

NORDBLICK

Heft 1 | April 2016 | Forschung an der NORDAKADEMIE

BIG DATA **Hochschullehre** **ATAM** **Virtuelles Klassenzimmer** **Produktspezifikation** **webbasierte Systeme** **Industrie 4.0** **ALMA** **Wertschöpfungsketten** **Lean IT-Management** **Komplexität** **Produktdefinition** **Bewertung** **IT-Organisation** **MOOC** **Produktentwicklungsprozess** **E-Learning-Didaktik** **kundenindividuelle Produktentwicklung** **SOFTWAREARCHITEKTUR** **AUDITING** **Revision** **Produktentstehungsprozess** **Enterprise Resource Planning** **Digitalisierung** **Open Source** **E-Learning** **Review** **Bibliotheken** **Audit** **Inverted Classroom Model**

NORDBLICK

Heft 1 | April 2016 | Forschung an der NORDAKADEMIE



Impressum

NORDBLICK
Forschung für die Wirtschaft
Heft 1 | April 2016
ISSN-Print 2367-1173

Herausgeber:
Präsidium der NORDAKADEMIE – Hochschule der Wirtschaft
Köllner Chaussee 11 | D-25337 Elmshorn

Redaktion:
Prof. Dr. habil. Stefan Behringer

Redaktionsassistenten:
Simon Hachenberg, M.A.
Eline Joosten, M.A., M.A.
Köllner Chaussee 11 | D-25337 Elmshorn
Telefon (04121) 4090 0
nordblick@nordakademie.de

Eine Sonderausgabe erscheint im Juni 2016.
Die nächste reguläre Ausgabe erscheint voraussichtlich im Dezember 2016.
Redaktionsschluss: 1. Oktober 2016
Zusendung von Beiträgen bitte an obengenannter E-Mail-Adresse.

INHALTSVERZEICHNIS

Vorwort	3
<i>Stefan Behringer</i>	

Ingenieurwesen

Prozesse für die kundenindividuelle Produktentwicklung.....	4
<i>Bernhard Meussen</i>	

Informatik

Merkmale von Open-Source-Entwicklungsprozessen in Bibliotheken	20
<i>Matthias Finck</i>	
Indikatorenbasierte Audits – ein stringenter Ansatz zur Prüfung von Geschäftsprozessen? .	30
<i>Nick Gehrke</i>	
Komplexitätsmanagement für IT-Organisationen.....	46
<i>Arno Müller und Hinrich Schröder</i>	
Kritischer Vergleich von Methoden zur Bewertung von Softwarearchitekturen.....	60
<i>Stefanie Jasser und Joachim Sauer</i>	

E-Learning

Einsatzszenarien von Adobe Connect an der NORDAKADEMIE	72
<i>Cristina Trujillo, Simon Hachenberg und Faria Essenwanger</i>	
Ein Massive Open Online Course als E-Learning-Komponente im Inverted Classroom Modell.....	80
<i>Frank Zimmermann und Uwe Neuhaus</i>	

VORWORT

Liebe Leserinnen und Leser,

Sie halten das erste Exemplar von NORDBLICK in den Händen, der Zeitschrift für anwendungsorientierte Forschung an der NORDAKADEMIE Hochschule der Wirtschaft. Zwei Mal jährlich erhalten Sie ab sofort mit dem NORDBLICK Einblicke in die Forschungsarbeit an der NORDAKADEMIE, Ausblicke auf interessante Themen der Wissenschaft und Durchblick in aktuellen Fragestellungen der Wirtschaftswissenschaften, Informatik, Ingