf angewiesen, Mono-Kultur im Denken des Projektteams zu vermeiden. Bekannte Probleme in der Teamarbeit bei einer solchen Mono-Kultur sind unter anderem das Gruppendenken und das Risk-Shifting, das Akzeptieren unvernünftiger Risiken durch eine Gruppe aufgrund des „Teilens“ des zu tragenden Risikos. Empfohlene Gegenmaßnahmen liegen darin, die Gruppe zum offenen vielfältigen Denken aufzufordern und bzw. oder dies durch eine entsprechend vielfältige Gruppenzusammensetzung vorzubereiten (vgl. wieder [PrGo09]).

Bekannte Probleme bei einer einseitigen Besetzung eines Entwicklungsteams sind Akzeptanz-Probleme bei Kunden- bzw. Benutzerkreisen, die nicht ausreichend berücksichtigt wurden, Probleme der Usability u.a.m.

Die Bedeutung der Berufsgruppenzugehörigkeit für Innovationen, insbesondere die Unterschiede zwischen Ingenieuren und Ingenieurinnen im Vergleich zu Betriebswirtinnen und Betriebswirten, untersucht Lüthje ([Lüth08]). Für unsere Betrachtung unterstellen wir eine Nähe der Informatikerinnen und Informatiker zum Ingenieurwesen. Lüthje diagnostiziert signifikante Unterschiede in Aufgaben- und Tätigkeitspräferenz, Informationsstil und Entscheidungen sowie Verkaufsneigung, daneben Unterschiede, die sich schon bei der Auswahl der Projekte und der Ziel- und Zeitorientierung auswirken: im Vergleich riskieren Ingenieure eher weniger, sind eher an kurzfristigen Innovationen interessiert, und setzen eher auf höhere technische Qualitätsziele. Die Betriebswirte neigen eher zu riskanten, langfristigen und eben auch strategischen Projekten. Es liegt auf der Hand, dass nur eine ausgewogene Auswahl aus beiden Lagern der Innovationstätigkeit einer Organisation auf die Dauer gut tut. In der betrieblichen Praxis technischer Unternehmen ist dies in der Regel der Fall – wird nur oft nicht bewusst reflektiert, ebenso wenig wie die oben genannten Unterschiede in der Arbeitsweise. Auch hierin liegt Potential für Konflikte und Missverständnisse in der Projektarbeit.

Die Unterschiede zwischen den Geschlechtern sind vielfach erforscht. Aus den Darstellungen von Inga Woudstra ([Woud15]) und Sandberg [Sand13]) haben wir in Tabelle 1 zusammengestellt:

|   | <b>Männer</b>   | <b>Frauen</b>   |
|---|---|---|
| <b>Wettbewerb um</b>                        | Erfolg und Anerkennung  | Gefallen und wertgeschätzt werden   |
| <b>Sicherheit findend</b>                   | durch Größe, Erfolg ("I am the best")                           | durch Freundschaften und Beziehungen  |
| <b>Entscheidungen</b>                       | aufgrund von Fakten   | mit Fokus auf Menschen  |
| <b>Denken</b>                               | fokussiert, faktenbasiert, bottom-up zum System zusammensetzend | Big Picture, top-down, im System zuerst die Beziehungen, dann die Komponenten sehend  |
| <b>Empathie</b>                             | für Gefahr, Selbst-Fokus, Intuition für den Raum                | für Gefühle, Außen-Fokus, Intuition für Menschen                                      |
| <b>Verbindungen</b>                         | durch gemeinsames Tun   | durch Teilen von Gefühlen   |
| <b>Herausforderungen für Gender-Balance</b> | Erfolgreiche Männer beliebt                                     | Erfolgreiche Frauen unbeliebt<br>Beliebte/Nette Frauen eher als inkompetent angesehen |
| <b>Beförderung</b>                          | aufgrund Potential  | aufgrund vergangener Erfolge (BaYe11, Sand13)   |

Tabelle 1: Typische Unterschiede zwischen Frauen und Männern (nach [Woud15] und [Sand13])

Die tendenziell stärkere Empathie und Beziehungsorientierung der Frauen bedeutet einen Vorteil für die Teamarbeit in Projekten wie auch für das Stakeholdermanagement – wird aber für Situationen, in denen Durchsetzungsvermögen gezeigt werden muss, den Frauen eher zum Problem.

Dies gilt allerdings nicht in jeder Kultur. Kulturen, die „femininer“ im Sinne von Hofstede orientiert sind, wie z.B. in Skandinavien, fordern weniger Durchsetzungsvermögen und mehr Konsensorientierung – was sich zum Vorteil der Frauen auswirkt bzw. wird es von Vorteil sein, dort in ausreichendem Maße Frauen an Bord des Projektteams oder der Teamleitung zu haben.

#### **4 Chancen von Vielfalt im Projekt**

Systematische Berücksichtigung und Integration von Vielfalt im Projektteam hat für das Team eine beflügelnde Wirkung. Gemeint ist damit aber eben nicht die Beteiligung von Personen mit vielfältigen Merkmalen um der Vielfalt willen, möglicherweise widerwillig und lustlos, etwa weil es das Gesetz oder die Gesellschaft verlangen. Gemeint ist stattdessen die Wertschätzung der vielfältigen Hintergründe, das aktive, bewusste und auch unterschiedliche Sichtweisen fördernde und fördernde Zugehen auf andersartige Personen: „Diversity is being invited to the party - inclusion is being asked to dance.“ – dieser im Zusammenhang mit Diversity öfter zitierte Ausspruch bringt auf den Punkt, dass es aus Sicht der Andersartigen nicht um Dabeisein geht, sondern darum, dass ausdrücklich gewünscht ist, dass die Andersartigen mitfeiern und alle sich daran freuen, dass dies gelingt. Dass dies aus Sicht der veranstaltenden oder bislang dominierenden Gruppierung nicht bequem ist, dass es Anstrengungen kostet, etwaige „Mauerblümchen“ zum Mitfeiern zu bewegen, und zwar auf allen Seiten, das wird im nächsten Abschnitt noch zu thematisieren sein. Wer eine Situation schon einmal miterlebt hat, in der dies erfolgreich gelungen ist, braucht nicht weiter überzeugt zu werden, dass dies eine besondere Anstrengung wert ist.

Eine andere Sichtweise als die der herrschenden Mehrheit zu verstehen kann eine Bereicherung bedeuten, zunächst eine persönliche Bereicherung und Weiterentwicklung. Selbst wenn die bislang eingenommene Position nicht verändert werden kann oder will, führt das Verständnis zu einem Zugewinn an Selbsterkenntnis und Verbesserung der Argumentation für die bisherige Position. Handelt es sich um einen Teamprozess, so werden die Positionen im Team transparenter und verständlicher für alle, was ein Gewinn für die einzelnen Personen wie für die Zusammenarbeit im Team darstellt. Mit jedem kleinen Fortschritt im Verständnis untereinander wird ein positives Gefühl erzeugt, das selbstverstärkende Mechanismen freisetzen kann und die Motivation, hieran weiter zu arbeiten, hoch hält oder gar steigert. Das Engagement des Teams für alle Projektziele steigt, Commitment steigt, aber auch die Verlässlichkeit der Planung in Bezug auf das Projektergebnis sowie Zeit und Kosten, die dafür aufzubringen sind, weil mehr Sichten einfließen und alle Entscheidungen auf einer breiteren Basis stehen.

Vielfalt kann besonders hilfreich sein, wenn Polarisierungen oder Konflikte zwischen beteiligten Parteien auftreten. Eine weitere, nochmal „andersartige“ Partei kann hier neutralisierende Wirkung ausüben, und sie kann dies umso mehr, je weniger formelle Macht ihr zugesprochen wird, weil sie dann ihren Standpunkt verdeutlichen kann, ohne dass darin eine starke Einflussnahme gesehen wird. Scheinbar machtlosen bzw. weniger einflussreichen Parteien wird eher ein sachorientiertes Vorgehen zugesprochen, und genau diese Rolle können sie hier positiv wahrnehmen.

Ein in der Praxis zu beobachtender Effekt ist auch, dass mehr „Andersartigkeit“ manchmal hilft, eine stereotype Polarisierung zu vermeiden: eine Frau mit einem anderen kulturellen Hintergrund etwa kann freier ungewöhnliche Lösungsideen vorbringen, die weder ihrem Frau-Sein noch ihrer anderen kulturellen Zugehörigkeit zugeschrieben werden. Vorausgesetzt, dass die Andersartigkeit und Ungewöhnlichkeit den notwendigen Respekt erfährt, gibt eine solche Konstellation die Chance, einen ungewöhnlichen Vorschlag vorurteilsfreier anzuhören und zu prüfen.

Vielfalt im Team bedeutet, über unterschiedliche Stärken und Schwächen zu verfügen und diese hilfreich einsetzen zu können, sofern man sich derer bewusst ist.

Projektmanagerinnen und Projektmanager sind Menschen, die an ihre Selbstwirksamkeit glauben (siehe [GPM09], S. 31), und sind an lebenslanges Lernen gewöhnt. Damit tragen sie direkt und indirekt auch für ihr Projektteam dazu bei, dass selbstverstärkende Mechanismen auch bei zunächst zähen Fortschritten in der Behandlung von Vielfalt eintreten.

Im wirtschaftlichen Sinne kann die Weiterentwicklung der Personen und der Teams dazu führen, dass ein zu entwickelndes Produkt mehr Akzeptanz auf dem Markt bekommt, weil mehr Sichten und Nutzungsmöglichkeiten betrachtet werden. In den heute stark umkämpften Märkten kann es den entscheidenden Wettbewerbsvorteil bedeuten, wenn das Denken der bislang vorherrschenden Gruppe – das sogenannte Silo-Denken – durchbrochen wird und weitere bzw. alle Stakeholder, neue Ideen und Geschäftsmodelle mit in den Blick genommen werden.

Umfassende Darstellungen zu Chancen von Vielfalt mit Fallbeispielen und Studien finden sich bei DiStefano/Maznevski [DiMa00] und bei Stuber [Stub14].

Denk-Mono-Kulturen führen zu Defiziten, Vielfalt zur Verbesserung, Andersartige haben nicht ein Defizit, sondern bringen Gewinn. Auch die Integration einer Person, die ein deutliches Defizit hat, gehört zu den positiven Erinnerungen an gelingende Vielfalt: nie war die Zusammenarbeit im Seminar und die Qualität der Ergebnisse der Teamaufgaben so gut wie bei dem Durchgang, als es gelungen ist, eine taubstumme Person in die Teams zu integrieren. Die kleine Anstrengung, die vorab zu absolvieren war: sich darauf einzulassen und zu verstehen, wie man mit der taubstummen Person kommunizieren kann.

Wie angekündigt möchten wir noch Fallbeispiele aus der IT vorstellen:

In unserem ersten Beispiel ist in einem internationalen Konzern eine Software zur Unterstützung technischer Produktentwicklungen einzuführen, ein sogenanntes Product Lifecycle Management (PLM-) System mit einigen Komponenten und konzerninternen Hilfsmitteln. Das System wird die Grundlage schaffen, die gemeinsame Nutzung von technischen Unterlagen bei der Produktentwicklung zu ermöglichen. Mit der Einführung sind neue Prozesse der Zusammenarbeit zu installieren, vor allem für die Nutzer in Fernost – Indien und China - wird vieles neu sein, wobei die Nutzer in Indien aufgrund bereits vorhandenen Austauschs von Daten einen Informationsvorsprung gegenüber den Nutzern in China besitzen.

Die bekannte asiatische Scheu, Probleme offen anzusprechen, stellt ein Risiko für die Einführung dar. Um diesem zu begegnen, wird eine gemeinsame Systematik erarbeitet, in dem Testergebnisse bezüglich Prozessablauf, Prozessergebnis und Antwortzeitverhalten transparent festgehalten und zwischen den beteiligten Standorten verglichen werden. Diese Systematik wird bei einem gemeinsamen Treffen aller Key-User an einem europäischen Standort vertieft und anschließend weltweit genutzt. Bestehende technische Hindernisse an einzelnen asiatischen Standorten werden so schnell transparent, ohne dass diese Gefahr laufen, auf einzelne Nutzer zurückzufallen.

In einer täglichen gemeinsamen Web-Konferenz werden nach dem Go-Live die Probleme zusammengetragen und anschließend bearbeitet. Für die Asiaten ist es anscheinend hilfreich zu sehen, dass auch europäische Standorte Probleme haben – seien es Probleme, für die technische Nacharbeiten erforderlich sind, oder solche, die durch Nachschulungen gelöst werden können.

Als Schlüsselperson erweist sich eine Managerin am chinesischen Standort, die indische Wurzeln hat: als einzige Person aus dem Management besucht sie eine Schulung, zusammen mit ihren überwiegend männlichen Mitarbeitern – „typisch Frau“ hat sie Statusdenken

zugunsten von Sich-Eindenken- und –Einfühlen-Können zurückgestellt. Mit sehr guten Sprachkenntnissen und viel Einfühlungsvermögen sammelt und adressiert sie Anlaufprobleme am Standort in China und motiviert ihre Mitarbeiter, Lösungsvorschläge auszuprobieren und bringt damit das Projekt an diesem Standort entscheidend weiter. Die Zusammenarbeit mit den Chinesen auf der Arbeitsebene fühlt sich für alle Beteiligten gut an, und dies führt auch bei vorhandenen Problemen zu einer schnellen und zufriedenstellenden Lösung. In der Nachbetrachtung hat genau das akzeptierte Anders-Sein der genannten Schlüsselperson viel bewirkt – ein positives Beispiel für Vielfalt.

Ein zweites Beispiel stammt aus dem militärischen Bereich: im Rahmen eines NATO-Einsatzes ist eine IT-Infrastruktur in einem zum östlichen Kulturkreis gehörenden Land auszurollen. Vertreter großer westlicher Nationen sind vor Ort, zum Teil halten sie sich mit Polarisierungen aufgrund ihres leicht unterschiedlich verstandenen westlichen Sendungsbewusstseins auf. Gut tut dabei die wenigstens teilweise Neutralisierung durch Vertreter einer kleineren westlichen Nation. Und gut im Sinne von neuen Chancen täte auch die im militärischen Bereich noch nicht heimische femininere Herangehensweise, die mehr auf Kommunikation und Kompromisse setzt. Es ist klar, dass dies für diesen Fall nicht ohne weiteres direkt umsetzbar wäre, aufgrund der andersartigen Kultur vor Ort sogar besonders herausfordernd, aber die Frage, wie sich eine andere Herangehensweise hier doch einbringen lässt, verdient in jedem Fall Aufmerksamkeit und wird von einem Beteiligten aktiv gesucht, um in der schwierigen Kommunikationssituation aus der eingespielten Denke herauszukommen.

## **5 Herausforderungen durch Vielfalt und Unterstützungsmöglichkeiten für das Projektmanagement**

Wie bereits angesprochen, erfordert Vielfalt größere Anstrengungen. Wir kommen auf den oben genannten Vergleich mit der Party zurück: Alle Team-Mitglieder auf die jeweils für sie passende Art zum Tanzen aufzufordern, kostet mehr Aufwand und verlängert Anlaufphasen von Teams. Die Sprache und die Sichtweise der Anderen muss erst erlernt werden, als erster Schritt muss erlernt werden, dies überhaupt als erstrebenswertes Ziel zu sehen. Und auch mit viel Bemühen und Schulung wird keine Person mit jeglicher Art von Vielfalt gut umgehen können. Selbst bei guter Vorbereitung springt keine und keiner ohne weiteres über seinen kulturellen Schatten. Wer z.B. in einer maskulinen Kultur aufgewachsen ist, wird sich in einer femininen Kultur rational hineindenken können, aber immer noch unwohl – eben nicht zuhause – fühlen. Darüberhinaus ist die Gefahr, wichtige Unterschiede zu übersehen oder nicht damit umgehen zu können, immer vorhanden.

Die Vorteile von Vielfalt – wie eine bessere Akzeptanz der Projektergebnisse – werden oft später und an anderer Stelle geerntet, so dass sie für diejenigen, die vorne investieren in die Anlaufphase, nicht unbedingt spürbar werden und diesen ersten Anstrengungen auch nicht ohne weiteres direkt zuzuordnen sind. Oder noch schwieriger: Belohnungen werden möglicherweise nicht “verursachergerecht” verteilt – ein Thema, das bei interdisziplinären Teams mit Anreizsystemen “je Disziplin” Schwierigkeiten verursachen kann und bei der oben genannten Untersuchung von Lüthje [Lüth08] thematisiert wird.

Ganz klar bedeutet Vielfalt auch Kompromisse und Verlust an Einflussnahme aus Sicht einer vorherrschenden Gruppe. Wenn diese nicht als Bereicherung der gesamten Situation verstanden werden kann oder will, sind Konflikte vorprogrammiert.

DiStefano und Maznevski ([DiMa00]) unterscheiden aufgrund ihrer Studien Destroyers, deren Umgang mit Andersartigkeit von Misstrauen und gegenseitigen Angriffen durchsetzt ist, Equalizers, die Vielfalt ignorieren, sich nur auf Gemeinsamkeiten konzentrieren und nur zu mittelmäßigen Ergebnissen kommen, und schließlich die sogenannten Creators, die Vielfalt

produktiv nutzen. Diese produktive Nutzung erfolgt über das Verstehen der Unterschiede, das Kommunizieren darüber und deren permanente Handhabung. Im Anhang ihrer Arbeit (siehe [DiMa00]) bieten sie eine Checkliste an, die helfen kann, Merkmale zu identifizieren, bei denen es Unterschiede im Team gibt, die dazu führen können, dass Barrieren für die Zusammenarbeit daraus entstehen, aber auch Potential zur Nutzung der Projektziele beinhalten. Die Checkliste beinhaltet Fragen zur Sicht von Beziehungen, Umwelt, Aktivitäten, Menschenbild und Umgang mit der Zeit. Sie berichten von Fallbeispielen, bei denen Teams erschrocken waren über die Vielzahl der Unterschiede. Nach dem Erkennen und Auswählen sind die Unterschiede im Team und ihre Auswirkungen zu beschreiben. Die Unterschiede gilt es dann in der Kommunikation explizit zu berücksichtigen. Kommunikation wird durch die Unterschiede komplexer und anstrengender, so dass mehr Motivation, mehr Ausdauer erforderlich ist, Vertrauen erst gebildet werden muss. In der Regel will der Kommunizierende selbst verstanden werden, und muss sich immer wieder anstrengen, den oder die anderen zu verstehen, Vorurteile und schnelle Bewertungen in sich bekämpfen, Unterschiede als neutral und vorhanden bewerten, statt andersartiges Verhalten abzuwerten. DiStefano und Maznevski sprechen hier von den Fähigkeiten des “De-Centering” und des “Re-Centering”. Die Integration der andersartigen Beteiligten erfolgt durch sorgfältige Beobachtung, dass alle Beteiligten teilhaben, nicht nur eine dominante Gruppe oder Person zur Meinungsbildung beiträgt. Dies kann gefördert werden, indem eine Person dafür verantwortlich gemacht wird dies sicherzustellen, und indem auch andere Kommunikationskanäle als Sitzungen gewählt werden, die oft von wenigen Aktiven dominiert werden. Unstimmigkeiten, auch unsichtbare, gilt es aufzuspüren und aufzulösen, z.B. in dem nach Gründen dafür gesucht wird, um alle Sichten zu verstehen und dadurch zu einem erweiterten Verständnis und Lösungsraum zu kommen. Ideen von allen müssen einfließen und dürfen nicht zu schnell aussortiert werden. Nur durch bewusstes, durch Maßnahmen unterstütztes Diversity Management ist es möglich, das Potential heterogen besetzter Teams auszuschöpfen. Ansonsten bleiben sie hinter den Leistungen homogener Teams möglicherweise zurück.

Es ist auch offen zu thematisieren, dass Vielfalt nicht für jedes Vorhaben notwendig und gewinnbringend ist. Wenn schnell Ergebnisse vorliegen müssen, ist die längere Anlaufphase, die Vielfalt mit sich bringt, eventuell nicht hilfreich und sogar schädlich und es muss ein eingespieltes homogenes Team eingesetzt werden.

Unterstützung für den erfolgreichen Umgang mit Vielfalt liegt also in der Verbreitung von Wissen, Beispielen im Allgemeinen wie auch in der Transparenz darüber im konkreten Fall. Für bekannte Wissensgebiete wie Interkulturelle Kompetenzen gibt es Schulungs- und Beratungsmöglichkeiten. Ein vielfach probates Mittel der Unterstützung ist ein Perspektivenwechsel bzw. Austausch – machbar für internationale oder interdisziplinäre Vielfalt, für andere Merkmale aber eben nicht oder nur sehr schwierig möglich.

Hilfreich können auch “Botschafter” zwischen den Kulturen wirken – für alle Merkmale von Diversität – sofern die Botschafter sich in mehreren Welten verständigen und dazwischen vermitteln können.

Lüthje [Lüth08] weist aus berechtigten Gründen darauf hin, dass Anreizsysteme Vielfalt unterstützen müssen. Insbesondere dürfen sie nicht für Teilanliegen oder Teilgruppen falsche Anreize setzen – dann bergen sie geradezu die Gefahr, kontraproduktiv zu wirken und die verschiedenen Gruppen gegeneinander auszuspielen.

Vielfalt ist ein Thema, welches vom Top-Management gewollt sein muss, und die Offenheit für Andersartigkeit, das Aushalten von Unterschieden und von länger dauernden Einigungsprozessen muss dort vorgelebt werden. Und die Ziele, die eine Organisation oder ein Projekt damit erreichen will, müssen geteilt werden können und am besten auch transparent und offen geteilt werden.

Die Projektwirtschaft kennt sich aus mit Team-Building und hat sicherlich viel Erfahrung und Wertschätzung für Vielfalt in den verschiedensten Dimensionen. Dies gilt auch für IT-Projekte, die schon immer die Zusammenarbeit mit dem vielfältigen künftigen Nutzer erfordert haben, und dessen Rolle in der Geschichte der Informatik gestärkt worden ist. Dieses Gedankengut gilt es auch in den Alltag der Software-Entwicklung noch konsequenter einzubringen.

Wir kommen noch einmal auf die Beispiele aus dem vorherigen Abschnitt zurück:

Bei der Einführung des PLM-Systems wird deutlich, dass das Bemühen, Testergebnisse und Probleme so transparent zu machen, dass es für die Europäer wie auch die Asiaten möglichst gut passt, aufwendiger, aber eben auch zielführender ist. Und die Besetzung der geschilderten Schlüsselperson ist eher ein Erfolg guter Intuition als systematischen Diversity Managements, zeigt aber das Potential von Vielfalt deutlich und lädt ein, durchaus auch mutige und diverse Personalentscheidungen für schwierige Situationen zu treffen.

Im Fall des Ausrollens einer IT-Infrastruktur in den Osten ist die Beteiligung mehrerer Nationen an der Mission ein Ergebnis systematischen Umgangs mit Vielfalt bei der NATO. Die Überlegung, hier einen weiblichen Rat einzuholen, ist aus einer systematischen Reflektion der schwierigen Situation entstanden und verweist ebenfalls auf das Potential von Vielfalt. Ob hier eine andersartige personelle Besetzung oder die rein virtuelle Beteiligung einer andersartigen Person hilfreicher sind, lässt sich diskutieren.

## **6 Zusammenfassung und Ausblick**

Vielfalt gab es in IT-Projekten schon immer, mit der Globalisierung und der Zunahme der Komplexität nimmt die Vielfalt in der Projektarbeit zu und damit die Herausforderungen diese zu managen. Die Chancen, die darin liegen, haben wir versucht zu verdeutlichen und am Beispiel zu illustrieren. Wie die Herausforderungen zu bewältigen sind, dafür haben wir Hinweise vorgestellt.

Bezogen auf IT-Projekte bzw. Informatik gilt es weiterhin, mehr und vielfältigere Beteiligung von Andersartigen zu erreichen, die blinden Flecken der vorherrschenden „Nerd“-Kultur ein Stück weit zu bekämpfen und auch insgesamt mehr Personen für diese Domäne zu begeistern. Dafür muss ein vielfältigeres Berufsbild von der Arbeit in IT-Projekten nach außen transportiert werden (vgl. [RGDK17]). Dafür muss auch in der Innensicht Aufgeschlossenheit gelebt werden, und täglich müssen kommunikationsfreudige Personen beiderlei Geschlechts, die nicht schon seit frühester Kindheit täglich am Computer tätig gewesen sind, zum „Mittanten an der Party“ an diesem Gerät, mit seinen Benutzern, mit den immer neuen Möglichkeiten eingeladen werden.

### **Danksagung**

Für das, was ich in vergangenen Jahr über Vielfalt gelernt und ausprobiert habe, danke ich den Kolleginnen und Kollegen in meinem Umfeld, vor allem bei der GPM Deutsche Gesellschaft für Projektmanagement e.V., den Teilnehmerinnen und Teilnehmern der ersten Tagung über „Diversity in Project Management“ im April 2017, den Teilnehmerinnen und Teilnehmern an meinem Workshop beim PM-Forum 2017, meinem Koreferenten Klaus Wagenhals, und den Kolleginnen und Kollegen in der Kommission für Diversity an meiner Hochschule.

Insbesondere danke ich allen, die mir geholfen haben, die oben erläuterte „Defizit-Denke“ zu überwinden: Andersdenkende bzw. Andersartige sind immer eine Bereicherung, wenn man/frau sich auf sie einlässt!

## Literaturverzeichnis

- [BaYe11] Barsh, Joanna; Yee, Lareina: Special Report: Unlocking the full potential of women in the U.S. economy, McKinsey & Company, 2011. [www.mckinsey.com](http://www.mckinsey.com)
- [BeBl92] Bents, R.; Blank, R.: Der M.B.T.I. Claudius Verlag, München 1992.
- [Belb93] Belbin, R. M.: Team Roles At Work. Oxford 1993, Butterworth Heinemann.
- [Böck16a] Böckeler, B.: Born for it – How the image of software developers came about. Frauen machen Informatik 40, Gesellschaft für Informatik e.V., September 2016.
- [Böck16b] Böckeler, B.: Born for it – How the image of software developers came about. <https://www.thoughtworks.com/de/insights/blog/born-it-how-image-software-developers-came-about>, zuletzt abgerufen 15.11.2016.
- [Ensm10] Ensmenger, N.: "The computer boys take over: Computers, programmers, and the politics of technical expertise", The MIT Press, Cambridge MA, 2010.
- [ChdV17] <https://www.charta-der-vielfalt.de/diversity-verstehen/diversity-dimensionen/> , zuletzt abgerufen am 06.11.2017.
- [DiMa00] DiStefano, J.; Maznevski, M.: Creating Value with Diverse Teams in Global Management. In: Organizational Dynamics, Vol. 29, No. 1, 2000, S. 45 – 63.
- [GaRo95] Gardenswartz, L.; Rowe, A.: Diverse Teams at Work – Capitalizing on the Power of Diversity. Irwin Professional Publishing, Chicago 1995.
- [GPM09] GPM Deutsche Gesellschaft für Projektmanagement e.V., Studie “Zur beruflichen Situation von Projektmanagerinnen und Projektmanagern”, 2009, [https://www.gpm-ipma.de/fileadmin/user\\_upload/GPM/Know-How/PMExpertinnen\\_Studien\\_final\\_091115.pdf](https://www.gpm-ipma.de/fileadmin/user_upload/GPM/Know-How/PMExpertinnen_Studien_final_091115.pdf), zuletzt abgerufen am 20.11.2017
- [GPM15] GPM Deutsche Gesellschaft für Projektmanagement e.V., 5. Gehaltsstudie 2015, [https://www.gpm-ipma.de/fileadmin/user\\_upload/GPM/Know-How/GPM\\_Studie\\_GehaltKarriere\\_2015\\_ES.pdf](https://www.gpm-ipma.de/fileadmin/user_upload/GPM/Know-How/GPM_Studie_GehaltKarriere_2015_ES.pdf) , zuletzt abgerufen am 06.11.2017.
- [GPM17] GPM Deutsche Gesellschaft für Projektmanagement e.V., Flyer der PM-Expertinnen, [https://www.gpm-ipma.de/fileadmin/user\\_upload/GPM/Know-How/150609\\_Flyer\\_PM-Expertinnen\\_Web\\_Final.pdf](https://www.gpm-ipma.de/fileadmin/user_upload/GPM/Know-How/150609_Flyer_PM-Expertinnen_Web_Final.pdf) , zuletzt abgerufen am 06.11.2017.
- [HeOk07] Heilman, Madeline E.; Okomito, Tyler G.: Why are women penalized for success at male tasks? The implied communylity deficit. In: Journal of Applied Psychology 92, Nr. 1 (2007), S. 81-92.
- [Hofs91] Hofstede, G.: Cultures and Organizations – Intercultural Cooperation and Its Importance for Survival. McCraw-Hill International (UK) Limited, 1991.
- [HSF04] Hoffmann, H.-E.; Schoper, Y.; Fitzsimons, C.J.: Internationales Projektmanagement – Internationale Zusammenarbeit in der Praxis. Beck-Wirtschaftsberater im dtv, München 2004.
- [Lüth08] Lüthje, Christian: Der Prozess der Innovation. Mohr Siebeck, Tübingen, 2008.
- [PrGo09] Prudix, D.; Goerner, M.: Teamarbeit. In: Gessler, Michael (Hrsg.): Kompetenzbasiertes Projektmanagement (PM3). GPM, Nürnberg 2009.
- [RGDK17] Resch, D.; Graf, I.; Dreiling, A.; Konrad, J.: Attraktivität von ICT-Berufen – Synthesericht, <https://www.fhnw.ch/de/die-fhnw/hochschulen/hsw/pmo/forschung-und-dienstleistung/attraktivitaet-von-ict-berufen>, zuletzt abgerufen am 13.11.2017.
- [Sand13] Sandberg, Sheryl: Lean in. Econ, Berlin, 2013.
- [Stub14] Stuber, Michael: Diversity & Inclusion – Das Potenzial-Prinzip. Shaker, Aachen 2014.

- [TsWa08] Tscheuschner, M.; Wagner, H.: TMS – Der Weg zum Hochleistungsteam. Gabal, Offenbach 2008.
- [Woud15] Woudstra, Inga: Be Gender Smart. St. Albans, Herts, UK Panoma Press Ltd., 2015.
- [UN13] UN Secretary-General's Report on Women and Political Participation (2013). "Measures taken and progress achieved in the promotion of women and political participation", A/68/184, [http://www.un.org/ga/search/view\\_doc.asp?symbol=A/68/184&Lang=E](http://www.un.org/ga/search/view_doc.asp?symbol=A/68/184&Lang=E), zuletzt abgerufen am 13.11.2017.
- [UN17] UN Women, Women's Leadership and Political Participation, <http://www.unwomen.org/~media/headquarters/attachments/sections/library/publications/2013/12/un%20women%20thembriefuswebrev2%20pdf.ashx>, zuletzt abgerufen am 13.11.2017.



# Agile Qualitätssicherung von Big Data Projekten - Bedarf an integrierten Test- und Bewertungsansätzen

Andreas Schmietendorf, Jan Hentschel

HWR Berlin, Ultra Tendency UG

Email: [andreas.schmietendorf@hwr-berlin.de](mailto:andreas.schmietendorf@hwr-berlin.de), [jan.hentschel@ultratendency.com](mailto:jan.hentschel@ultratendency.com)

**Abstract:** Big Data Projekte finden sich sowohl im industriellen als auch öffentlichen Umfeld. Ständen zunächst eher methodische und technologische Aspekte im Mittelpunkt der Betrachtung, setzt sich zunehmend die Erkenntnis durch, dass die Qualität der eingebrachten Daten und der korrekte und vor allem nachvollziehbare Einsatz entsprechender Analysealgorithmen über den nachhaltigen Erfolg entscheiden. Dennoch kann die Reife der Qualitätssicherung konkreter Big Data Ansätze nicht überzeugen. Da benötigte Maßnahmen zur Qualitätssicherung unmittelbar mit den durchzuführenden Datenanalysen zusammenfallen, ergeben sich massive Veränderungen für das Softwarequalitätsmanagement. Vielfach fehlt es dabei an methodischen Vorgehensweisen einer inhärent benötigten Qualitätssicherung, welche sich sowohl an konstruktiven als auch analytischen Maßnahmen orientiert. Typische Beispiele beziehen sich auf unzureichend qualitätsgesicherte Rohdaten, den unkorrekten Einsatz von Algorithmen zur Datenauswertung aber auch unzureichend begründete Schlussfolgerungen auf der Basis von Analyseergebnissen. Der Beitrag geht auf ausgewählte Aspekte der vorgenannten Bereiche ein, ohne dabei einen Anspruch auf Vollständigkeit zu erheben.<sup>1</sup>

## 1 Motivation und Ziele

Anwendungen im Bereich von Big Data bzw. Data Science benötigten ein verändertes Verständnis der Aufgaben im Softwarequalitätsmanagement. Während im Diskurs der klassischen Softwareentwicklung eine möglichst vollständige, detaillierte und unzweideutige Spezifikation der fachlich benötigten Funktionalitäten als Ausgangspunkt der Entwicklung zur Verfügung stehen sollte, findet sich diese bei Softwarelösungen zur Datenanalyse (u.a. Big Data) nicht. Die Gründe liegen in den möglichst flexibel nutzbaren Eigenschaften entsprechender Systeme. Typischerweise soll erst zur Laufzeit über aufzunehmende Daten, anwendbare Analysefunktionen oder auch adäquate Formen der Visualisierung von Ergebnissen entschieden werden. Daher kann die Spezifikation nicht zum Nachweis erfüllter Kundenanforderungen im Rahmen von Aufgaben der Verifikation und Validation herangezogen werden.

In Anlehnung an [Dumke 2003] überprüft die Verifikation die korrekte Verwendung der Big Data Lösung. Dem gegenüber bezieht sich die Validation auf die Überprüfung der Gültigkeit erzeugter Analyseergebnisse in Bezug auf den fachlich untersuchten Diskursbereich.

Grundlegend lässt sich feststellen, dass der Bezugsbereich einer Qualitätssicherung bei Big Data Lösungen in mindestens zwei Betrachtungsweisen zerfällt. 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 Verfügbarkeit, Performance, Sicherheitsmechanismen und angebotenen Funktionen zur Administration greifen, können „ad hoc“ Datenanalysen mit spezifikationsorientierten Testansätzen der Softwareentwicklung nicht mehr erfasst werden. Dennoch bedarf es auch hier der Berücksichtigung vielfältiger Qualitätsaspekte in Bezug auf die Korrektheit der aus verschiedenen

---

<sup>1</sup> Bemerkung: Der Beitrag wurde im Diskurs eines Vortrags auf der ASQT-Tagung 2017 (TU Graz) erstellt. Er stellt eine erweiterte Variante des unter [Schmietendorf 2016] veröffentlichten Artikels dar.

Quellen übernommenen Daten, die über diesen ausgeführten Analysealgorithmen sowie bei der Ergebnisdarstellung und -interpretation.

## 2 Existierende Arbeiten

Mit einer unzureichenden Reife und Qualitätsdefiziten existierender Big Data Ansätze beschäftigt sich eine von der T-Systems und der Hochschule Reutlingen durchgeführte Studien [Big Data Report 2016]. An der Befragung nahmen 609 Vertriebsverantwortliche aus mittelständischen und großen deutschen Unternehmen teil.

„Wenn es nach der Auffassung von über 71 Prozent der deutschen Vertriebsentscheider geht, dann hat der Motor noch gehörig Sand im Getriebe. Besonders die schlechte Datenqualität sehen sie als Kernproblem der Big Data-Nutzung im Vertrieb. Entsprechende Probleme entstehen vor allem bei der Einbindung eigener CRM-Systeme in Big Data-Infrastrukturen.“

[Klās 2016] räumt mit dem Vorurteil auf, dass die Quantität verfügbarer Daten wichtiger sei als die Datenqualität, wodurch Fehlinterpretationen geradezu herausgefordert werden.

„Hier bleibt es beim bekannten „Garbage In, Garbage Out“: Ein Haufen Abfall wird erst einmal nicht dadurch nützlicher, dass er besonders groß ist.“

Ein einführender Beitrag zu den vielfältigen Facetten des Testens von Big Data Anwendungen findet sich unter [ISTQB 2018]. Neben der Vermittlung grundlegender Aspekte von Big Data Anwendungen geht dieser auf den Bezugsbereich notwendiger Testaktivitäten ein. Unter Berücksichtigung des Hadoop-Technologiestacks wird auf benötigte Tests zur Validerung der Übernahme und Verarbeitung von Daten bzw. der Ausgabe von Analyseergebnissen eingegangen. Die folgende aus [ISTQB 2018] übernommene Abbildung verdeutlicht den entsprechenden Bezugsbereich potentieller Testansätze.

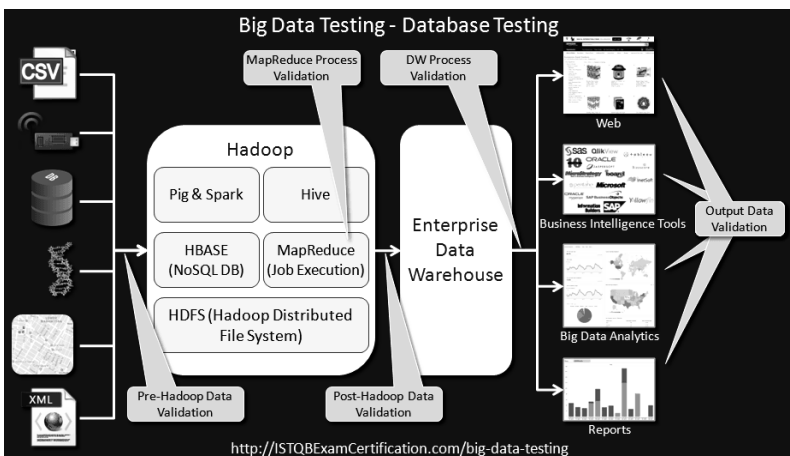


Abbildung 1: Big Data Tests (Quelle der Abbildung: [ISTQB 2018])

Darüber hinaus beschäftigen sich die Autoren mit Fragen des Werkzeugeinsatzes bzw. Ansätzen für benötigte Performancetests.

### 3 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 auf 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. die genutzten Persistenzmechanismen. Im Mittelpunkt dieser Systeme steht die Gewährleistung der Datenkonsistenz und Transaktionsicherheit.
- Contra: Gegen den Fortbestand sprechen die Skalierbarkeit, Performance, Verfügbarkeit und Kosteneffektivität potentieller Hadoop-Installationen. 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 Neudefinition 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 Performancegründen zunächst verzichtet werden. Klassisch beziehen sich diese z.B. auf Formatanpassungen, den Umgang mit unbelegten Datenattributen, die Verdichtung bzw. Zusammenführung von Daten, die Gewährleistung von Konsistenzbedingungen oder aber auf Aufgaben der Datenbereinigung. Diese qualitätssichernden Aufgaben fallen mit der Datenauswertung zusammen, wobei es entsprechend [Dong 2013] folgende Aspekte zu berücksichtigen gibt:

- *Schema Mapping* – Zusammenführung der ggf. schemarmen 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. Rohdatendaten 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 verschiedensten Quellsystemen bedarf es kontextbezogener bzw. semantischer Informationen (Metadaten) zur analytischen Verarbeitung. Diese werden benötigt, um ggf. vorhandene Muster und Zusammenhänge zwischen den Daten eindeutig zuzuordnen zu können bzw. eine empirische Interpretation überhaupt erst zu ermöglichen.

chen. Beispiele für derartige Daten beziehen sich z.B. auf Zeitstempel, einsetzbare Schlüsselattribute, bekannte Beziehungen zwischen unterschiedlichen Fachdomänen oder auch auf die Abbildung/Mapping heterogener Datenstrukturen.

## 4 Qualitätssicherung eingesetzter Analysealgorithmen

Das bloße Sammeln von Daten führt nicht automatisch zu verwertbaren Ergebnissen. Entsprechend dem folgenden Zitat von [Reinhard 2016] gilt es, ein zielgerichtetes Vorgehen zu etablieren und Fragen nach kausalen Zusammenhängen zu stellen bzw. mit Hilfe statistischer Verfahren zu beantworten.

*„Ich warne davor, einfach einen Algorithmus auf vorhandene Daten zu schmeißen und das Ergebnis zu nutzen, weil schon irgendwas rauskommt. Das kann man natürlich für einen ersten Überblick machen, aber irgendwann kommt man immer an den Punkt, eine Frage formulieren zu müssen. Was will man eigentlich wissen? Und erst wenn die Frage formuliert ist, kann ich mich damit beschäftigen, was die richtigen Daten sind, um diese Frage zu beantworten. Und habe ich diese Daten überhaupt, und wenn nicht, woher könnte ich sie bekommen?“*

Zur Umsetzung statistischer und abfrageorientierter Verfahren werden Algorithmen eingesetzt, die im Folgenden in vier Kategorien eingeteilt werden:

- Verwendung externer Service APIs,
- Query-basierte Algorithmen,
- individuell entwickelte Algorithmen,
- 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.

### 4.1 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 Sentimentanalysen, die Klassifizierung von Dokumenten oder auch auf die Bildererkennung. Konkrete Angebote finden sich mit IBM Bluemix<sup>2</sup> und den Watson Service APIs, dem Azure ML Studio<sup>3</sup> von Microsoft oder auch dem Marktplatz Algorithmia<sup>4</sup>. 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, für Testzwecke einsetzbare Musteranalysen, betriebliche Qualitätsattribute oder auch auf benötigte Informationen zum Gerichtsstand.

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:

---

<sup>2</sup> <https://console.ng.bluemix.net/catalog/>

<sup>3</sup> <https://studio.azureml.net/>

<sup>4</sup> <https://algorithmia.com>

- 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.

## 4.2 Query-basierte Analysealgorithmen

Die Implementierung von komplexen Analysealgorithmen, 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 Hive<sup>5</sup> und Impala<sup>6</sup> 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 Testerstellung bedacht werden:

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

Bezogen auf Hive bietet das Projekt HiveRunner<sup>7</sup> 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. 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 Cucumber<sup>8</sup>.

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 ist<sup>9</sup>.

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.

---

<sup>5</sup> <https://hive.apache.org/>

<sup>6</sup> <http://impala.apache.org/>

<sup>7</sup> <https://github.com/klarna/HiveRunner>

<sup>8</sup> <https://cucumber.io/>

<sup>9</sup> <https://github.com/apache/hbase/blob/master/hbase-server/src/test/java/org/apache/hadoop/hbase/HBaseTestingUtility.java>

### 4.3 Individuell entwickelte Analysealgorithmen

Der traditionelle Ansatz des im Data Science bestimmenden Hadoop-Ökosystems ist die Entwicklung individueller Analysealgorithmen. 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 an<sup>10</sup>. Alternativ liefert HBase Mini-Cluster ebenfalls verschiedene Clustervarianten mit. Verglichen mit den Query-basierten Analysealgorithmen 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 Spark<sup>11</sup>, 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 Analysealgorithmen 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 verfolgt<sup>12</sup>.

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

### 4.4 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 Analyseworkshops 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 Softwarequalitätssicherung 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.

## 5 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 in der Realität als Ausreißer präsentieren. Sie können bspw. 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. Daher ist es

---

<sup>10</sup> <https://github.com/apache/hadoop>

<sup>11</sup> <http://spark.apache.org/>

<sup>12</sup> <https://github.com/holdenk/spark-testing-base>

von entscheidender Bedeutung, Personen mit domänenspezifischem Wissen in den Test- und Qualitätssicherungsprozess mit einzubinden, bspw. durch deren Einbeziehung in aus der Softwareentwicklung bekannte Reviewverfahren.

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 dem bereits erwähnten Behaviour Driven Development die Chancen, Big Data orientierte Datenanalysetools für die Dokumentation der Tests einzusetzen. Ein für den Data Scientist passendes Werkzeug ist z.B. Apache Zeppelin<sup>13</sup>. Zeppelin bietet eine Art agiler Dokumentation, in der der Data Scientist neben der Dokumentation auch funktionierende Beispiele und Visualisierungen einbetten kann.

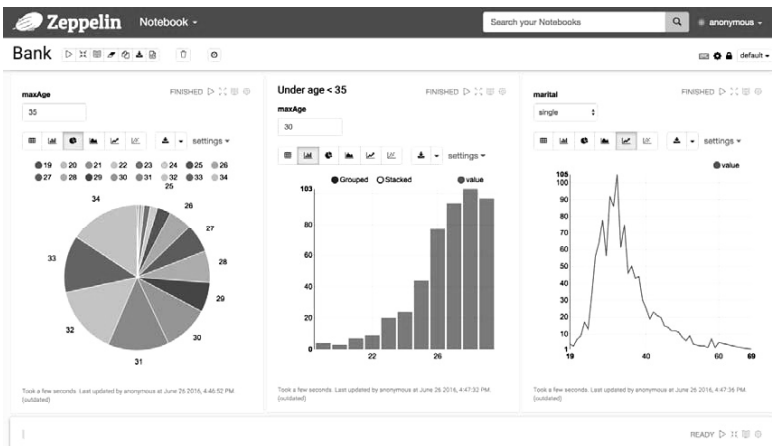


Abbildung 2: Beispielhafte Dokumentation in Apache Zeppelin (Quelle: [Zeppelin 2016])

Anhand der in Zeppelin erstellten Notebooks können Test Suites zusammengestellt werden, die wiederholbar ausgeführt werden können. Abbildung 1 stellt ein entsprechendes Beispiel.

## 6 Zusammenfassung

Testaktivitäten im Diskurs von Big Data Projekten zeigen eine hohe Diversifikation (vgl. Abbildung 3). Bereits beim Testobjekt finden sich vielfältige datenverwaltende Technologieansätze (Hadoop, Spark, SAP-Hana, NoSQL, ...). Neben der korrekten Datenverarbeitung gilt es, Architektureigenschaften wie Performance und Sicherheitsfragen in Testaktivitäten einzubeziehen.

<sup>13</sup> <https://zeppelin.apache.org/>

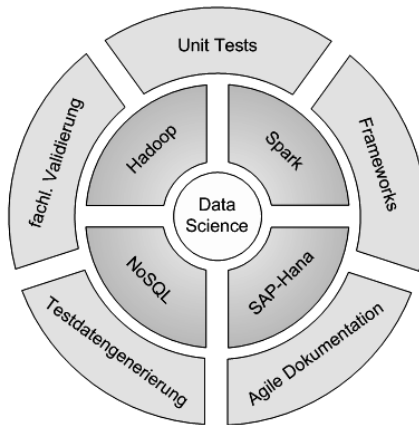


Abbildung 3: Big Data Testaktivitäten mit vielen Facetten

Im Falle der Validation und Verifikation implementierter Analysefunktionen sind algorithmische Fähigkeiten unerlässlich! Das in diesem Zusammenhang benötigte „Know-How“ erstreckt sich über das Beherrschen objektorientierter bzw. funktionaler Programmiersprachen wie Java, Scala und Python, über deklarativ orientierte Abfragesprachen wie HiveQL bis hin zur Implementierung mathematisch/statistischer Algorithmen mit Hilfe von R. Nicht immer ist das Beherrschen 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 (insbesondere domänenspezifischem) ein Erfolgskriterium. Durch die Breite der eingesetzten Methoden und Technologien und das benötigte Wissen um diese ist eine traditionelle Aufteilung zwischen Entwicklung und Test eher schwierig.

## Quellenverzeichnis

- [Big Data Report 2016] Deutsche Vertriebler hadern mit Big Data-Qualität, T-Systems/Hochschule Reutlingen, März 2016, <http://www.t-systems-mms.com/unternehmen/newsroom/detail/deutsche-vertriebler-hadern-mit-big-data-qualitaet.html>, Abruf: Jan. 2018
- [Bitkom 2014] Bitkom Leitfaden Big-Data-Technologien - Wissen für Entscheider, Februar 2014, <https://www.bitkom.org/Bitkom/Publikationen>, Abruf: Jan. 2018
- [Dong 2013] Dong, X. L.; Srivastava, D.: Big Data Integration, in Proc. Of the VLDB Endowment, Vol. 6, No. 11, August 2013
- [Dumke 2003] Dumke, R.: Software Engineering, 4. Auflage, Vieweg-Verlag, Wiesbaden 2003
- [Henschen 2012] Henschen, D.: Big Data Debate: End Near For ETL?, informationweek.com, March 2012
- [ISTQB 2018] Big Data Testing – Complete beginner’s guide for Software Testers, International Software Testing Qualifications Board (ISTQB), Abruf: Jan. 2018
- [Klās 2016] Klās, M.: Data Quality – A BIG Challenge for BIG Data, <http://blog.iese.fraunhofer.de/quantity-over-quality-mistake>, Abruf: Jan. 2018
- [Reinhard 2016] Reinhard, N.: “Leider gibt es bisher relativ wenig gute Data Scientists”: Interview mit Prof. Göran Kauermann, December 2016, <http://idalab.de/blog/interview/dr-prof-goran-kauermann-lmu>, Abruf: Jan. 2018



- [Schmietendorf 2016] Schmietendorf, A.; Hentschel, J.: Veränderte Anforderungen an die Qualitätssicherung von Data Science Lösungen, Beitrag für SQ-Sonderausgabe „20 Jahre im Dienst von Qualität“, Arbeitskreis Software-Qualität und -Fortbildung (ASQF), S. 76-83, Dezember 2016
- [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)

# Automating Micro-Service Testing

Harry M. Sneed  
TU Dresden  
Dresden, Germany  
[Harry.sneed@t-online.de](mailto:Harry.sneed@t-online.de)

Chris Verhoef  
VU Amsterdam  
Amsterdam, NL  
[x@cs.vu.nl](mailto:x@cs.vu.nl)

**Abstract:** There is currently a major trend in the direction of micro-service architectures. Micro-Services are small components – less than 1.5K statements – with a message-oriented interface that can be independently compiled and executed. The language they are developed in does not matter. They are isolated from the outer world by their interface. What matters is that they can be developed and tested within two person weeks. Their test should take no longer than five days. With a median test productivity of 40 test cases per day that means the service should require no more than 160 test cases, leaving some time for debugging and test repetition. From this point of view the size of micro-services is restricted by the effort required to test them. To keep this effort as small as possible the service test should be automated. Automation not only reduces the test effort but also enables the test to be repeated and the results compared. If micro-services are to achieve that what is expected from them, their testing must be automated. WebsTest is a tool which was designed to do just that. It not only generates requests and validates responses, but also evaluates the overall service quality. The tool has been under development for over three years and has been applied in two commercial projects.

**Keywords:** Micro- Services, REST, WSDL, Service Oracles, Service specification, Static interface analysis, Test case generation, Test result Validation, Test Metrics, Service Evaluation

## 1 Defining micro services

It is the opinion of this author that a micro service should contain no more than 1500 statements, otherwise it is not micro. It must be possible to change and retest a micro service within a week. This can only be possible if there are less than 200 paths and 100 states to test. Manually no more than 40 test cases can be tested per day. That amounts to generating up to 40 requests and validating 40 responses within a six hour period. Automatically more than 800 cases could be tested per day but the tester still needs time to check the results. For a micro service with 1200 statements and 200 test paths a tester needs at least five days to systematically test that service. These numbers have been verified earlier in testing web services [1].

There are according to Robert Binder four levels of class test coverage [2]:

- testing individual methods in a class (non-modal testing)
- testing all the combinations of methods, i.e. all paths (semi-modal testing)
- testing individual methods with all their states (quasi-modal testing)
- testing all sequences of methods with all combinations of their states (modal-testing).

The same what applies to classes also applies to micro services. The minimum goal should be to test all combinations of operations contained within the service [3]. The maximum goal is to test all sequences of methods with all relevant states. The number of operation sequences can be extracted from the code via interface analysis. The number of method and state combinations is equivalent to the number of execution paths through the service. Each condition on a path has to be set once true and once to false to ensure that all states are tested. The number of test cases required for this level of coverage is roughly the sum of all branches times 2. If there are 10 operations in a service with an average of 4 branches per operation that adds up to

40 Branches \* 2 = 80 test cases.

Both branches and paths can be counted via a static analysis of the service code. This will inform the tester how many test cases are required to achieve either minimum or maximum coverage of the service under test. The size of a micro service should not exceed the minimum number of statements which can be tested within four tester days. It must be possible to code and test a micro service within one sprint [4].

## 2 Goal of micro-service testing

When micro services are kept small < 1.5K statements they can be readily tested within a short time <= 5 person days. This requires that the services not only be small but that their test is automated. For that they require test drivers to test them independently of their clients and the environment in which they will be operating in. Of course it is possible to create a client, e.g. a BPEL process or a Java script application to invoke the services, thus testing them indirectly via the client, but just as with unit testing it is more efficient and more effective to test a service directly under a driver. The test driver gives testers more control over the test object and allows them to test all of the features of that object instead of only those used by one particular process [5].

By using a test driver, testers can refer to an oracle to determine what test data should go in and what test results should come out. If an error occurs testers can be sure that the error is in the service and not in the environment. When testing by means of a real client, testers cannot control the interaction between the client and the service, and should an error occur, they cannot be sure whether the error is in the client, in the service or in the middleware between the two. The task of a service test driver is to control and monitor the sequence of operation requests within a single service. For testing service integration other tools are needed which focus on the sequence of service execution and the data exchange between the services. The task of the service integration driver is to drive a sequence of requests to different services and to monitor the flow of data between the services. The two tools have many functions in common but some which are unique to that particular mode of testing.

A micro-service testing tool serves the same purpose for services as JUnit or NUnit do for code objects [6]. It gives the testers control over the test object and allows them to observe its behavior by simulating the environment in which the object is operating. The difference is that the testing of objects is a white-box test, i.e. the testers have insight into the code and can derive their test cases from the structure of that code, whereas the testing of services is a true black-box test. Test cases are derived either from the service interface description or from a specification of the service. In conventional unit testing, testers can also trace the tests through the code and measure the degree of test coverage based on the amount of code executed. In testing services testers cannot see what is going on inside the service. It is hidden behind the service interface. They can only see what goes in and what come out.

If test and development are to be separated as recommended here new means of measuring test coverage are required as well as new ways to determine correctness. Testers will most likely never see the source code of the service under test. It should not be their business to investigate how the service is implemented. It is not their job to know how the service functions internally. It is even better if they do not know. Their job is to determine if the service satisfies the requirements of the product owner they represent, just as real estate brokers are obliged to see if a house satisfies the requirements of their client. Only the results count. The test of a micro- service is aimed at validating the results of the service when used by the clients of that particular application. The services should behave correctly and effectively in accordance with the functional and performance criteria stipulated in the service requirement specification as well as in the pertinent service level agreement [7].

### 3 Deriving test cases from an oracle

Every test is a test against something. That something is referred to as the test oracle [8]. In a unit test the developer tests the module or the object under test against a module or object specification. In an integration test the integration tester tests the interaction of several modules or objects against the interface specifications. In a system test, the system tester tests the system as a whole against the requirement specifications. In the unit test, the test cases are derived from the code. In the integration test, the test cases are taken from the interface definitions. In a system test, test cases are extracted from the requirement texts and the user interfaces. Micro-services have the advantage that their requirement specs are also limited in size and can be more easily processed. The number of requirements and use cases is limited.

When testing a micro-service, test cases are derived either from the service interface definition, e.g. WSDL or WADL, or from the service requirement specification in English or some DML – Domain Specific Language. If taken from the interface definition the service can only be tested mechanically against random values generated from the data type definitions. There is no real way to validate the correctness of their results, since random testing also produces random results [9]. This type of random testing is not adequate to determine the suitability of a service. A requirement specification is needed, one which can be readily analyzed to derive meaningful test cases. The service requirement specification – SRS - supplements the interface definition by the requirements on the service, not only the functional requirements, but also the non-functional requirements such as time constraints, precision of results, security checks and transaction loads [10]. It is the job of a service tester to determine:

- a) if the service under consideration fulfills the requirements of the potential user, and
- b) if the service fulfills those requirements according to the constraints imposed by the service level agreement.

Thus, there are three sources from which service test cases can be derived:

- 1) the service interface definition (WSDL/WADL)
- 2) the service requirement specification and
- 3) the service level agreement

In the case of the service interface definition, a tool is needed which parses the interface code to recognize the operations and their parameters. The parameters are assigned data types defined in an XML schema. From these data types random values can be generated such as whole numbers for integers and letter combinations for strings. Such data will certainly not trigger a proper path through the service but it will test if the service reacts.

In the case of the service requirement specification, a tool is needed which can parse the natural language text and identify the use cases. The challenge here is to associate the use cases prescribed in the specification with the operations defined in the service interface definition. The function and data names used in the requirement text must be matched with the operations and parameters of the interface definition. If that is achieved it becomes possible to generate different functional test cases for each use case with sample data values. Each of these test cases will trigger a request to the service and lead to a specific response. The tester should be able to manipulate the input data and to validate the output results. That is the purpose of the pre- and post-condition assertions [11].

For the test of a single service that might not be too difficult, but for the test of multiple services it becomes complicated as will be shown later. In the case of the service level agreement, a tool is needed which can process the non-functional requirements and construct cases to test them. If there is a response time limit then cases are needed which will test that limit. If security checks are prescribed test cases must be constructed which will trigger those

checks. The important thing here is that some numeric boundary value is given, i.e. for time, for the number of transactions and for storage capacity. The tool is intended to monitor the performance of the service and to check it against those limits. For services intended for a cloud the quality features are just as important as the functional features in assessing the suitability of the service.

Micro-service testing tools are intended to support the tester in this endeavor. They should enable him to set up and conduct the test of any service with a minimum of effort while obtaining a maximum amount of information for determining the suitability of the service for the application at hand. To that end the tools should use both the web service interface definition and the service requirement specification as a basis for generating the requests and for validating the correctness of the service responses [12]. They should also measure the performance of the service and determine with reference to the service level agreement if the non-functional requirements are met. Finally, they should execute the test in a fully automatic mode, even without the presence of the tester. The test of a service should be automated to the greatest extent possible so as to allow the testers to achieve their goal of evaluating a given service at a low cost within a minimum time [13].

## **4 Functions of the webtest tool**

The two primary goals of any service testing tool are to find out whether a web service can be trusted, if its technical performance is adequate, and if it is suitable for the purpose of the user as stated in the service requirement specification. All functions performed by the tool are subordinate to these three main goals,

- The extraction of logical test cases,
- The generation and editing of test scripts,
- the generation of test requests,
- the execution of the service under test,
- the validation of the service responses and
- the evaluation of the service test results.

All of these sub functions contribute to the goals of gaining trust and assessing the suitability of the service in question. For that a service has to be comparable and for comparison numeric data is needed.

### **4.1 Extracting Logical Test Cases**

The starting point for a service test is the service requirement specification - SRS. Here it should be stated what functions are expected to be fulfilled by what service. It should also be stated how those functions are to be fulfilled, i.e. their qualitative requirements. The requirements can be formulated in a natural language type structured text; however the operation names as well as the parameter and result name should coincide with the names defined in the web service interface definition (WSDL). Otherwise, the service requirement specification cannot be associated with the service interface definition. It is the operation and data names which bind these two documents together.

If the test is to be a functional, requirement based test, the requirements have to be analyzed in order to gain a base of comparison. On the one hand the service specified is compared with the service offered. On the other hand, the requirement test cases have to be extracted from the requirement text. Every test requires an oracle and in this case the oracle is the service

requirement specification. The service specification, or service level agreement, should not only define the interface but also the data objects referred to, the processing rules and the use cases. It is used here as a basis for the test [14].

The WebsTest tool includes components for analyzing the service requirement specification either in English or in German language. It generates a test case table from the requirements, rules, objects, interfaces and use cases with a test case for every action, state, rule and use case step. These logical test cases are then stored for further processing by the tester. If there is no requirement specification, then the test has to be based on the structure of the data in the interface definition. In this case the requirement analysis is skipped over. (see Figure 1)

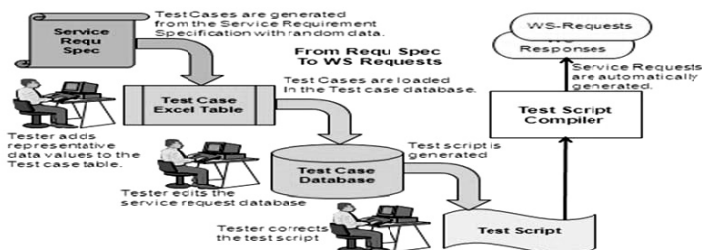


Figure 1: From Service Requirements to Service Test Cases

## 4.2 Static Analysis of the Service Interface

The next function is the static analysis of the service interface definition. It will be contained in a separate web service interface definition document – WSDL or WADL. The web service interface definition is based on a schema which describes the structure and types of the service data. This document needs to be statically analyzed before testing begins, since it may turn out that the web service interface does not conform to the interface coding standards or that it does not fit to the requirements at all. This should be known before testing begins. It is also recommended to measure the service interface and to check its static quality before beginning with the test. This function is performed by the WSDL static analyzer WSDAUDIT. This tool parses the WSDL interface to check the conformance to rules, to compare the operations with those specified in the requirement document and to measures the size, complexity and quality of the interface definition. Comparison is made on the basis of names and types. If the names and types in the SRS differ from those in the WSDL, the service is considered unsuitable. Depending on the outcome of the analysis, it may be decided already here that it is not worth dealing further with this particular service [15].

## 4.3 Generating and Editing Test Scripts

After analyzing the service interface definition the testers should proceed with the test. To make a functional test, the logical test cases have to be merged with the service interface definition. To make a structural test, it is enough just to take test cases out of the interface definition. Thus, the function of preparing the test script has two alternate paths – one in which the test case table is derived from the interface definition with the tester filling in the test data out of the top of his head, and one for merging the logical test cases and the test values contained therein with the service interface definition. The end result is in both cases a test case table with specified test data values for all operations offered by the service. From this test case table, the test script is automatically generated to include the test values. If there are no test values specified, then random values will be generated based on the data types. In

any case the generated test script will contain data values for both the arguments and the results [16].

The following sample of the generated test script shows the test of a single operation in one service.

```
service: BausparerSOAPServerservice;
// It should be possible to select a Bausparer by means of the // ground number
and the BBZS
if (testcase = „Bausparer001“);
    if ( operation = „Bausparer“);
        if ( request = „Bausparer1Request“);
            if ( object = "Person" );
                assert inp.CRole = "Besitzer";
                assert inp.nGrdNr = „4711“;
                assert inp.bBBZS = „120036“;
                assert inp.tsReadTime = „201303130700“;
                assert inp.iSeqNumber = "100";
            endObject;
        endRequest ;
        if ( response = „Bausparer1Response“);
            assert out.$ResponseTime < „1200“;
            if ( object = "return" );
                assert out.ResponseCode = „00“;
                assert out.cNachname = „Schmidt“;
                assert out.cVorname = „Karl“;
                assert out.dGeburtsdatum = „19220420“;
            endObdeterminedject;
        endResponse ;
    endOperation;
endCase;
```

#### Sample 1: Micro Service Test Script

Once the test scripts are generated the tester has the possibility to edit them. If the script is generated from the test case table and that table already includes representative test data values, those values will be carried over into the script. If not, random values are generated which the tester can overwrite in the script. The data values given should reflect real data with numbers, codes and alphabetical strings, and they should vary from one test case to another. Since the number of data combinations is practically unlimited, there has to be some kind of artificial limit imposed, which means the data combinations selected should be representative of the possible data combinations. This requires a combination of controlled data assignment and random data generation [17]. It might be added that the tester can also give in values which conflict with the data type to test the exception handling.

#### 4.4 Generating Service Requests

From the test script requests are generated, one for each operation invoked in each test case. The number of requests is the number of test cases times the number of operations referred to. Each request contains a different set of input parameters in a standard XML format. Each time a new request is generated, the parameter values are altered. In the case of numeric ranges the boundary values are set. In the case of value classes, representative values are taken, one for each class partition. In this way the requests represent varying combinations of input data. The requests are placed in a queue to be dispatched sequentially to the service under test. (see Sample 2)

```
<request operation="BuySomething" requestid="Request_001">
  <AnOrder type="struc">
    <CustNo type="string">009999</CustNo>
    <OrderNo type="int">001000</OrderNo>
    <DeliveryMode type="string">RAIL</DeliveryMode>
    <PaymentMode type="string">POD</PaymentMode>
```

```

<Orders type="array" occurs="01">
  <item type="struc">
    <OrderArtNo type="string">004711</OrderArtNo>
    <OrderArtType type="string">BOOK</OrderArtType>
  <OrderArtName Type="string">A_Book</OrderArtName>
  <OrderArtAmount type="int">5</OrderArtAmount>
</item>
  <item type="struc">
    <OrderArtNo type="string">004712</OrderArtNo>
    <OrderArtType type="string">MAGA</OrderArtType>
    <OrderArtName type="string">News</OrderArtName>
    <OrderArtAmount type="int">100</OrderArtAmount>
  </item>
</Orders>
</AnOrder>
</request>.

```

Sample 2: Generated Web Service Request

#### 4.5 Executing the Service under Test

Once the service requests are generated, it is possible to execute the target service by sending off messages to its internet address. For that the service request has to be packed into a special envelope – a SOAP protocol – for transmission. Then, when an answer comes back, that response has to be unpacked and made available. Packing, sending, receiving and unpacking internet messages are the functions of the web service test driver. There has to be one for WSDL and another for REST messages [18].

#### 4.6 Validating Service Responses

It is not enough to confirm that the web service under test functions and that it returns some response. The responses need to be validated to ensure that they contain the proper return values. The proper return values are specified in the test script. That means the responses have to be matched to the requests and they have to be compared with the expected responses. Every response value which deviates from the expected value is considered to be an error. The actual response time of each response is also compared with the maximum allowable response time. Those response times which exceed their limit are also considered errors. The tool produces a validation report on the incorrect results and the exceeded response times. In addition it produces an XML file with the test metric to be passed on to the test evaluation. (see Figure 2)

| WSDL Response Validation Report              |                               |
|--|-------------------------------|
| Tester: iWareHouseWeb Service                | Date: 04.09.12                |
| TestName: WSDL-Response                      | System: Orders                |
| TestCase: 001                                |                               |
| Non-Matching Params                          | Non-Matching Values           |
| RespId: GetTypes1Response_001_return         |                               |
| lst: item[2]                                 | BOOK                          |
| SoI: Asserted_Value                          | NEWS                          |
| lst: item[3]                                 | 0                             |
| SoI: Asserted_Value                          | B=BOOK                        |
| lst: item[1]                                 | Empty                         |
| SoI: Asserted_Value                          | M=MAGA                        |
| lst: Price[2]                                | 23.50                         |
| SoI: Asserted_Value                          | =25.50                        |
| RespId: GetTypes1Response_001                | missing from the old Response |
| Total Number of old Responses:               | 05                            |
| Number of old Responses matched to new ones: | 06                            |
| Total Number of Params checked:              | 13                            |
| Total Number of non-Matching Params:         | 16                            |
| Percentage of matching Params:               | 11 %                          |
| Percentage of matching Responses:            | 80 %                          |

Figure 2: Service Response Validation Report



## 4.7 Evaluating Service Quality

The ultimate task of web or cloud service testers is to assess the suitability of a particular web service. They have to judge whether the service is adequate for the task at hand or not. For this decision they need information; information on the performance, the functionality, the reliability and the security of the service in question. The testing tool provides this information in form of reports and graphics which can be easily interpreted by the tester. The ultimate evaluation report is the service tester's dashboard with 12 quality gauges for modularity, conformity, interoperability, reusability, testability, test case coverage, functional coverage, data coverage, correctness, reliability and efficiency. The top six gauges are derived from the static analysis of the interface definition, the bottom six from the dynamic analysis of the service itself. Each gauge shows the ratio of fulfillment on a scale of 0 to 1. There are three color fields to depict unacceptable, acceptable and good ratings. The good ratings are in the green field, acceptable ratings in the yellow and unacceptable rating in the red field. These individual ratings are intended to help the tester in evaluating the overall quality of the service. (see Figure 3).



Figure 3: Service Evaluation Dashboard

## 5 Experience with automated service testing

Experience with the automated testing of web services has demonstrated that many deviations from the oracle can already be uncovered by statically comparing the service interface with the service requirements. Those remaining are for the most part uncovered in the dynamic analysis. Of 10 bugs planted into a trial service 5 were uncovered by static analysis and 3 were found by comparing the contents of the responses. The remaining two bugs could only have been found by analyzing the database and that is beyond the scope of this service testing tool [19].

The scope of this tool is limited to testing individual micro services in a black-box test. It does not test sequences of services and it has no insight into the backend storage of the service under test.

## 6 Future work

Future work on the tool WebsTest will concentrate on improving the quality of test data. Currently it is up to the testers to assign test data values based on their interpretation of the specification. This approach depends very much on the domain knowledge and experience of the testers. For the future it is planned to include value domain specifications in the service requirement specification which define possible values for particular parameters [20]. This

will increase the degree of automation but also add to the work required to produce the specification. The more testing is automated the more knowledge has to be captured in the test specification. Somewhere the exact behavior of the test object has to be described. Whether it is better to define everything in one document remains to be seen. The problem with such an oracle is that it also has to be proven correct.

## References

- [1] Sneed, H.: “Testing Web Services in the Cloud” in Software Testing in the Cloud, Ed.: S. Tilley & T. Parveen, IGI Global, 2013, p. 136-172
- [2] Binder, R.: Testing Object-oriented Systems, Addison-Wesley, Reading MA., 1999, p. 455-496.
- [3] Newman, S.: Building MicroServices, O’Reilly, Boston, 2014, p. 2
- [4] Wolff, E.: Micoservices – Grundlage flexibler Softwarearchitekturen, dpunkt verlag, Heidelberg, 2016, p. 31
- [5] Tsai, W.T., Zhou, X., Chen, Y., 2008, On testing and evaluating service-oriented software, *IEEE Computer Magazine*, (August, 2008), p. 40
- [6] Beck, K., 2003, Test-driven Development by Example, Addison-Wesley, Boston, 2003, p. 42
- [7] Tilley, S./ Parveen, T., 2013, Software Testing in the Cloud – Perspectives on an emerging Discipline”, IGI Global Publishers, Hershey, 2013
- [8] Howden, W. 1987, Functional Program Testing & Analysis, McGraw-Hill, New York, 1987, p. 11
- [9] G. Dai, X. Bai, Y. Wang, and F. Dai, 2007, Contract-based testing for web services, in COMPSAC 2007: Proceedings of the 31st Annual International Computer Software and Applications Conference, vol. 1, Beijing, China, July 2007, IEEE Computer Society, pp. 517–526,.
- [10] Peters, D./ Parnas, D., 1998, Using Test Oracles generated from Program Documentation, *IEEE Trans. on S.E.*, Vol. 24., No. 3, March, 1998, p. 161
- [11] Martin, R, 2005, The Test Bus Imperative - Architectures that Support Automated Acceptance Testing. *IEEE Software Magazine* 22(4): July 2005, p. 65
- [12] Sneed, H., 2008, Certification of Web Services, Workshop on Service-oriented Architecture Maintenance – SOAM, CSMR2008, Athens, March, 2008, p. 336
- [13] Canfora, G. & DiPenta, M., 2006: Testing Services and Service-centric Systems – Challenges and Opportunities. *IT Professional*, Vol. 8, March 2006, p. 10.
- [14] C. Ma, C. Du, T. Zhang, F. Hu, and X. Cai, 2008, WSDL-based automated test data generation for web service, in CSSE ’08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, Dec. 2008, IEEE Computer Society, pp. 731–737.
- [15] Gebhart, M., 2013, Design of Services with SoamL – an engineering approach to specifying web services, *Object Spectrum*, Feb. 2013, p. 42.
- [16] Sneed, H., Huang, S., 2006, WSDLTest - a tool for testing web services, in WSE ’06: Proceedings of the Eighth IEEE International Symposium on Web Site Evolution, Philadelphia, PA, USA, Sept. 2006, IEEE Computer Society pp. 14–21.
- [17] Offutt, J., Xu, W., 2004, Generating test cases for web services using data perturbation, *ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 5, pp. 1–10.
- [18] Martin, E., Basu, S., Xie, T., 2007, Automated testing and response analysis of web services, in Proceedings of the 2007 IEEE International Conference on Web Services, pp. 647–654, Salt Lake City, UT, USA, July 2007, IEEE Computer Society.

- [19] Sneed, H., 2008, Certification of Web Services, Workshop on Service-oriented Architecture Maintenance – SOAM, CSMR2008, Athens, March, 2008, p. 336
- [20] Bertolino, A., Gao, J., Marchetti, E., Polini A., 2007, Automatic test data generation for XML Schema-based partition testing, in AST '07: Proceedings of the 2nd International Workshop on Automation of Software Test, Minneapolis, Minnesota, USA, May 2007, IEEE Computer Society, pp. 4–14.

# Business Rule Recovery as a Fata Morgana

Harry M. Sneed  
harry.sneed@t-online.deed

**Abstract:** The wish and the need to recover the business logic from existing systems may be driving the reverse engineering research community in a wrong direction. The countless number of papers on this subject compared with the meager results would tend to indicate this. The question whether it is even possible to gain any useful business information from low-level code has always been a subject of dispute. To answer that question one must go back to how the code is produced in the first place. The original programmer will have been given some design document such as a class diagram or a structogram supplemented by some written text. In today's agile world he or she may have been given a user story. Then he or she will have become involved in a discussion with the user about the detailed requirements of the task to be accomplished. He may produce a prototype solution and try it out until it comes close enough to what the user wants to turn it over to him. After the turnover he continues with the fine tuning of the code until the user is adequately satisfied. Software development, i.e. coding is seldom a precisely planned process. It is a game of trial and error.

## 1 Top-Down Stepwise Refinement

It may be that the developer is given a list of business rules that he should fulfill. It will, however, be seldom the case that he transforms those rules 1:1 into the target programming language. There will be more or less interpretation involved. If the developer uses the same names in the code as appear in the rule text the code for that rule will be recognizable, but even this is not to be expected., Especially in older code programmers tended to use short abbreviated names which only resemble the prosa names contained in the rule text., In short, the code will more likely be scrambled and far removed from the original specification. Perhaps the developer will use only if statements to express simple first order predicate logic. However if the logic is more complex he may resort to using some kind of case statement with many branches. It is at this point that programming becomes an art as defined by Donald Kruth in his classical publication on the Art of Computer Programming – the programmer as an artist. The programmer may have a pattern in mind which helps him to structure his code. Once the structure is in place, the programmer can start to refine it.

## 2 Bottom-Up Stepwise Backtracking

In recovering business rules from code, it is a big help if the original English language rules are still available. By comparing the nouns in the English text with the operands in the code, it is possible to identify which blocks of code statements belong to which rules, If no such texts are available, it becomes difficult., The best one can do is to select a given output variable and to track the assignment or computation of that variable back through the code to the point of entry. What we then have is a code slice which leads to a given result. This can be interpreted as being the rule for that particular result, e.g. the monthly interest payment. How these rules are presented is another matter. As a start the program slices with their conditions and assignments can be presented in the original programming language. This is the simplest and for the programmers who work with that language the easiest to understand. It is also the least distorted rule description. The person dealing with the code receives a narrow view of the processing steps leading up to the final result.

```

| 05 W1-WORK.W1AN-BLZ PIC X(8).
| SET IN B12-PRUEFEN-EINGABE:
|+-----> PERFORMED BY LA031K03:
| 1 IF (
| 1 L1AX-BLZ NOT NUMERIC
| 1 );
| 2 MOVE L1AX-BLZ TO W1AN-BLZ;
|

```

If the user of the rules is not familiar with the original language it will be necessary to convert the rules into another more general language. The logic path remains unchanged – the conditions and assignments are the same, but they are expressed in another higher level design language such as OCL or Gherkin. This is what the more academically inclined researchers prefer. The IF and LOOP conditions are expressed in the other language. This may cause the original maintenance programmers to complain since for them it is a step backwards, to a restricted form of expression. Instead of a single EVALUATE statement in COBOL they have a sequence of many IF conditions and since there can be no ELSE statements in business rule predicate logic, the else's will be replaced by ifnot's. This can make the conditions even harder to understand but they are separated from one another and can be processed independently.

GIVEN: Function = B12-PRUEFEN-EINGABE:

IF: If L1AX-BLZ NOT NUMERIC

DO: MOVE L1AX-BLZ TO W1AN-BLZ;

A third form of rule expressions is natural language. Rule English is a subset of the English language designed to express rules in standard English using the constructs:

if <condition> then <action>

If variable L1AX.Bankleitzahl is not numeric

Set variable W1AN\_Bankleitzahl to L1AX.Bankleitzahl;

This is the approach suggested by the Rule-English Method proposed by the OMG in their SBVR standard for describing business rules. (see OMG Document Number: formal2009-12-01 <http://www.omg.org/spec/PRR/1.0>).

A fourth form of presentation is to express the business logic with decision tables.

| Conditions          | Decision Rules |   |   |   |   |   |   |
|---------------------|----------------|---|---|---|---|---|---|
| W1A8-RICHTIG        | Y              | - | - | - | - | - | - |
| L1AX-BLZ NUMERIC    | -              | N | - | - | - | - | - |
| L1AX-KONTO NUMERIC  | -              | - | N | - | - | - | - |
| L1AX-PRUEFZ > SPACE | -              | - | - | Y | N | - | - |
| W1A8-FALSCH         | -              | - | - | - | - | Y | - |
| W1AN-FETCH = ZERO   |                |   |   |   |   |   |   |
| L1AX-RCODE = '00'   |                |   |   |   |   |   |   |
| SQLCODE = ZERO      | :              |   |   |   |   | N |   |
| WHEN SQLCODE = ZERO |                |   |   |   |   |   |   |
| WHEN SQLCODE = 100  |                |   |   |   |   |   |   |
| WHEN OTHER          |                |   |   |   |   |   |   |
| SQLCODE = ZERO      | -              | - | - | - | - | - | - |
| Actions             |                |   |   |   |   |   |   |
| LA031K03-1          | X              |   |   |   |   |   |   |

|                       |   |   |
|-----------------------|---|---|
| B11-INIT-1            | X |   |
| B12-PRUEFEN-EINGABE-1 |   | X |
| B12-PRUEFEN-EINGABE-2 |   | X |
| B12-PRUEFEN-EINGABE-3 |   | X |
| SubTable-1            |   | X |
| B30-VERARBEITUNG-1    |   |   |
| B30-VERARBEITUNG-2    |   | X |
| B30-VERARBEITUNG-3    |   | X |
| B30-VERARBEITUNG-4    |   | X |
| B30-VERARBEITUNG-5    |   | X |
| B30-VERARBEITUNG-6    |   | X |
| B30-VERARBEITUNG-7    |   | X |
| B30-VERARBEITUNG-8    |   | X |
| B40-AUFRUF-SX945A-1   |   | X |

### 3 Crossing Module Boundaries

The slicing of code to recover rules is difficult enough when dealing with a single module. It becomes even more difficult when dealing with module hierarchies. Legacy Systems in COBOL or PL/I often consist of a module hierarchy. The main module is started by the job control in batch processes or by the teleprocessing monitor in dialog processes and it passes control to the lower level modules via call operations in which parameters are passed. In PL/I such intermodular calls are distinguished from internal procedure calls by the fact that the entry name is declared as an external entry. In COBOL external modules are invoked by a "CALL" statement as opposed to the "PERFORM" statement for invoking internal procedures in form of "SECTIONS" and "PARAGRAPHS". The CALL statement passes parameters, i.e. addresses of data, to the called module. In both COBOL and PL/I the CALL of an external module is a call by reference.

Using the CALL statement, developers can build up a hierarchy of modules. Data declared in an upper level module can be passed on for further processing in lower modules. This feature is used extensively in many old COBOL applications. Most of the batch processes and the majority of the online transactions consist of several modules connected by call statements with common parameters. Many of these modules are small service routines but they have to be included to make a complete many data analysis. If the path to a data result of the system is to be traced completely it must be traced from the point of origin to the point where it is written out or displayed. This path can transcend several modules. In one module it is initialized, in another module it is stored in the database. In online transactions data variables are often read in from the user interface in an I/O module and passed on to other modules to be processed before being returned for display by the I/O module. Physically the data remains at the same location, but logically it wanders from one module to another. It may even have different names in the different modules where it is used but remains in the same location and must therefore be considered the same data.

Decisive is not the name but the position in the parameter list of the connecting call statement. In the calling module the name "PRICE" may appear as the third parameter in the list

MODULE MODA:

CALL MODX USING ITEM-NR, AMOUNT, PRICE, DATE

In the called module the same data variable appears in the entry parameter list under a different name, but at the same position

## PROCEDURE DIVISION USING ARTICLE, SUM, COST, DATUM

Here in the called module COST is equivalent to PRICE. It must have the same type and be at the same location, but it can have another name. Thus in flowing through several modules the same variable may take on several alias names. The code analysis tool must note the alias names and qualify them with the module name

MODA.PRICE = MODX.COST = AMOUNT-DUE

The operations in Module-A on PRICE have to be joined with the operations on COST in Module-X and the operations on AMOUNT-DUE in Module-Y. Only by joining the data flow slices of all modules is it possible to construct a complete business rule. Otherwise one only has partial rules, i.e. a rule for each module.

The approach taken by the author is to first analyze each module separately to create the partial rules as text fragments. The name of the target result and the names of variables is assigned directly to it are matched against the parameter list of the module where the result is set or computed. If a match occurs the name of the corresponding variable is the parameter list of the calling module is registered as an intermediate result. The matching with the calling module is made not by name but by the position of the variable in the parameter list. The two code slices, that of the called and that of the calling module are joined together to create a compound rule list in which the assignment of values to the target result is traced back through the preceding module.

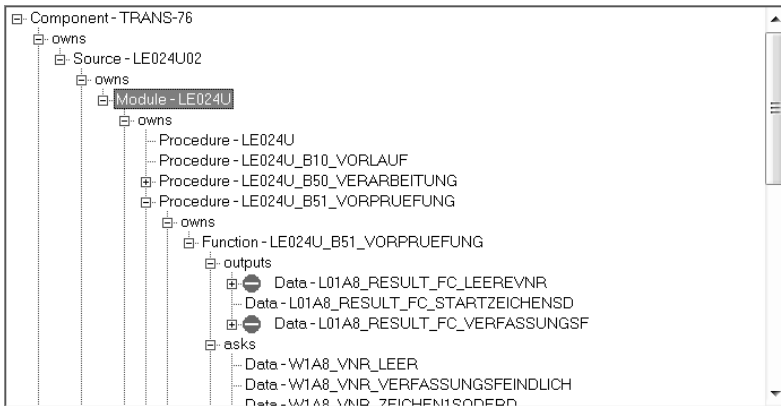
Since there can be any number of preceding modules the processing is recursive. It is repeated for every superordinate module in the calling hierarchy going back to the main module, i.e. that module which is not called by any other. This module is the origin of the data flow path leading to the final result. It may transcend many modules and contain an intermediate result under a different name for each module traversed. What starts out with the name Amount-Due in the main module may end with the name Computed-value in the lowest level service module. By matching the parameter lists of the CALL statements with the parameter lists of the ENTRY statements it is possible to identify the same variables.

In legacy mainframe systems it is a common practice to pass over long parameter lists to the subordinate modules. This has to do with the architecture of those applications. The program control blocks, the screen contents and the database access paths are passed from module to module in a chain. For instance, the inputs submitted by the user are passed down to be processed and the contents of the database are passed up to be displayed. In between the data can be altered at different levels. Somewhere the result being traced is assigned to a report or an output screen where it is displayed to the user. This is where the data flow path ends. The challenge here is to recognize the fields of an output mask and to locate the statements which assign value to them. This is accomplished by matching the output variable names with the names of the fields in the output map.

It can very well be that the same data is initialized in one module, processed in the next, accumulated in the third and presented in a fourth module, In each module the data will be partially transformed and in the end the data stored. The business rule, i.e. the path to the final result will be split up among the fourth modules. In each of the four modules values will be assigned to the target variables under given conditions. The business rule is the concatenated string of all assignment statements and their conditions. It tells to the user how a given output value comes to be what it is, not only in one module but in all modules.

## 4 Less is More

Slicing purists will claim that all variables leading into a selected result should be followed back to their origin like tributaries flowing into a bigger river. In principle this can be done. Each variable whose value is joined with the target value can be traced back across module boundaries. Even within a single module the data flow branches multiply quickly. If a module is big such as those in mainframe legacy systems the number of data flow paths quickly exceeds the limits of what a human being can comprehend. If the goal were to test all data flows through the module that would be a goal to strive for. However, the goal here is to depict how an output is derived. By including all incoming data flows one only clouds the issue. The resulting rule documentation with its many data flow paths quickly becomes incomprehensible. Such a complete data flow analysis is only possible with small modules having less than 20 data variables or when the rule display is interactive. The user must have the possibility of telescoping the html files, i.e. selecting one of the many data flow paths and following it back through the network of data nodes. This is a feature of the SoftRepo Repository System. The user can select some node to start from, either a data element or a function element and click his way through a network of procedure and data nodes. This way he can learn a lot about the technical contents of the code but he is still far away from the original business problem.



## 5 Given the Solution what was the Problem

This motto of the first international workshop on Software Reverse Engineering in 1991 has haunted the software maintenance community for the past 27 years. Users with legacy systems would like to recover the original requirements of those systems. However, it is not an easy goal to reach. The business functions are intertwined with technical housekeeping functions making it impossible to pull them out individually. One can generate all kinds of views of the code – tabular and graphic – but one cannot generate a view of the pure business logic. That remains a research challenge for the future.



# Six-Sigma-getriebene Optimierung des Applikationsmonitorings

Stephan Tost  
stephan.tost@soprasteria.com  
Sopra Steria Consulting, Hamburg

**Abstract:** Six Sigma ist eine systematische Vorgehensweise zur Prozessverbesserung und gleichzeitig ein statistisches Qualitätsziel. Beides dient dazu, Prozessabläufe bzw. deren Ergebnisse nahezu fehlerfrei zu gestalten. Dazu werden Geschäftsvorgänge mit statistischen Mitteln beschrieben, gemessen, analysiert, verbessert und überwacht. Dieses Vorgehen orientiert sich streng an den Kundenbedürfnissen.

Im vorliegenden Projektbericht wurde diese Methode dazu genutzt, das Monitoring einer IT-Anwendungslandschaft strukturiert und transparent zu verbessern. Ausgehend von einem Problem Statement und einer Zieldefinition wurde der Ist-Zustand aufgenommen und eine messtechnische Baseline gezogen. Auf Basis der konkreten Kundenbedürfnisse wurden Verbesserungsmaßnahmen definiert, bewertet und umgesetzt. Im Endeffekt wurden bereits durch die erste Stufe der Umsetzung erhebliche Verbesserungen erzielt, die innerhalb von weniger als sieben Monaten zum ROI führten.

## 1 Einführung

### 1.1 Methodik

Six Sigma (siehe für Details zur Methodik [1], [2], [3]) ist eine systematische Vorgehensweise zur Prozessverbesserung und gleichzeitig ein statistisches Qualitätsziel. Beides dient dazu, Prozessabläufe bzw. deren Ergebnisse nahezu fehlerfrei zu gestalten. Dazu werden Geschäftsvorgänge u. a. mit statistischen Mitteln beschrieben, gemessen, analysiert, verbessert und überwacht. Dieses Vorgehen orientiert sich streng an den Kundenbedürfnissen. In Six Sigma-Projekten wird üblicherweise der DMAIC-Zyklus (Define / Measure / Analyze / Improve / Control) angewandt, um das Projekt strukturiert abzuwickeln. Außerdem wird eine Vielzahl von Werkzeugen definiert, um die verschiedenen Phasen optimal zu unterstützen.

Der Zusatz „Lean“ (Lean Six Sigma) ergänzt den Six Sigma-Ansatz um Aspekte des Lean Managements, also die stringente Reduktion von Verschwendung im Ablauf von Prozessen bzw. der Herstellung von Prozessergebnissen (z. B. Produkten).

### 1.2 Konkretes Projekt

Das hier beschriebene Six Sigma-Projekt wurde methodisch am DMAIC-Zyklus mit nachfolgender R-Phase (für „Reinforce“) ausgerichtet. Das Projektteam bestand aus dem Autor selbst, einem Kollegen im Projekt (nicht aktiv in das Thema Six Sigma - wohl aber in die operative Umsetzung - eingebunden), und einem Six Sigma-Master Black Belt, welcher den Autor punktuell beratend unterstützte.

Das Projekt war in Umfeld eines IT-Anwendungsbetriebsteams angesiedelt, in welchem Sopra Steria Consulting seit Anfang 2016 externe Unterstützungsleistungen für den Kunden erbrachte. Fokus des Kundenauftrags war die Ergänzung eines unterbesetzten internen Betriebsteams, welches primär zwei Anwendungen „A“ und „B“ betreut. Im Rahmen der hier beschriebenen Projektstätigkeit wurde das Aufgabengebiet „Monitoring des Betriebs“ näher untersucht und im Sinne des „Lean Six Sigma“-Ansatzes verbessert.

Im Themenfeld „Monitoring“ herrschten zu Projektbeginn manuelle Tätigkeiten zur Prüfung des Systemzustands vor. Das Umfeld war stark geprägt von historisch gewachsenen Denk- und Handlungsstrukturen und einer ausgeprägten Intransparenz.

Das folgende Schaubild (welches im Rahmen der Unterstützung des Kundenteams durch Sopra Steria Consulting erstellt wurde, siehe dazu [4]) gibt beispielhaft einen Überblick über die Systemlandschaft „B“:

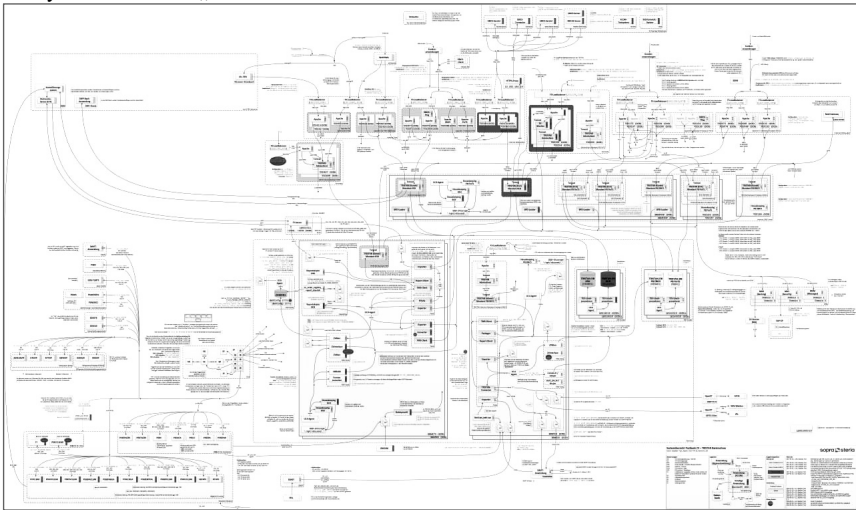


Abbildung 1: Systemlandschaft „B“

Als erschwerend für den inhaltlichen Projekterfolg sei vorausgeschickt, dass die Mitarbeiter des externen Dienstleisters naturgemäß keine Weisungsbefugnis gegenüber kundeninternen Mitarbeitern hatten. Durch gewisse negative Grundhaltungen von Zuständigen sind daher einige Verfahrensschritte des DMAIC-Zyklus (z.B. die Erhebung der Messdaten) nicht optimal durchführbar gewesen. Dieser Umstand führt zu einigen suboptimal erscheinenden Aspekten der Durchführung der einzelnen Phasen bzw. der Anwendung der Werkzeuge.

Im Übrigen ist die Darstellung an einigen Stellen stark gekürzt und auf die (größere) Systemlandschaft „B“ reduziert, um den hier gegebenen Rahmen nicht zu sprengen.

## 2 Define-Phase

Ziel der Define-Phase ist es ein gemeinsames Verständnis über das Problem und das Projektziel zu erarbeiten, den Projektumfang zu definieren, die Kundenstimme (VOC) aufzunehmen und die kritischen Anforderungen (CTQ) des Kunden zu identifizieren.

In der Define-Phase kamen folgende Werkzeuge zum Einsatz: 7W, Problem Statement, Stakeholder Map, VOC, CTQ.

### 2.1 7W

Um die Ausgangslage detailliert zu erfassen, wurde zunächst mit Hilfe der 7W hinterfragt, in welchem Kontext sich das Projekt bewegt. Konkret wurden - mit Fokus auf das Projektthema „Monitoring“ folgende Aspekte betrachtet: Was ist zu tun? Wer macht es? Warum macht es es? Wie wird es gemacht? Wann wird es gemacht? Wo soll es getan werden? Wieso wird es nicht anders gemacht?

Bemerkenswerte Erkenntnisse daraus sind:

- Täglich erfolgt eine Prüfung der Anwendung „A“ anhand einer Checkliste mit ca. 20 Prüfpunkten. Täglich 6x wird eine über 40 Punkte umfassende Checkliste abgearbeitet, um die korrekte Arbeit der Anwendung „B“ zu überprüfen.
- Auf viele Störungssituationen, die in der Vergangenheit vorgekommen sind, wurde mit noch mehr manuellen Prüf-, Sicherheits- und Zwischenschritten reagiert.
- Ansätze zur Automatisierung (z. B. Monitoring) waren da, sind aber nicht konsequent verfolgt und bei auftretenden Problemen als „nicht durchführbar“ deklariert worden.

## 2.2 Problem Statement

Das Problem Statement genügt den SMART-Kriterien und wurde wie folgt formuliert:

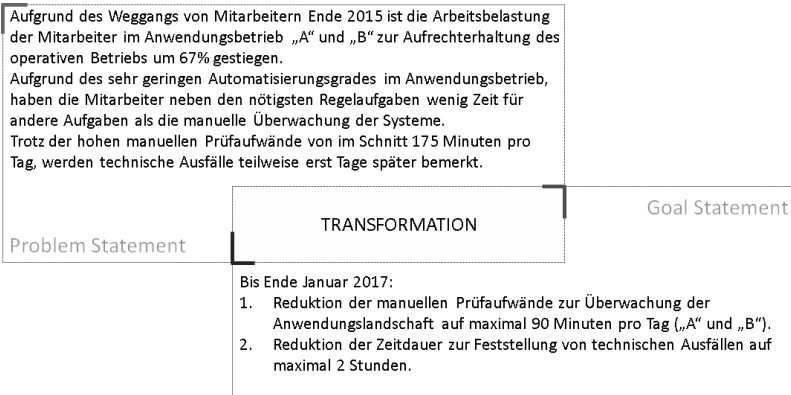


Abbildung 2: Problem- und Goal-Statement

## 2.3 Stakeholder Map

Der Stakeholder Map musste in diesem Projekt eine große Bedeutung beigemessen werden, da hiermit die Standpunkte und Befindlichkeiten der einzelnen Mitarbeiter transparent gemacht werden. Dies ließ einen individuellen Umgang mit jedem Einzelnen zu. Wie die Graphik zeigt, waren die Standpunkte der direkten Stakeholder (kundenseitig) und ihrer Führungskraft durchaus kontrovers:

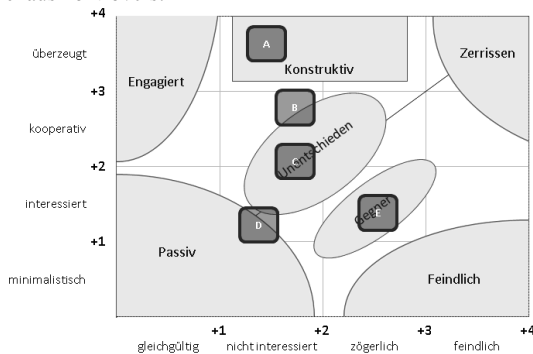


Abbildung 3: Stakeholder Map zu Beginn des Projekts

Im Verlauf des Projekts hat sich die Stakeholder Map geändert und es konnte nach Abschluss des Projekts eine deutlich positivere Einstellung der Beteiligten zum Thema „Monitoring“ festgestellt werden.

## 2.4 VOC

Als Kunde wurde in diesem Projekt der Auftraggeber der Unterstützungsleistung betrachtet. Folgende Monitoring-spezifischen Kundenbedürfnisse („specific needs“) sind in mehreren Gesprächen mit dem Auftraggeber identifiziert worden:

| VOC                  | Kernaussage / Specific need statement  |
|----------------------|--|
| Es muss laufen.      | Operativer Betrieb soll reibungslos funktionieren.<br>Das Betriebsteam soll die Anwendungslandschaft unter Kontrolle haben.<br>Der Betrieb soll bei Problemen schnell und korrekt (zielgerichtet) reagieren. |
| Es darf Geld kosten. | Der Kunde ist sich bewusst, dass der Status quo weit weg vom Idealzustand ist.   |

Tabelle 1: VOC

## 2.5 CTQ

Aus den spezifischen Kundenbedürfnissen lassen sich folgende Hauptfaktoren ableiten, die kritisch für die Erfüllung der Kundenbedürfnisse („CTQ“) im Themenfeld „Monitoring“ sind:

| CTQ-ID | Bezeichnung                             | Beschreibung  |
|--------|---|---|
| C01    | Ausfall-Melddauer                       | Dauer von Ausfall einer Komponente bis zum Bemerkten des Ausfalls darf maximal 2 Stunden betragen.  |
| C02    | Verzögerungsalarmierung                 | Wenn zu verarbeitende Dateien länger als 30 Minuten (ggf. unterschiedlich je Verarbeitungsstrang/Datentyp) liegenbleiben, muss eine Alarmierung erfolgen.   |
| C03    | Alarmierungsqualität                    | Die Alarmierung muss so konkret sein, dass aus der Alarmierungsmeldung hervorgeht:<br>- zu welcher Kategorie die Alarmierung gehört (Ausfall, Fehler bei der Verarbeitung, Verarbeitung stockt)<br>- welche technische Komponente oder welcher Verarbeitungsstrang den Alarm ausgelöst hat. |
| C04    | Aufwand der manuellen Systemüberwachung | Hierunter fallen die Aufwände, die die regelmäßigen Statusprüfungen der Systemlandschaften mit sich bringen.  |

Tabelle 2: CTQ

Die weitergehende Konkretisierung der CTQ und die Ableitung von definierten Kennzahldefinitionen finden sich im Kapitel „Datensammelplan“.

## 3 Measure-Phase

Die Measure-Phase ist gekennzeichnet durch die Ist-Aufnahme der vorgefundenen Prozesse, der Definition zur Sammlung von Daten und der Sammlung der Messdaten. Hiermit kann eine Baseline definiert werden, anhand derer zukünftige Prozessverbesserungen messbar sind. In der Measure-Phase wurden in diesem Projekt folgende Werkzeuge des L6S-Werkzeugkastens genutzt: Ist-Aufnahme des Prozesses, Datensammelplan, Datensammlung.

### 3.1 Ist-Aufnahme des Prozesses

Die Ist-Aufnahme des Überwachungsprozesses der Anwendungslandschaften „A“ und „B“ erfolgte im Rahmen der Einarbeitung in die operativen Abläufe des Anwendungsbetriebs-teams. Die Überwachung ließ sich in die folgenden Themenbereiche gliedern:

- Benachrichtigung per E-Mail aus der Jobsteuerung
- Benachrichtigung per E-Mail aus der Kernkomponente „B“ selbst
- Manuelle Prüfungen

Betrachtet wurden dabei im Detail die spezifischen Eigenarten der monitoringspezifischen Abläufe (Jobsteuerung, E-Mail-Benachrichtigungen, manuelle Prüfungen mit nur geringer Skriptunterstützung): Wer prüft wann (konkret!) was (konkret!)? Die manuellen Prüfungen unterlagen in vielen Punkten keiner definierten Struktur; vielmehr oblag die Beurteilung, ob eine Fehlersituation vorliegt oder nicht, dem jeweiligen Prüfer nach individueller Einschätzung.

### 3.2 Datensammelplan

An den Kennzahlen muss der Projekterfolg im Verlauf messbar werden. Die Kennzahlen wurden direkt aus den CTQ abgeleitet (vgl. Kapitel „CTQ“).

#### 3.2.1 Ausfall-Meldedauer (C01)

Gemessen wird beim Bemerkten einer Störung (durch Alarmierung oder manuelle Prüfung) die Differenz ( $t_1 - t_0$  in Minuten) zwischen den Zeitpunkten

- zu dem die Störung tatsächlich aufgetreten ist ( $t_0$ ) und
- zu dem die Störung bemerkt wurde ( $t_1$ ).

#### 3.2.2 Verzögerungsalarmierung (C02)

Gemessen wird beim Bemerkten einer Verarbeitungsverzögerung (durch Alarmierung oder manuelle Prüfung) die Differenz ( $t_1 - t_0$  in Minuten) zwischen den Zeitpunkten

- zu dem die Verzögerung tatsächlich aufgetreten ist ( $t_0$ ) und
- zu dem die Verzögerung bemerkt wurde ( $t_1$ ).

#### Bewertung

Das tatsächliche Auftreten einer Verzögerung bemaß sich am Überschreiten der für den jeweiligen Verarbeitungsstrang definierten maximalen Dauer des Verweilens von eingehenden Daten ins System vor der eigentlichen Verarbeitung. Dafür konnten allerdings keine konkreten Werte sondern nur vage Anhaltspunkte definiert werden. Die Aussagekraft dieser Messwerte musste also als ungenügend eingeschätzt werden. Die Kennzahl wurde trotzdem beibehalten, um ggf. zu einem späteren Zeitpunkt eine Verfeinerung des Messsystems zu dieser Kennzahl vorzunehmen.

#### 3.2.3 Alarmierungsqualität (C03)

Die Messung zur Alarmierungsqualität erfolgt bei jeder Alarmierung. Dabei ist zu bewerten, ob auf Basis der Alarmierung eine zielgerichtete Reaktion möglich war (Messwert = 0) oder nicht. Wenn nicht, wird gemessen wie lange es dauerte, bis die Störungsanalyse „auf der richtigen Spur“ war (in Minuten).

#### 3.2.4 Aufwand der manuellen Systemüberwachung (C04)

Gemessen wird

- bei jeder Durchführung einer manuellen Prüfung durch den Prüfer selbst

- der zeitliche Aufwand (in Minuten) von Start bis Abschluss der Abarbeitung der Prüfcheckliste.

Die bestehenden (kundeninternen) Teamkollegen hatten sich leider nicht dazu bereit erklärt die o. g. Messwerte zu erheben. Sie wurden also nur für die beiden externen Kollegen gemessen. Die Erfassung der Messwerte erfolgte in der Prüfcheckliste selbst.

### 3.3 Daten

Für die einzelnen Kennzahlen wurden folgende Daten erhoben.

#### 3.3.1 Ausfall-Meldedauer (C01)

Neben den praktischen Messungen war hier zusätzlich eine theoretische Betrachtung sinnvoll, da zwischen der Messrealität und den Betriebsabläufen eine erhebliche Diskrepanz bestand.

##### Theoretische Betrachtung

Wenn eine Komponente ausfiel, zog dies nicht immer unmittelbar sichtbare Auswirkungen nach sich, da die Systeme redundant ausgelegt waren. Die theoretische maximale Dauer, bis eine Störung bemerkt wurde, bemaß sich daher aus der Differenz zwischen

- dem Zeitpunkt des Auftretens der Störung direkt nach einem Prüflauf und
- dem Zeitpunkt des Bemerkens der Störung im darauffolgenden Prüflauf.

Im Zeitraum jeweils von Montag bis Freitag ergab sich somit eine Dauer von bis zu 16 Stunden, bis ein Ausfall bemerkt wurde (es sei denn, die Kunden oder abhängige Systeme bemerken die Nichtfunktionalität und alarmieren das Betriebsteam). Noch kritischer war die Betrachtung der Wochenenden: Zwischen der letzten Prüfung am Freitag bis zur ersten montäglichen Prüfung vergehen mehr als 60 Stunden!

##### Reale Messwerte

In der Realität wurden weitaus größere Zeitspannen gemessen, bis eine Störung erkannt wurde. In einem Fall vergingen 117 Stunden, währenddessen trotz Kundenbeschwerden (manuelle Systemprüfung ergab keine konkrete Auffälligkeit) der Ausfall von zwei der drei Server nicht bemerkt worden ist.

#### 3.3.2 Verzögerungsalarmierung (C02)

Für diese Kennzahl gilt die gleiche theoretische Betrachtung wie für die Ausfall-Meldedauer.

##### Reale Messwerte

Im Projektverlauf gab es keine Vorfälle, bei denen eine Verzögerung tatsächlich zu einem Problem geführt hat. Annahme ist, dass Daten häufiger länger unbearbeitet blieben als sie sollten - es bemerkte aber niemand.

#### 3.3.3 Alarmierungsqualität (C03)

Es gab Alarmierungen per E-Mail bei bestimmten Statusänderungen von Aufträgen aus dem System „B“. In der Alarmierung wurde der betroffene Auftrag genannt und somit ein direkter Einstieg in die Störungsanalyse ermöglicht. Oft waren diese Alarmierungen nicht wirklich relevant, da sie eher einen Hinweis darstellten, als auf eine echte Störung hinzuweisen.

Alarmierungen aus der Jobsteuerung bei Jobabbrüchen und anderen Auffälligkeiten waren wenig sprechend, gaben aber Hinweise auf den abgebrochenen Job und ermöglichten somit einen direkten Einstieg in die Störungsanalyse. Weitere Alarmierungen waren nicht vorhanden. Bei den bestehenden Alarmierungen war also sofort klar, woher sie stammen und was der Auslöser war. Daher ergaben die Messwerte zu dieser Kennzahl immer „0“ - die Messung war somit aussageelos.

### 3.3.4 Aufwand der manuellen Systemüberwachung (C04)

Bei den täglichen Prüfungen wurde in der jeweiligen Checkliste die Dauer der Prüfung in Minuten erfasst, wenn die Prüfung durch einen der externen Kollegen durchgeführt wurde (vgl. Kapitel 3.2.4).

## 4 Analyse-Phase

Die Analysephase dient dazu, den Unterschied zwischen der Realität und den theoretisch gewünschten Qualitätsmerkmalen der Abläufe herauszuarbeiten und die Grundursachen für die Abweichungen zu identifizieren. Im Rahmen dieses Projekts kamen folgende Werkzeuge zum Einsatz: Ursache-Wirkungs-Diagramm, Time Series Plot.

### 4.1 Ursache-Wirkungs-Diagramm

Mit Hilfe eines Ishikawa-Diagramms (mit den klassischen „Hauptgräten“ Methode, Maschine, Material, Mensch, Messung und Mitwelt), wurde mit einem gesamtheitlichen Fokus auf das Anwendungsbetriebsteam die Problemstellung analysiert, warum die Mitarbeiter des Anwendungsbetriebs überlastet sind. Mit Fokus auf das Themengebiet „Monitoring“ ergaben sich die Handlungsfelder:

- Methode: „Kaum Automatisierung“
- Messung: „Keine Messung“ sowie „Kein Bedarf an Messungen“
- Mitwelt: „Management-Attention wird gefürchtet“

In Einzel-Interviews mit den Mitarbeitern und ihrer Führungskraft konnten diese Ursachen und Handlungsfelder verifiziert werden.

Die quantitative Betrachtung der Ursache „Kaum Automatisierung“ bestätigte, dass hier ein echtes Problem vorlag, da täglich fast 3 Stunden (175 Minuten) und somit (bezogen auf die drei internen Mitarbeiter) fast 12,5% der verfügbaren Arbeitszeit für manuelle Prüfungen aufgebracht wurden.

### 4.2 Time Series Plot

Die folgende Graphik zeigt den Verlauf der manuellen Prüfaufwände für das System „B“.

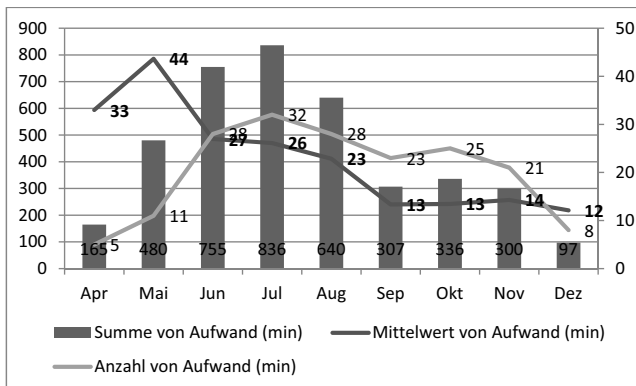


Abbildung 4: Manuelle Prüfaufwände System „B“

Dargestellt ist die monatliche Summe der Aufwände in Minuten; die Kurve „Anzahl von Aufwand“ gibt die Anzahl der Messungen je Monat wieder (was als Indikator für die Aussagekraft der Messung anzusehen ist); die Kurve „Mittelwert von Aufwand“ zeigt den durchschnittlichen Aufwand pro Prüfung.

Als Baseline für den täglichen Aufwand der manuellen Systemprüfung wurde der Wert von 26 Minuten pro Prüfvorgang festgelegt; dies entspricht 156 Minuten pro Tag. Zusammen mit den Aufwänden der manuellen Prüfungen des Systems „A“ (19 Minuten pro Tag) ergeben sich die im Problem Statement aufgezeigten 175 Minuten pro Tag.

Die Graphik zeigt, dass sich der manuelle Prüfaufwand mit Einführung der Verbesserungsmaßnahmen (Stufe 1, voll wirksam ab September) deutlich reduzierte.

## **5 Improve-Phase**

Charakteristisch für die Improve-Phase sind die Bemühungen Lösungen zu generieren und geeignete Lösungen auszuwählen, die die aufgezeigten Probleme bestmöglich beseitigen; ebenso ist die Umsetzung der definierten Maßnahmen Bestandteil dieser Phase.

Aus dem Lean Six Sigma-Werkzeugkasten wurden in diesem Projekt genutzt: Kreativitätstechniken, Priorisierungsmatrix, Business Case, Implementierungsplan, Implementierungsplan umsetzen.

### **5.1 Kreativitätstechniken**

Es wurden verschiedene Kreativitätstechniken angewandt, um Ideen für mögliche Verbesserungsmaßnahmen zu identifizieren.

Im Nachgang wurden Kriterien an die generierten Ideen angelegt, um sie zu bewerten, wobei das Kundenumfeld (z. B. organisatorische Fragestellungen) berücksichtigt wurde:

- Realistisch: Kann die Maßnahme (z. B. im Rahmen der Kundenorganisation und -ausstattung) umgesetzt werden?
- Akzeptabel: Würde der Kunde / das Kundenteam eine solche Maßnahme akzeptieren?

### **5.2 Priorisierungsmatrix**

Die gefundenen Ansätze zu Verbesserungsmaßnahmen wurden nun einer detaillierten Bewertung zugeführt und mit Hilfe einer Priorisierungsmatrix priorisiert. Dabei wurden die Kriterien mit einer Gewichtung versehen und die einzelnen Vorschläge anhand der Kriterien bewertet.

Die Auswahl der individuell auf das Projekt zugeschnittenen Kriterien und ihrer Gewichtung erfolgte so, dass die Umsetzbarkeit und Wirksamkeit der Maßnahmen im Umfeld der speziellen Kundensituation eine relevante Bewertungsgröße wurde. Besonderes Gewicht wurde daher auf das Kriterium „Akzeptanzwahrscheinlichkeit“ gelegt, da die zu bewertenden Maßnahmen mehr oder weniger ausschließlich Kraft ihrer eigenen Wirkung auf das Kundenteam langfristig überzeugen konnten (vgl. Kapitel 1.2).



|  | Kriterien |                 |                             |                | Summen     |           |
|--|-----------|-----------------|-----------------------------|----------------|------------|-----------|
|  | Aufwand   | Erfolgsaussicht | Akzeptanzwahrscheinlichkeit | Nachhaltigkeit |            |           |
| Gewichtung   | 1         | 4               | 5                           | 3              | Ungeachtet | Gewichtet |
| <b>Maßnahme</b>  |           |                 |                             |                |            |           |
| Team trainieren und schneller prüfen   | 2         | 4               | 1                           | 1              | 8          | 26        |
| Team trainieren und genauer prüfen   | 2         | 3               | 1                           | 2              | 8          | 25        |
| Ablenkung minimieren   | 1         | 4               | 2                           | 1              | 8          | 30        |
| Systemüberwachung automatisieren   | 5         | 5               | 3                           | 5              | 18         | 55        |
| Prüfungstätigkeiten auslagern  | 3         | 3               | 2                           | 2              | 10         | 31        |
| Häufiger prüfen und Probleme schneller erkennen                                    | 1         | 1               | 2                           | 1              | 5          | 18        |
| Prüfpunkte auf besonders wichtige Punkte reduzieren, um schneller prüfen zu können | 2         | 2               | 4                           | 1              | 9          | 33        |

Tabelle 3: Priorisierungsmatrix der Verbesserungsvorschläge

Gemäß gewichteter Bewertung stellte sich der Punkt „Systemüberwachung automatisieren“ ein herausragender Ansatz dar, der damit weiter verfolgt wurde.

### 5.3 Business Case

Der Business Case zum Thema „Systemüberwachung automatisieren“ zeigte auf, inwieweit sich die Umsetzung rein rechnerisch lohnt. Dabei wurden schlussendlich die detaillierten Investitionskosten den konkreten Einsparungen gegenübergestellt und somit eine Bewertung ermöglicht. In die Berechnung flossen verschiedene Detaildaten

- zu den Umsetzungskosten die investiert werden mussten, um die Effekte zu erzielen, die letztendlich zu den Einsparungen führten und
- zu den eingesparten Zeitaufwänden (gem. Goal-Statement) bei der manuellen Systemprüfung in Verbindung mit einem Personalkostensatz (auf Basis des beispielhaften Tagessatzes eines externen Dienstleisters).

Das Endergebnis der Berechnungen stellte sich wie folgt dar:

| Einheit       | Investition | Einsparung |
|---------------|-------------|------------|
| Minuten / Tag | -           | 85 / Tag   |
| PT / Jahr     | 27          | > 46       |
| EUR           | 23.700      | 40.672     |

Tabelle 4: Überblick Business Case

Der Investition von 27 PT stand die Einsparung von >46 PT innerhalb eines Jahres gegenüber. Der ROI wurde also schon nach gut sieben Monaten erreicht. Rückblickend zeigte sich, dass die Einsparung tatsächlich noch um gut 10% höher ausfiel.

Zusätzlich kommen zu den aufgezeigten Einsparungen die Qualitätsgewinne hinzu, die sich aus einer automatisierten Systemüberwachung ergeben und hier nicht berechnet worden sind.

## **5.4 Implementierungsplan**

Die Implementierungsplanung beschreibt, welche Schritte zur Umsetzung der geplanten Maßnahmen zur Prozessverbesserung wann und durch wen umgesetzt werden sollen. Der Implementierungsplan sah für beide Systeme nahezu identisch aus. Hier die grobe schematische Planung:

1. Nutzung von Standardtools
2. Klärung von Schnittstellen und unterstützenden Einheiten
3. Nachhaltigen Prozess definieren
4. Anforderungen dokumentieren und abstimmen
5. Überwachungen implementieren und Visualisierung aufbauen
6. Überwachung verifizieren

Ergebnis: Teilautomatisierte Überwachung der Systemlandschaft

## **5.5 Implementierungsplan umsetzen**

Für das System „A“ wurden die einzelnen Überwachungen sukzessive eingerichtet. Somit wurden die Effekte nach und nach spürbar, was sich in den manuellen Prüfaufwänden widerspiegelte. Bei der Realisierung der Überwachungen in der Systemlandschaft „B“ waren erste positive Effekte bereits während der Testphase bemerkbar; seit Abschluss der Tests zu Anfang September 2016 ist der volle Effekt sichtbar (vgl. Datengraphik in Kapitel „Time Series Plot“).

# **6 Control-Phase**

Die Control-Phase beinhaltet die Überwachung der verbesserten Prozesse. Hierzu wurden in diesem Projekt folgende Werkzeuge genutzt: Messungen, visuelles Management, Dashboards.

## **6.1 Messungen**

Der Erfolg der umgesetzten Maßnahmen ließ sich insbesondere an den manuellen Aufwänden der Systemüberwachung erkennen. Siehe hierzu das Kapitel „Time Series Plot“, wo auch die Daten der Messreihen dargestellt sind, die über den Zeitpunkt der Umsetzung der Verbesserungsmaßnahmen hinausgehen.

## **6.2 Visuelles Management**

Für die Systemlandschaft „B“ gab es zwei Aspekte, bei denen mit Hilfe von visuellem Management Sachverhalte „präsentiert“ wurden, dass wesentliche Informationen auf einen Blick erkennbar wurden.

### **6.2.1 Eventmanagement**

Die automatische Überwachung von Sachverhalten zur Beurteilung des Zustands der Systemlandschaft erzeugte bei Auffälligkeiten Events (sogen. „eventgetriebene Überwachung“). Diese wurden im Standardwerkzeug so dargestellt, dass den verschiedenen Kritikalitäten eine

definierte Farbgebung zugeordnet war, sodass direkt erkennbar war, ob schwerwiegende Störungen oder unkritische Warnmeldungen vorlagen.

### 6.2.2 Prüfnachweis

Zum Nachweis der aktiven Prüfung der Systemlandschaft wurde durch das Anwendungsbetriebsteam eine Prüfnachweisliste geführt. Hier wurde nachgehalten, dass täglich durch eines der Teammitglieder der Systemstatus anhand der Überwachungswerkzeuge geprüft wurde und ggf. welche Auffälligkeiten aufgetreten waren. In der Prüfnachweisliste wurde zu den einzelnen Überwachungen für alle Komponenten der Systemlandschaft dokumentiert, ob Monitoring-Events vorlagen. Somit ist transparent nachvollziehbar, dass die Events - und ggf. daraus hervorgehende Störungssituationen - registriert und bearbeitet wurden.

| Datum      | Uhrzeit | Aufwand (min) | Status | Tickets | Eingang | Ausgang | Spoolverz. | Auslastung | Srv-M-Proz. | Reboot? | TTrace | Auslastung | Srv-M-Proz. | Reboot? | Sharesgriff | Auslastung | Srv-M-Proz. | Reboot? | Status | Fehl |   |
|------------|---------|---------------|--------|---------|---------|---------|------------|------------|-------------|---------|--------|------------|-------------|---------|-------------|------------|-------------|---------|--------|------|---|
| 14.10.2016 | 7:32    | 2             | OK     | ok      | M       | M       | M          | M          | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 17.10.2016 | 7:49    | 2             | OK     | ok      | M       | M       | M          | M          | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 18.10.2016 | 8:18    | 4             | Fehler | ok      | M       | M       | M          | M          | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 20.10.2016 | 7:18    | 1             | OK     | ok      | M       | M       | M          | M          | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 21.10.2016 | 8:36    | 3             | Fehler | ok      | M       | M       | (ok)       | (ok)       | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 24.10.2016 | 7:15    | 5             | Fehler | M       | M       | M       | (ok)       | (ok)       | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 25.10.2016 | 7:23    | 3             | OK     | ok      | M       | M       | (ok)       | (ok)       | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |
| 26.10.2016 | 7:35    | 2             | OK     | (ok)    | M       | M       | (ok)       | (ok)       | M           | M       | M      | M          | M           | M       | M           | M          | M           | M       | M      | M    | M |

Abbildung 5: Visuelles Management beim Prüfnachweis

Auch hier wird durch farbliche Hervorhebungen der einzelnen Prüfstatus gekennzeichnet, ob eine Auffälligkeit anliegt, wodurch sich über den Zeitverlauf eine Art „Heatmap“ ergibt:

### 7.3 Dashboards

Für die Systemlandschaft „B“ wurden für die überwachten (in der ersten Ausbaustufe vorwiegend fachlichen) Aspekte Dashboards entwickelt, in denen die Daten der Überwachungstätigkeiten tageweise fortlaufend dargestellt wurden.

Da die Systemlandschaft aus vier Mandanten besteht, wurden die einzelnen Datenreihen den Mandanten zugeordnet.

|    |   |                                   |
|----|---|-----------------------------------|
| M1 | ! | ! > M1 - Kanäle<br>✓ > M1 - Daten |
| M2 | ✗ | ✗ > M2 - Kanäle<br>✓ > M2 - Daten |
| M3 | ✓ | ✓ > M3 - IDs<br>✓ > M3 - Aufträge |
| M4 | ✓ | ✓ > M4 - Traffic                  |

Abbildung 6: Hauptampel der Systemlandschaft „B“ je Mandant

Die Zusammenfassung der Messreihen je Mandant und der jeweilige Status der einzelnen Metriken ergaben somit eine „Ampel“ je Mandant. Hieran war in Sekundenschnelle der Gesamtstatus der Überwachung ablesbar.

Aus der Hauptampel konnte in die detaillierte Darstellung der Metriken und deren Status je Mandant und von dort weiter auf die Visualisierung der einzelnen Datenreihen gezoomt werden:

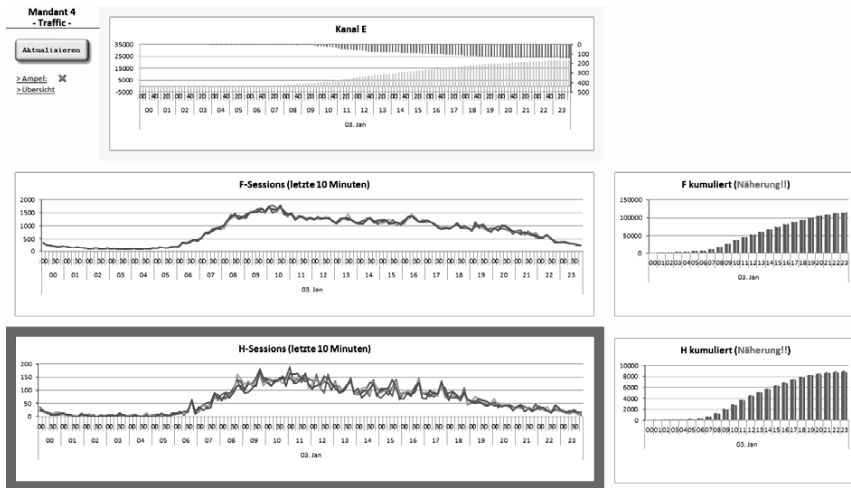


Abbildung 7: Dashboard-Seite eines Mandanten der Systemlandschaft „B“

Somit konnte das Anwendungsbetriebsteam jederzeit den Status der einzelnen Metriken ablesen, ohne aufwendige manuelle Prüfungen in oder an den Systemen selbst vornehmen zu müssen. Bei durch die Überwachung erkannten Störungen werden die Metriken (vgl. o. s. Abbildung) gelb bzw. rot umrandet, um die Status „Warnung“ oder „Alarm“ zu symbolisieren.

## 7 Reinforce

Die Phase Reinforce beinhaltet die Aspekte der langfristigen Etablierung der verbesserten Prozesse, des Umgangs damit („Wird der neue Prozess gelebt?“) und die langfristige Wirksamkeit der Veränderungen. Weiterhin werden die Themen „Lessons learned“ und „Marketing“ thematisiert.

### 7.1 Etablierung der Prozesse

Die Einführung der automatischen Monitoringlösung im Betriebsteam war kein Selbstläufer. Zu viele historisch bedingte Vorbehalte und negative Grundeinstellungen standen einer reibungslosen Einführung im Wege. Daher wurde seitens des Projektteams die Methode der „schleichenden Gewöhnung“ angewandt. Nachdem jeder Einzelwert und jede Messtoleranz mit allen Beteiligten abgestimmt worden und die Umsetzung abgeschlossen war, wurde ausgiebig zu den neuen Möglichkeiten nebst ihren Vorteilen und Anwendungshinweisen kommuniziert und geschult. Danach wurden von den externen Kollegen im Betriebsteam die neuen Werkzeuge rigide angewandt und zu jedem möglichen Zeitpunkt die erfolgreiche Anwendung kommuniziert (z. B. wenn Störungen oder Unregelmäßigkeiten erkannt wurden oder im Rahmen von Störungen Ursachenforschung zu betreiben war oder unterstützt werden konnte). Dadurch ergaben sich nach und nach eine Annäherung des Kundenteams und ein sukzessiver Vertrauensaufbau in die neuen Werkzeuge.

Im Fall der Systemlandschaft „A“ wurden die täglichen Systemprüfungen regelmäßig nur noch von den externen Kollegen durchgeführt (die ausschließlich das Monitoring nutzen). Ab ca. Januar 2017 war es endlich soweit, dass auch durch die kundeninternen Kollegen die

Überwachung der Systemlandschaft „A“ ausschließlich auf Basis der automatischen Monitoringfunktionen erfolgte. Manuelle Prüfungen fanden seitdem nicht mehr statt.

Einige Wochen nach der Veröffentlichung und Schulung der Dashboards zur Systemlandschaft „B“ war das Kundenteam aktiver (und begeisterter) Nutzer der Dashboards. Die manuellen Prüfungen wurden hingegen trotzdem weiterhin parallel durchgeführt (nur vom internen Kundenteam). Bzgl. der Systemlandschaft „B“ zeichnete sich Stand September 2017 ab, dass die manuelle Doppel-Prüfung der im Monitoring eingebundenen Aspekte durch das interne Kundenteam reduziert werden könnten und sich somit die volle Wirkung des im Business Case (vgl. Kapitel „Business Case“) dargestellten Automatisierungspotentials entfalten würde.

## **7.2 Lessons learned**

Für folgende Projekte kann mitgegeben werden, dass dem Change Management bzgl. der betroffenen Mitarbeiter noch mehr Aufmerksamkeit zugewandt werden muss. Die Bedarfe, Ängste und Befindlichkeiten müssen noch intensiver erforscht werden, um damit optimal umzugehen. Im hier dokumentierten Projekt gab es Eskalationen, in denen die Kundenmitarbeiter klar kommuniziert haben, dass die externen Kollegen sie mit ihrem Vorgehen überfordern. Der Konflikt konnte nur in einem gemeinsamen Gespräch mit Team und Auftraggeber (dem es eigentlich noch zu langsam ging) mit dem Konsens, dass das Kundenteam noch intensiver abgeholt und auf die neuen Abläufe geschult werden wird, gelöst werden.

Ergänzend hierzu sollte in Folgeprojekten die Managementebene der jeweils von Veränderungen betroffenen Kollegen stärker eingebunden werden. Vorteil dabei wäre, dass eine direktere Einflusnahme und Weisung möglich ist, die aus dem Projektteam (als unterstützende externe Kräfte) so nicht möglich ist. Somit mussten in diesem Projekt alle Änderungen über einen unnötig langen Zeitraum vorgelebt werden, um Akzeptanz zu finden und die althergebrachten Vorgehens- und Denkweisen („Das machen wir schon immer so und es funktioniert gut!“) zu ändern.

Als hilfreich hat sich letztendendes erwiesen, dass unsere Arbeit auch außerhalb des Betriebsteams und vor allem auf der Managementebene im IT-Betrieb großen Anklang fand (vgl. Kapitel „Marketing“). Dadurch stieg auch im vom Projekt „betroffenen“ Anwendungsbetriebsteam die Akzeptanz unserer Arbeit (i. S. v. „Wenn die das alle gut finden, muss es ja gut sein.“).

Zusammenfassend kann man sagen, dass die kundeninternen Kollegen ganz klassisch die Kübler-Ross-Kurve [5] durchlaufen haben und somit gemäß den Erwartungen am Ende „alles gut“ wurde. Als Projektteam wäre es wünschenswert gewesen, wenn die Zeitachse nicht ganz so langgestreckt gewesen wäre.

## **7.3 Marketing**

Die Marketingaktivitäten fanden bereits ab Juli 2016 vor allem auf der Managementebene im IT-Betriebsumfeld statt. Es wurde im Kollegenkreis vorhandene Kontakt genutzt, um die verschiedenen Highlights unserer Arbeit vorzustellen. Durch Multiplikationseffekte wurde das Sopra Steria-Team zu verschiedenen Terminen eingeladen, um unsere Arbeit zu präsentieren, worauf immer ein positives Feedback folgte. Neben den Themen „Systemübersicht“, „Knowledge Management / Wiki“ und der Strukturierung und Organisation der allgemeinen Arbeitsweise (vgl. auch [4]), fand im Januar 2017 auch zum Themenfeld „Monitoring“ ein Termin mit Vertretern eines laufenden Betriebsautomationsprojekts statt. Die Resonanz war so positiv, dass wir zu Folgeterminen geladen, die Vorgehensweise „Monitoring System B“ zum Prototypen erklärt wurde und schlussendlich eine Folgebeauftragung entstanden ist, die die Entwicklung eines übergreifenden Monitoringkonzepts für den gesamten IT-Betrieb zum Gegenstand hatte.

## 8 Fazit und Ausblick

Abschließend ist festzuhalten, dass das Projekt an sich ein voller Erfolg war und ist, die umgesetzten Maßnahmen eine nachhaltige Wirkung zeigen und das nutznießende Kundenteam die aufgebauten Monitoring-Werkzeuge und -prozesse gern und aktiv nutzt.

Erfolgsfaktoren in diesem Projekt waren

1. die Hartnäckigkeit des Projektteams, welches dafür sorgte, dass die Verbesserungsmaßnahmen immer und immer wieder thematisiert und gegen die Widerstände der kundeninternen Kollegen behauptet und vorangetrieben wurden sowie
2. die Anwendung der Six Sigma-Methodik. Hiermit konnte transparent und nachvollziehbar dargestellt werden, welche tatsächlichen Probleme in welchem Maße bestehen und wie die durchgeführten Maßnahmen wirkten. Dies führte zu einer starken und nachhaltigen Motivation des Projektteams und unterstützte die Kommunikation mit den Kundenvertretern - vor allem auf der Management-Ebene - enorm.

Der nächste Schritt ist die Umsetzung eines technischen Grundlagenmonitorings auch für die Systemlandschaft „B“, sodass z. B. laufende und aktive Prozesse nicht mehr manuell geprüft werden müssen. Diese Maßnahme wurde (nach den in diesem Bericht beschriebenen Maßnahmen zum erweiterten Monitoring) umgesetzt und entlastet das Anwendungsbetriebsteam in noch größerem Maße. Der Business Case ließe sich also sehr erfolgreich fortschreiben.

## Literatur

- [1] Michael L. George: Lean Six SIGMA for Service, Mcgraw-Hill Education Ltd. (2003)
- [2] Barbara Wheat: Leaning Into Six Sigma, General Finance & Investing (2003)
- [3] Peter S. Panda: The Six SIGMA Way, Mcgraw Hill Book Co (2000)
- [4] Stephan Tost: Erfolgsfaktoren für das Application Management Outsourcing, 41. WI-MAW-Rundbrief, Jahrgang 23 Heft 1, 01. April 2017, S. 25 ff., ISSN 1610-5753
- [5] Dr. Patrick Fritz: „Change Kurve nach Kübler-Ross“, URL abgerufen am 20.01.2018 <http://www.fritz.tips/change-kurve-nach-kuebler-ross/>



**Fünfte gemeinsame Tagung der Fachgruppen  
Vorgehensmodelle (WI-PM) und Projektmanagement (WI-PM)  
im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V.  
in Zusammenarbeit mit der Fachgruppe IT-Projektmanagement der GPM e.V.**



In Zusammenarbeit mit:  
**GPM**  
FG IT-Projektmanagement

**Call for Papers / Aufruf zur Einreichung von Beiträgen**

**PVM 2018**  
Projektmanagement & Vorgehensmodelle

**Der Einfluss der Digitalisierung auf Projektmanagementmethoden  
und Entwicklungsprozesse  
15.+16.10.2018 in Düsseldorf**

Unter dem Leitthema der digitalen Transformation wurden von Wirtschaft und Politik vielfältige Projektinitiativen gestartet. Mit innovativen IT-Lösungen (z. B. Cloud Technologien, Internet of Things, Big Data, Künstliche Intelligenz, Augmented Reality, Robotik, UX-Designpatterns etc.) will man zukunftsorientierte Lebens- und Arbeitssituationen schaffen. Die Digitalisierungsstrategien verbinden hierbei aktuelle technologische Trends mit neuen Geschäftsmodellen, bei deren Umsetzung auch weitere Einflussfaktoren aus Politik und Gesellschaft (z.B. Folgen der neuen EU Datenschutz Grundverordnung) zu beachten sind.

Das Management der Digitalisierungsprojekte erfordert daher einen inter- und transdisziplinären Projektansatz, der Kompetenzen und auch Arbeitsweisen und Modelle verschiedener Disziplinen verbindet und dabei agile Denkmuster als Grundlage für experimentelle Lernprozesse erlaubt.

Die Digitalisierung der Produktionsprozesse stellt auch die Industrie vor neue Herausforderungen. Durch die wachsenden Ansprüche an die Funktionalität steigt die Komplexität der eingesetzten Software stetig. Nicht nur durch die zunehmenden Berichte über industrielle Security-Vorfälle nimmt die Sensibilisierung für Notwendigkeit für Industrial Security zu. Immer mehr Unternehmen erkennen hier den Bedarf ihre Vorgehensmodelle zu optimieren, um weiterhin hochqualitative, sichere Software schnell zu entwickeln.

## **Themen**

Im Mittelpunkt der Fachtagung steht die Frage, welche Projektmodelle, Methoden und Instrumente für die komplexen Digitalisierungsprojekte bereits nutzbar sind und wie diese für inter- bzw. transdisziplinären Projekte zu erweitern sind.

Um dieses vielfältige Thema im Spannungsfeld zwischen Academia und Praxis zu diskutieren, laden wir Beiträge insbesondere zu den folgenden Fragestellungen ein:

- Inter- und transdisziplinäre Projektmodelle für Digitalisierungsprojekte
- Systemische Ansätze für vernetztes Lernen in Projekten
- Innovative Arbeitsumgebungen für Digitalisierungsprojekte (Labs, Coworking)
- Rollenkonzepte und Entscheidungsinstanzen in transformativen Projekten
- Kundenintegrationsmodelle in Projekten (z. B. Open Innovation Ansatz)



- Auswirkung von agiler bzw. hybrider Projektstrukturen auf die Projektsteuerung
- Trends und Prognosen für Vorgehensmodelle und Projektmanagement (Ausblick)
- Security by Design
- Teamzentrierte Vorgehensmodelle (z.B. Agile, Lean Development, DevOps)

Anlässlich des 25-jährigen Jubiläums der Fachgruppe Vorgehensmodelle laden wir außerdem Beiträge zur Geschichte der Vorgehensmodelle ein

*Natürlich können auch andere interessante Themenvorschläge aus den Bereichen Projektmanagement und Vorgehensmodelle eingereicht werden.*

## **Ziele der Fachtagung**

Ziel der Veranstaltung ist es, fundierte Ansätze aus der Wissenschaft mit Erfahrungen zu deren Anwendung in der Praxis einem Fachpublikum vorzustellen und Raum für die fachübergreifende Diskussion und den Erfahrungsaustausch zu geben.

## **Beiträge**

Bei den Einreichungen kann es sich handeln um

- Übersichtsbeiträge, die einen oder mehrere der o.g. Fragestellungen aufarbeiten und einfürend darstellen,
- Ergebnisse laufender oder abgeschlossener Forschungsarbeiten,
- Erfahrungsberichte aus der Praxis,

als Fachbeiträge (max. 10 Seiten) oder Future Track (Impulsbeiträge, max. 5 Seiten).

Die Fachgruppen prämiieren einen herausragenden Beitrag mit dem *Best Paper Award*.

Von der Einwendung von sog. Success Stories mit starker Marketing-Ausrichtung (Eigenwerbung) bitten wir generell abzusehen.

## **Session „Future Track“**

Eine wichtige Aufgabe der GI-Fachgruppen ist es, sich mit der Zukunft des Fachgebiets zu beschäftigen. Autoren im Future Track können reife Ideen oder kontroverse bzw. provokative Ansichten in einem Impulsbeitrag vorstellen, welche anschließend im Auditorium diskutiert werden sollen. Durch den Austausch sollten Denkanstöße und Impulse für die Teilnehmer und auch die künftige Fachgruppenarbeit entstehen. Die Impulsbeiträge (nur als Kurzbeitrag) für die Future Tracks werden einem separaten Review-Verfahren unterzogen.

## **Einreichungen**

- **15.05.2018:** Einreichung durch die Autoren
- **13.07.2018:** Benachrichtigung der Autoren
- **05.08.2018 :** Einreichung finale Version

Reichen Sie ihr Paper über folgenden Submission-Link ein:

<https://easychair.org/conferences/?conf=pvm2018>

Die verbindliche Word-Vorlage und weitere Informationen zur Einreichung finden Sie auf der Tagungs-Website [pvm-tagung.de](http://pvm-tagung.de) (Bitte nicht die Vorlage der GI verwenden!)

## **Tagungsband**

Der Tagungsband wird in gedruckter Form in den *GI Lecture Notes in Informatics* publiziert (<https://www.gi.de/service/publikationen/lni.html>) und von dblp (<http://dblp.uni-trier.de/>) indiziert.

## **Tagungsort**

Die Tagung findet am 15.+16.10.2018 in Düsseldorf statt:

FOM Hochschule für Oekonomie & Management

Toulouser Allee 53

40476 Düsseldorf

## **Kontakt**

Auf der Webseite <http://pvm-tagung.de> werden laufend aktualisierte Informationen zur Tagung bereitgestellt. Für Rückfragen wenden Sie sich bitte an [info@pvm-tagung.de](mailto:info@pvm-tagung.de).

## **Für das Programmkomitee der Tagung:**

Prof. Dr. Martin Engstler (Sprecher der Fachgruppe Projektmanagement)

Alexander Volland (Stv. Sprecher der Fachgruppe Projektmanagement)

Prof. Dr. Eckhart Hanser (Sprecher der Fachgruppe Vorgehensmodelle)

Dr. Masud Fazal-Baqaie (Stv. Sprecher der Fachgruppe Vorgehensmodelle)

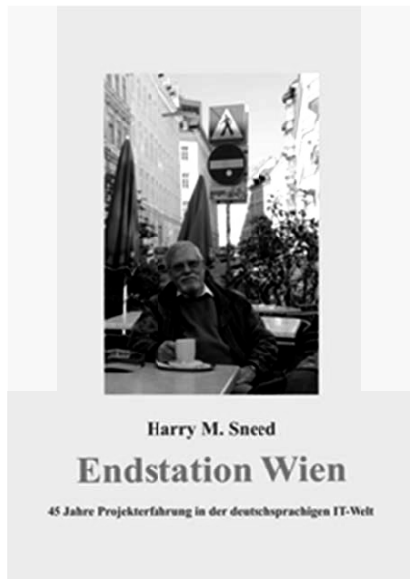
Prof. Dr. Oliver Linssen (Sprecher der Fachgruppe IT-Projektmanagement der GPM)

# **Endstation Wien – Das Ende einer langen Reise durch die deutschsprachige IT-Welt**

## **Gesammelte Erfahrungen 1970-2015**

### **von Harry M. Sneed**

Die Geschichte der deutschsprachigen IT-Welt der letzten 45 Jahre aus der Sicht von Jemandem der alle Höhen und Tiefen miterlebt hat. Harry M. Sneed hat als Entwickler, Designer, Tester, Manager und Berater in über 50 Projekten mitgewirkt. Daneben hat er in der Zeit von Juni 1974 bis Juni 2014 mehr als 300 Seminare gehalten und 24 Bücher allein oder mit anderen verfasst. Er wurde mehrfach für seine Pionierleistungen im Softwaretest und Qualitätssicherung sowie in der Softwarewartung und -sanierung ausgezeichnet, von der GI, von der ASQF, von der ISTQB, von der ACM und von der IEEE. Er gründete den ersten Nearshore Outsourcing Betrieb in Budapest, von wo aus er für viele Jahre die Westdeutsche Industrie mit State of the Art Softwarewerkzeugen belieferte. Zu seinen Kunden zählten viele namhafte Anwenderbetriebe der DACH Länder. Es war aber nicht immer einfach. Er wurde mehrmals von deutschen und österreichischen Betrieben für seine Fehlleistungen auf die Straße gesetzt. Die Erfolge und Misserfolge glichen sich aus. Wer sich als Pionier an der Spitze der Softwaretechnologie platzieren will und das gleichzeitig in zwei Welten, muss damit rechnen. Zum Schluss ist er in Wien gelandet, eine Stadt zwischen Ost und West, wo manch eine gescheiterte Karriere sein Ende findet. Kaum jemand in der IT-Praxis hat mehr Erfahrungen gesammelt wie er. Jetzt hat Sneed in diesem einmaligen Buch diese Erfahrungen zusammengestellt. Die junge IT-Generation kann manches davon lernen, nicht nur im positiven sondern auch im negativen Sinne. Es werden immer noch dieselben Fehler gemacht wie vor 50 Jahren.



Das Buch kann über die Firma ZT Prentner IT GmbH am Kagraner Platz 40 in A-1220 Wien – [www.ztp.at](http://www.ztp.at) – für 32,- bestellt werden.

## **Projektmanagement und Vorgehensmodelle 2017 (PVM217): Die Spannungen zwischen den Menschen und den Prozessen im Projekt (Tagungsbericht)**

Prof. Dr. Martin Engstler, Sprecher WI-PM

Am 5. und 6. Oktober 2017 führten die GI-Fachgruppen Vorgehensmodelle (WI-VM) und Projektmanagement (WI-PM) die vierte gemeinsame Fachtagung „Projektmanagement und Vorgehensmodelle“ (PVM2017) unter dem Leitthema „Die Spannungen zwischen den Menschen und den Prozessen im Projekt“ durch. Gastgeber war in diesem Jahr die Hochschule Darmstadt. Die Durchführung der Fachtagung erfolgte in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V. und setzt damit die langjährige erfolgreiche Zusammenarbeit der Fachverbände fort.

Mit rund 100 Teilnehmern hat sich die PVM inzwischen als Fachtagung zum Projektmanagement in Deutschland etabliert. Die Teilnehmer schätzen die Verbindung des Fachprogramms mit fundierten Beiträgen aus Wissenschaft & Wirtschaft und dem im Tagungsprogramm fest integrierten intensiven Erfahrungsaustausch, der von den Teilnehmern auch als Plattform zur Weiterentwicklung eigener Ideen genutzt wird. Die Tagung besteht daher aus einem durch das Programmkomitee in einem Doppelt-Blind-Verfahren ausgewählten wissenschaftlichen Hauptprogramm, einem Future Track mit innovativen Themen und Praxisberichten, Kompaktbriefings sowie Open Spaces, bei denen die Teilnehmer eigene Themen einbringen und mit den anwesenden Experten vertiefen können.



Die Sprecher der veranstaltenden GI-Fachgruppen Projektmanagement und Vorgehensmodelle mit ihren Stellvertretern und der Sprecher der Fachgruppe IT-Projektmanagement der GPM (von links: Prof. Dr. Eckhart Hanser, Prof. Dr. Oliver Linssen, Alexander Volland, Prof. Dr. Martin Engstler, Dr. Masud Fazal-Baqaie) (Foto: Hochschule Darmstadt)

Im Mittelpunkt des Fachdiskurses der PVM2017 standen moderne Ansätze des Projektmanagements, die das Spannungsfeld zwischen dem Prozess und den Menschen im Projekt als gezieltes Gestaltungsfeld aufgreifen. Dabei wurden insbesondere Lösungsansätze für hybride Projektstrukturen vertieft und interdisziplinäre Lösungsansätze diskutiert.

Beide Konferenztagen wurden jeweils mit einem eingeladenen Keynote-Vortrag eröffnet. Am ersten Tag gab Jan F. Morgenthal (Deutsche Telekom) einen interessanten Einblick in die Praxis der agilen Produktentwicklung im Forschungs- und Entwicklungsumfeld der Künstlichen Intelligenz.

Den zweiten Konferenztag eröffnete Andreas Witt (Trumpf Werkzeugmaschinen) mit einem Erfahrungsbericht zum Einsatz agiler Projektmuster in dynamischen und adaptiven Produktentwicklungsprojekten im mittelständischen Maschinenbau.

Das Hauptprogramm beider Konferenztage umfasste elf hochkarätige Beiträge aus Praxis und Wissenschaft, die in einen wissenschaftlichen Review-Prozess mit einem Double Review-Verfahren ausgewählt wurden. Unter dem Leitthema der Tagung analysierten die Experten in ihren Fachbeiträgen Methoden und Konzepte für klassische, agile und hybride Projektmodelle mit besonderem Blick auf die darin agierenden Menschen sowie die hierbei relevanten Prozessregeln. Hierzu gehören managementorientierte Aspekte wie die Ausgestaltung der Rolle des Projektleiters (z.B. als Vermittler und Moderator) und die Bewertung der Teamperformance (z. B. die Umsetzung agiler Werte in den Projektteams), die Verfestigung moderner Führungsansätze (z. B. Umsetzung eines vollständigeren Scrum?), die Weiterentwicklung erprobter Modelle des SW-Projektmanagements (z. B. Assessment für die Softwareprozessverbesserung, Konzept eines situativ-reaktiven Software-Projektmanagement), die Erfassung und Optimierung von formal geprägten Projektprozessen (z. B. Kostenschätzung in hybriden Projekten, Sicherstellung notwendiger Dokumentationspflichten, Compliance-Aspekte) sowie der Beitrag des agilen Projektmanagements für eine agil geprägte Unternehmenskultur (z.B. Etablierung agiler Cluster, Transfer agiler Werte im Unternehmen). Mit dem „Best Paper Award“ der Fachgruppe Vorgehensmodelle wurde in diesem Jahr der Beitrag „Transition einer projektbasierten Produktentwicklungsorganisation in ein agiles Cluster – ein Werkstattbericht“ ausgezeichnet.

In sechs weiteren „Future Track-Vorträgen“, die vom Programmkomitee ausgewählt wurden, erhielten die Tagungsteilnehmer spannende Erfahrungsberichte sowie diskussionsanregende Impulse, die mit dem Auditorium direkt oder in ergänzenden Open Spaces vertieft wurden. Die Beiträge adressierten dabei sowohl den Umgang mit veränderten Rahmenbedingungen in den Prozessen (z.B. Implikation der neuen ISO-9001:2015 für die Prozesse, Ansätze der Softwareunterstützung) als auch Ansätze und Herausforderungen in der Projektkommunikation (z. B. Social Media-Einsatz in Change Management-Projekten) und falsche Erwartungen an agile Projektmuster (z.B. Fake News: Simply Agile).

Auf Anregung früherer Tagungsteilnehmer wurden im Programm der PVM2017 erstmals „Kompaktbriefings“ zu aktuell diskutierten praktischen Methoden des Projektmanagements angeboten (z.B. Projekt-Canvas, Projekt-Kanban, User Stories / Story Mapping, Tools für Change Manager). Dieses neue Format wurde zahlreich genutzt und wird daher in künftigen PVM-Tagungen beibehalten und ausgebaut werden.

Die Fachgruppen WI-VM und WI-PM bedanken sich bei den Sponsoren und insbesondere bei der Hochschule Darmstadt für die Bereitstellung der Tagungsräume und die hervorragende Unterstützung der Tagungsdurchführung und das Tagungsbüro vor Ort.

Die PVM wird 2018 fortgesetzt. Das Leitthema sowie der Termin und der Tagungsort werden Anfang 2018 auf der Internetseite [www.pvm-tagung.de](http://www.pvm-tagung.de) bekannt gegeben.

Der Tagungsband wurde in den GI Lecture Notes in Informatics (Band P276) veröffentlicht:

*Volland, A.; Engstler, M.; Fazal-Baqaie, M.; Hanser, E.; Linssen, O.; Mikusz, M. (Hrsg.): Projektmanagement und Vorgehensmodelle 2017. Die Spannung zwischen dem Prozess und den Menschen im Projekt, Lecture Notes in Informatics (LNI) - Proceedings, Volume P276, Bonn: Gesellschaft für Informatik & Köllen 2017 (ISBN 978-3-88579-670-1)*



# Workshop “Evaluation of Service-APIs – ESAPI 2017“

## Service APIs als Enabler einer erfolgreichen Digitalisierung

### (detaillierter Bericht)

Michael Binzen, Konrad Nadobny, Hendrik Neumann,  
André Nitze, Andreas Schmietendorf

## 1 Motivation zum Workshop

Im Internet als Service zur Verfügung gestellte Informationen, Funktionen und Algorithmen bestimmen in zunehmendem Maße die Art und Weise, wie neue Softwarelösungen implementiert werden. Im technologischen Sinne handelt es sich dabei um webbasiert zur Verfügung gestellte Daten und Service-APIs, die eine „ad hoc“-Integration in neue Lösungskontexte unterstützen. Entsprechende Lösungen finden sich z.B. mit mobilen Applikationen (Apps), Anwendungen der künstlichen Intelligenz (Data Science), im Telematikbereich (Smart City) oder auch im Internet der Dinge (IoT). Wollen Unternehmen von den Möglichkeiten der Digitalisierung profitieren, ist es erforderlich, die eigenen Bedürfnisse und Möglichkeiten realistisch bewerten zu können. Aus dieser Tatsache ergeben sich mindestens zwei Perspektiven auf Open-Data und Open-APIs: die Nutzersicht und die Entwicklersicht.

Darüber hinaus gilt es den gesamten Lebenszyklus der angebotenen Daten und APIs einem Management zu unterziehen. Ein solches API-Management muss sowohl den Bedürfnissen der Serviceentwicklung, einer heute zumeist agil durchgeführten Serviceintegration, als auch denen des betrieblichen Einsatzes der dann zusammengesetzten Services entsprechen. Typische Managementfunktionen beziehen sich auf den gesamten Lebenszyklus, die Zugriffskontrolle, das Monitoring, die Verrechnung oder auch auf die eingesetzten Laufzeitumgebungen der Service-APIs.

Ein besonderes Spannungsfeld ergibt sich aus offen und kostenfrei zur Verfügung gestellten Schnittstellen (Open Data/Open API) und ausschließlich kommerziell angebotenen Schnittstellen (API economy). Im Zusammenhang mit der API economy finden sich schnell wachsende Unternehmen (z.B. Fahrdienstvermittler Uber, Unterkunftsvermittler Airbnb, Onlineversand Zalando), die zwar einen breiten Marktzugang besitzen, aber nur eine geringe Ressourcenbindung aufweisen. Alleinstellungsmerkmale dieser die Digitalisierung treibenden Unternehmen beziehen sich auf die optimale Erfüllung der Kundenbedürfnisse. Dafür benötigte Leistungen werden von einem globalen Markt unter Verwendung der hier im Mittelpunkt stehenden Service-APIs im Sinne einer agil gebildeten Lieferantenkette bezogen. Darüber hinaus werden unternehmensexterne Softwareentwickler bewusst am Innovationsprozess bisher eher geschlossener Unternehmen beteiligt, so dass mit Hilfe von Service-APIs Möglichkeiten zur Gestaltung globaler einsetzbarer Softwarelösungen entstehen. Dieser Sachverhalt unterstreicht die strategische Bedeutung von derart zur Verfügung gestellten Informationen, Funktionen und Algorithmen.

Die aus dem Diskurs der GI-Fachgruppe „Software-Messung und –Bewertung“ hervorgegangene ESAPI-Initiative widmet sich multidimensionalen Bewertungsaspekten der vielfältig im Web angebotenen Service-APIs. Aus technologischer Sicht handelt es sich bei diesen Service-APIs um Web Services, Web-APIs, Microservices und Angebote, die unter dem Begriff Open Data bzw. Open API subsumiert werden. Mit der inhaltlichen Fokussierung wird den Themenschwerpunkten vergangener Veranstaltungen der BSOA/BCloud-Interessensgemeinschaft Rechnung getragen.



## 2 Beiträge des Workshops

Um der Zielstellung eines interaktiv gestalteten Workshops Rechnung zu tragen, wurden im Vorfeld Keynotes, Impulsvorträge und Kurzseminare durch ein deutschlandweit zusammengesetztes Programmkomitee ausgewählt.

Für die Inhalte der Beiträge sei auf den publizierten Tagungsband verwiesen. Folgende Autoren beteiligten sich daran mit ihren Beiträgen:

*Michael Binzen: (Deutsche Bahn AG):*

Keynote: Revolution durch Evolution – mit APIs schrittweise zu mehr Business Agility und neuen Geschäftsmodellen

*Dr. Frederik Kramer (initOS GmbH)*

Impulsvortrag: Case Study zur Bewertung einer sinnvollen Nutzung von Microservices

*Prof. Dr. Olaf Resch (HWR Berlin)*

Impulsvortrag: Die API-Economy braucht ein iUDDI

*Konrad Nadobny, Andreas Schmietendorf (Bayer AG, HWR Berlin)*

Impulsvortrag: Empirische Analysen von Open-Data und Open-API „Service“ Angeboten

*Steffen Kusterski (Toll Collect)*

Keynote: Compliance-Aspekte bei der Datenerhebung und Verarbeitung

*André Nitze (TH Brandenburg, OvG-Universität Magdeburg)*

Kurzseminar: Contract-first Development - Vom OpenAPI (Swagger) basierten Vertrag zur Implementierung

*Lukas Böhme, Jonas Grunert (Bayer AG)*

Kurzseminar: Bewertung der Vor- und Nachteile einer GraphQL basierten Abfrage von HTTPbasierten Service-APIs

## 3 Ergebnisse des World-Cafés

Eine der Zielstellungen des ESAPI-Workshops war, dass die Teilnehmer ihre Standpunkte, Sichtweisen und Ideen zu ausgewählten Fragen rund um das Thema Open API bzw. Open Data austauschen. Hierfür wurde die interaktive Methode eines World-Cafés gewählt, da es den Teilnehmern auf diese Weise ermöglicht wurde, in begrenzter Zeit ohne vorherige Vorbereitung die Themen strukturiert zu diskutieren.

### Impulsfragen zur Abstimmung

Im Vorfeld des Workshops wurden den Teilnehmern die folgenden Fragen zwecks Priorisierung und ggf. Feedback zur Verfügung gestellt:

1. Welche Bedeutung hat Open Data bzw. Open API im Kontext globaler Unternehmensarchitekturen (Platform-Ecosystems)?
2. Wie kann Open Data bzw. Open API zu Open Innovation führen und damit zum Wettbewerbsfaktor werden?

3. *Wie kann Open Data bzw. Open API in eine umfassende Digitalisierungsstrategie eingebettet werden?*
4. *Wie ist das Verhältnis von Open Data bzw. Open APIs zu den Rahmenbedingungen einer API-economy?*
5. *Wie können die Inhalte des Open Data bzw. Open API Manifests der BITKOM für das eigene Unternehmen operationalisiert werden?*
6. *Welche Einflüsse gilt es beim Management von Open Data und Open API Serviceangeboten zu berücksichtigen?*
7. *Welche Aspekte gilt es beim Sicherheits- und Compliance-Management zu berücksichtigen, was ist aktuell „state-of-the-art“?*

Durch die Teilnehmer wurden die Fragen 2, 3 und 7 für eine vertiefende Diskussion innerhalb der World-Cafés ausgewählt. Dies erfolgte per Mehrheitsabstimmung, wobei jeder Workshop-Teilnehmer vier Stimmen auf die vorgeschlagenen Themen frei vergeben konnte. Im Zusammenhang mit ausgewählten Fragen wurden erste Aspekte des korrespondierenden Leitfadens der BITKOM (vgl. [BITKOM 2017]) diskutiert.

### **Ergebnisse der Diskussion**

*Wie kann Open Data bzw. Open API in eine umfassende Digitalisierungsstrategie eingebettet werden?*

Open Data bzw. Open API ist als Instrument zur Umsetzung einer strategischen Entscheidung zur Öffnung des Unternehmens zu verstehen und somit ein wichtiger Enabler im Rahmen einer umfassenden Digitalisierungsstrategie. Damit Open Data/Open API erfolgreich für das Erreichen strategischer Ziele eingesetzt werden kann, müssen einige Faktoren in Betracht gezogen werden. Die Entscheidung für die Umsetzung von Open Data/Open API sollte zunächst auf Grundlage einer Nutzwertanalyse getroffen werden, wobei Risiken und Benefits auch unter Betrachtung von Case Studies gefunden und abgewogen werden sollten. Ausgehend von einer Analyse des Status-Quo (interne und externe Rahmenbedingungen) müssen dann Wege gefunden werden, wie Open Data/Open API strategisch umgesetzt werden kann.

Versteht man unter einer Strategie die Regeln und Maßgaben für das unternehmerische Handeln, so sind zunächst die Etablierung einer guten Governance und die Schaffung der Voraussetzungen wichtig. Das Bekenntnis zu Open Data/Open API steht am Anfang des Transformationsprozesses. Es muss ein Rahmenwerk etabliert werden, welches dem einzelnen Mitarbeiter die Sicherheit gibt, dass sein Handeln hin zu Open Data/Open API richtig und erwünscht ist. Dieses Rahmenwerk muss hierbei flexibel genug sein, dass es innovative Lösungen erlaubt und dabei gleichzeitig klar genug ist, um Unsicherheit zu vermeiden. Auch die organisatorische Struktur muss an die entsprechende Strategie angepasst werden. Ein Treiber für die Umsetzung von Open Data/Open API sind dabei die jeweiligen Business-Needs, welche die technischen Lösungen bestimmen. Diese zu erkennen, zu fördern und Erfolgsgeschichten zu kommunizieren ist essentiell für den Erfolg der Strategie. Nur durch die Schaffung der richtigen Strukturen, eine konstante Kommunikation und Reflektion sowie eine stringente Umsetzung im operativen Alltag kann der kulturelle und organisatorische Wandel erfolgreich umgesetzt werden, welcher für die Implementierung einer Open Data/Open API-Strategie erforderlich ist.

Zusammenfassend kann Open API/Open Data in eine umfassende Digitalisierungsstrategie eingebettet werden, indem die richtigen Rahmenbedingungen geschaffen werden und die Transformation in Einklang mit den Unternehmenszielen steht. Open Data/Open

API kann somit zu einem entscheidenden Treiber für die digitale Transformation und die Innovationskraft eines Unternehmens werden.

*Welche Aspekte gilt es beim Sicherheits- und Compliance-Management zu berücksichtigen, was ist aktuell „state-of-the-art“?*

Bei der Implementierung von Open Data/Open API stellt sich zwangsläufig die Frage, welche Sicherheitsrisiken den Vorteilen gegenüberstehen. Als Vorteile von Open Data/Open API gelten vor allem das freigesetzte Innovationspotential sowie die Öffnung für Synergien, neue Geschäftsmodelle, die Demokratisierung der Daten und Open Data/Open API als Enabler für neue Lösungen. Dem gegenüber stehen Risiken in Bezug auf Datenschutz und Datensicherheit sowie der allgemeinen Sicherheit der über Open API angebotenen Services. Diese Risiken sind beim Sicherheits- und Compliance-Management zu berücksichtigen.

Der freie Zugriff auf Daten und Services über Open Data/Open API eröffnet zunächst das Risiko des kriminellen Missbrauchs, gezielter Angriffe auf die Services sowie der Ownership und Haftung für Daten und Services auf der Anbieterseite. Auf der Nachfrageseite besteht das Risiko der Fehlinterpretation der Daten und Unsicherheit bezüglich richtiger Funktion, Verlässlichkeit und Verfügbarkeit der nachgefragten Services. Zudem sollte der Service einfach und unkompliziert eingebunden werden können.

Den meisten dieser Herausforderungen lässt sich bereits mit bestehenden Lösungen begegnen. Die Handhabung des Datenschutzes ist eine grundsätzliche Frage, wobei hier die Handhabung unternehmensintern und extern nicht variieren sollte und der Gesetzgeber hier bereits sehr klare Vorgaben macht. Bezüglich der Zusammenarbeit von Anbieter und Nutzer sind vertragliche Lösungen, wie SLA's, Nutzungsbedingungen, Zertifizierungen etc. ein Instrument zur gegenseitigen Absicherung. So lassen sich Verlässlichkeit und eine verbindliche Haftung etablieren. Auch Transparenz über Datenherkunft und -Nutzung sowie die Implementierung von Standards und die gute Dokumentation auf beiden Seiten sind als integrale Instrumente des Sicherheits- und Compliance-Managements zu verstehen. Als Erfolgsbeispiel und Vorbild können Open-Source-SW-Lösungen dienen.

*Wie kann Open Data bzw. Open API zu open Innovation führen und damit zum Wettbewerbsfaktor werden?*

Open Data/Open API ist mit einer Vielzahl von Innovationsvorteilen verbunden. Zunächst entsteht über die strukturierte Öffnung große Transparenz bezüglich verfügbarer Services. Durch das stark vereinfachte Einbinden von Services können Parallelstrukturen abgebaut, Synergien genutzt und sogar gänzlich neue Geschäftsmodelle aufgebaut werden. Die Konzentration auf die Bereitstellung einzelner Services ermöglicht es, dass sich Unternehmen auf ihr Kerngeschäft konzentrieren und unnötige Kosten reduzieren. Die Öffnung für externe Nutzer eröffnet zudem die Möglichkeit für neue Nutzungskontexte, eine bessere Sichtbarkeit und Kundengewinnung sowie kontinuierliche Verbesserung durch Messen und Feedback. Zusammen mit der Nutzung gemeinsamer Ressourcen führt dies zu einer Win-Win-Situation für alle Beteiligten.

Damit die Vorteile von Open Data/Open API wirklich ausgeschöpft werden können und die Öffnung nicht zur überraschenden Disruption führt, müssen sich Unternehmen auf diese Transformation einstellen und sie aktiv vorantreiben. Ohne entsprechende Maßnahmen drohen Verlust von Kontrolle und bestehenden Wettbewerbsvorteilen durch exklusiven Zugriff auf die eigenen Daten und Services.

Letztendlich wird Open Data/Open API durch die einfache und unkomplizierte Verknüpfung von Services zu Innovation innerhalb wie außerhalb des Unternehmens führen, von der alle Beteiligten profitieren werden.

## 4 Weitere Informationen

Auch für das Jahr 2018 ist die Durchführung eines ESAPI-Workshops vorgesehen, der in München stattfinden wird. Weiterführende Informationen werden zeitnah unter der folgenden URL im Internet bereitgestellt:

<https://blog.hwr-berlin.de/schmietendorf/>



Abbildung 1: Tagungsband zum Workshop ([Schmietendorf 2017])

## 5 Quellenverzeichnis

[Schmietendorf 2017] Schmietendorf, A. (Hrsg.): ESAPI 2017 - 1. Workshop Evaluation of Service-APIs, in Berliner Schriften zu modernen Integrationsarchitekturen, Shaker-Verlag, Aachen, November 2017

[Bitkom 2017] Leitfaden Open Data - Neue Konzepte erfolgreich umsetzen, URL: <https://www.bitkom.org/Bitkom/Publikationen/Open-Data-Neue-Konzepte-erfolgreich-umsetzen.html>, Nov. 2017

## Dank

Unser Dank gilt den Referenten und Teilnehmern, aber auch den Sponsoren (Deutsche Bahn AG, T-Systems International GmbH und SEVEN PRINCIPLES AG), die eine solche Veranstaltung überhaupt erst ermöglicht haben. Ein herzlicher Dank geht auch an die beteiligten Medienpartner SIGS DATACOM GmbH aus Köln, dpunkt.verlag GmbH aus Heidelberg und an den Shaker Verlag GmbH aus Aachen.

## **Der Fachausschuss und die Fachgruppen WI-VM, WI-PM, WI-PrdM stellen sich vor**

### **Fachausschuss WI-MAW:**

#### ***Management der Anwendungsentwicklung und -wartung***

Anwendungssysteme sind aus Sicht der Wirtschaftsinformatik Aufgabenträger im Rahmen der Erfüllung der betrieblichen Gesamtaufgabe. Ihre Aufgabenstellungen werden aus den Unternehmenszielen und den strategischen Zielen der Informationsverarbeitung abgeleitet. Die Entwicklung von Anwendungssystemen erfolgt nicht "kontextfrei", sondern i.A. in einem bestimmten betrieblichen Umfeld. Dies bedeutet zum einen, dass sich das einzelne Anwendungssystem in bereichsübergreifende bzw. unternehmensweite Daten- und Funktionsmodelle oder Objektmodelle einordnen muss. Zum anderen existieren häufig bereits Anwendungen für andere betriebliche (Teil-)Aufgaben, mit denen das System zusammenarbeiten muss.

Der Fachausschuss beschäftigt sich aus dieser Sicht mit der Planung, der Entwicklung, der Einführung, dem Einsatz und der Wartung betrieblicher Anwendungssysteme. Im Vordergrund stehen Vorgehensweisen, Prinzipien und Methoden für die Anwendungsentwicklung im betrieblichen Umfeld sowie ihre Unterstützung durch Softwarewerkzeuge. Im Einzelnen setzt sich der Fachausschuss mit Themen wie den folgenden auseinander:

- Integration von Anwendungssystemen in eine existierende betriebliche DV-Landschaft;
- Sicherung der Investitionen in das Wirtschaftsgut Software; Bewertung von Vorgehensmodellen, Methoden und Werkzeugen zur Anwendungsentwicklung sowie Einsatzerfahrungen;
- Management von Softwareentwicklungsprojekten (Projektplanung, -durchführung und -kontrolle, Projektorganisation, Projektmanagementsysteme, Kosten/ Wirtschaftlichkeit),
- Software Produktmanagement, Configuration Management, Change Management, Migration Management, Reengineering.

Mitgliederzahl: ca. 500

#### **FA-Sprecher**

Prof. Dr. G. Herzwurm  
Universität Stuttgart  
Lehrstuhl für Allgemeine  
Betriebswirtschaftslehre und  
Wirtschaftsinformatik II  
(Unternehmenssoftware)

#### **stellv. FA-Sprecherin**

Dr.-Ing. Birgit Demuth  
Technische Universität Dresden  
Institut für Software- und Multimediatechnik

## **Fachgruppe WI-VM:**

### ***Vorgehensmodelle für die betriebliche Anwendungsentwicklung***

Betrachtungsgegenstand der Fachgruppe sind die als "Vorgehensmodelle" bezeichneten Beschreibungen der Aufbau- und Ablauforganisation von Projekten zur Entwicklung und Wartung von Anwendungssystemen. Solche Beschreibungen helfen, die Durchführung von Projekten innerhalb eines Unternehmens oder darüber hinaus zu standardisieren und zu verbessern. Der Begriff Anwendungssystem sei hier sehr weit gefasst: von technischen über betriebswirtschaftliche bis zu organisatorischen Systemen.

Um eine effektive und effiziente Gestaltung der Vorgehensmodelle und damit der Projekte zu erreichen, ist die Berücksichtigung der Schnittstellen zur Betriebswirtschaftslehre einerseits, insbesondere der Organisations- und der Managementlehre, und dem Software Engineering andererseits wesentlich.

Das Thema "Vorgehensmodelle" wird daher von der Fachgruppe aus verschiedenen Blickrichtungen betrachtet:

- Grundlagen: Begriffsdefinitionen, Bestandteile, (formale) Beschreibung von Vorgehensmodellen, Vorgehensmodell-Typen.
- Inhaltliche Bausteine: Konzepte, Methoden, Phasen, Projektmanagement, Qualitätssicherung.
- Werkzeugunterstützung: Vorgehensmodell-Driver, Meta-Modelle, Data-Dictionaries.
- Ökonomische, soziale und psychologische Aspekte: Einführung und Betrieb von Vorgehensmodellen, organisatorisches Umfeld.
- Beispiele aus der Praxis: Standard-Vorgehensmodelle in Organisationen, Branchen und für Anwendungstypen, spezielle Vorgehensmodelle von Unternehmen.
- Standardisierung von Vorgehensmodellen: V-Modell XT, Hermes

Die Fachgruppe fördert einen intensiven Gedankenaustausch durch die Pflege persönlicher Kontakte und unterstützt einen offenen und kritischen Dialog zwischen Wissenschaft und Praxis. Ein weiteres Ziel der Fachgruppe ist die Erarbeitung von Empfehlungen und Stellungnahmen zu den technischen, wirtschaftlichen, organisatorischen und sozialen Aspekten bei Auswahl und Einsatz von Vorgehensmodellen - dies insbesondere vor dem Hintergrund nationaler, europäischer und internationaler Normungs- und Standardisierungs-bestrebungen. Weitere Informationen über Vorgehensmodelle und die Arbeit der Fachgruppe sind im Internet zu finden unter [www.vorgehensmodelle.de](http://www.vorgehensmodelle.de).

#### **FG-Sprecher**

Prof. Dr. Eckhart Hanser  
Duale Hochschule Baden-Württemberg  
Lörrach

#### **stellv. FG-Sprecher**

M. Sc. Masud Fazal-Baqaie  
Fraunhofer-Institut für Entwurfstechnik  
Mechatronik IEM

## **Fachgruppe WI-PM: *Projektmanagement***

Die Fachgruppe befasst sich mit dem Einsatz, der Verbreitung sowie der Weiterentwicklung des Projektmanagements. Neben Vertretern aus den Hochschulen sollen vor allem Praktiker die Arbeitsschwerpunkte der Fachgruppe definieren, Ergebnisse erarbeiten und Erfahrungen weitergeben. Für die Aufgabengebiete des Projektmanagements sollen Methoden, Werkzeuge und Techniken untersucht werden. Neben den klassischen Aufgabengebieten wie beispielsweise Projektorganisation, Aufwandschätzung, Projektverfolgung und Projektsteuerung stehen folgende Themen im Vordergrund:

### **Bedeutung und Dimensionierung des Projektmanagements.**

Die Bedeutung des DV-Projektmanagements als entscheidender Faktor für den Erfolg oder das Mißlingen von DV-Projekten wird von vielen Entscheidungsträgern unterschätzt. Daher sollte die grundsätzliche Bedeutung sowie der Nutzen einer angemessenen Ausstattung des Projektmanagements mit eigenen Ressourcen transparent gemacht werden.

### **Human Factors.**

In zahlreichen Projekten liegen die größten Projektrisiken bei den sogenannten Human Factors (oder "weichen" Faktoren). Der Umgang mit solchen Risiken erfordert Kompetenz bei Themen wie Motivation, Führung, Teamfähigkeit, Überwindung "politischer" Widerstände u.a.m.

### **Programm Management.**

Immer öfter gefordert wird das Management eines Portfolios von Projekten, wobei nicht alle Projekte des Portfolios eigentliche DV-Projekte zu sein brauchen. Solche Projektportfolios können beispielweise als Folge einer veränderten Unternehmensstrategie entstehen und sollen dann einen größeren Veränderungsprozess bewirken. Hauptaufgabe eines Programme Managements ist dabei die zielorientierte Steuerung der Abarbeitung des Projektportfolios, wobei insbesondere unternehmerische Gesichtspunkte zu beachten sind.

### **FG-Sprecher**

Prof. Dr. Martin Engstler  
Hochschule der Medien Stuttgart

### **stellv. FG-Sprecher**

Alexander Volland  
Union IT-Services GmbH, Frankfurt am Main

## **Fachgruppe WI-PrdM:** ***Software Produktmanagement***

Effizientes und effektives Management softwareintensiver Produkte ist zu einer kritischen Kernkompetenz von Unternehmen geworden. Unternehmen sind mit einer stetig wachsenden Anzahl von Herausforderungen konfrontiert, die durch unterschiedliche Lebenszyklen von Systemen und unterschiedliche Kritikalität im System Einsatz in immer mehr – und neuen – Anwendungsfeldern entstehen. Hybride Systeme, z.B. im Internet-of-Things, in Automobilen, Flugzeugen, Drohnen, medizinischen Geräten oder in der Unterhaltungselektronik geben Software eine nie dagewesene Bedeutung. Zusätzlich entstehen durch die vielfältigen Initiativen im Rahmen Digitalisierung neue Arbeits- und Geschäftsmodelle und eröffnen vollkommen neue, durch Software getriebene Möglichkeiten zur Innovation.

In diesem dynamischen Umfeld findet softwaregetriebene Innovation an der Schnittstelle zwischen Informatik/Software Engineering und Wirtschaft statt, zwischen Forschung und industrieller Praxis. Das Produktmanagement umfasst hierbei die Entwicklung, Wartung und Evolution klassischer Softwarelösungen im gesamten Produktlebenszyklus, aber insbesondere auch innovative softwarebasierte Innovation. Die Fachgruppe befasst sich einerseits mit Konzepten, Methoden und Werkzeugen der Informatik/Wirtschaftsinformatik zur Gestaltung des Produktmanagements und der Produktinnovation. Andererseits wird insbesondere auch ein starker Fokus auf die praktische Anwendbarkeit theoretischer Konzepte gelegt.

Die Fachgruppe fördert auf dem genannten Gebiet den intensiven Gedankenaustausch, die Pflege persönlicher Kontakte und die Zusammenarbeit interessierter Personen und Gruppen. Dazu zählt u.a. die gegenseitige Information über Veranstaltungen, Projekte und Veröffentlichungen.

### **FG-Sprecher**

PD Dr. Marco Kuhrmann  
Technische Universität Clausthal  
Institut für Informatik &  
Institute for Applied Software Systems  
Engineering

### **stellv. FG-Sprecher**

Prof. Dr. Jürgen Münch  
Hochschule Reutlingen  
Herman Hollerith Zentrum



## **Mitglieder des Fachausschusses Management der Anwendungsentwicklung- und wartung (GI-MAW)**

Die Mitglieder des Leitungskreises des Fachausschusses finden Sie unter:

<http://fa-wi-maw.gi.de/leitungskreis.html>

# Impressum

Der Rundbrief des Fachausschusses *Management der Anwendungsentwicklung und -wartung (WI-MAW)* ist das Publikationsorgan des Fachausschusses sowie der Fachgruppen

WI-VM *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*

WI-PM *Projektmanagement*

WI-PrdM *Software Produktmanagement*

Der Rundbrief erscheint einmal jährlich. Er wird den Mitgliedern der Fachgruppen zugesandt. Durch den Rundbrief sollen wichtige Erfahrungen, neue Erkenntnisse und aktuelle Informationen unter den Mitgliedern ausgetauscht werden. Rundbriefbeiträge von Mitgliedern und Interessenten sind daher besonders willkommen. Es können Beiträge zu folgenden Rubriken eingereicht werden:

- Fachbeiträge: *Erfahrungsberichte; Theoretische Beiträge; Projektberichte (auch über laufende Projekte)*
- Informationen: *Buchbesprechungen; Tagungsberichte; Vorstellung von Arbeitsgruppen;*
- Leserbriefe: *Veranstaltungen; Call for Papers; Einladungen; Programme*

Es wird gebeten, Beiträge in elektronischer Form (Word) an die Rundbriefredaktion zu senden. Ein Ausdruck sollte keine Seitennummerierung enthalten, wegen der Verkleinerung auf DIN A5 jedoch einen großen Rand (mindestens 2,5 cm) sowie eine Schrift von mindestens der Größe wie Times Roman 12.

Die Beiträge können in deutscher oder englischer Sprache abgefasst sein. Mit der Zusendung eines Beitrags ist das Einverständnis zur Veröffentlichung im Rundbrief verbunden. Jeder Beitrag wird ohne Begutachtung veröffentlicht.

**Herausgeber** Fachausschuss *Management der Anwendungsentwicklung und -wartung*

**Auflage** 500

**Redaktion**

Christian Kop  
Institut für Angewandte Informatik  
Alpen-Adria-Universität Klagenfurt  
A-9020 Klagenfurt

E-mail: christian.kop@aau.at

Tel.: +43 463 2700 3735

Fax: +43 463 2700 993735

**Redaktionsschluss für das nächste Heft: 31.01.2019**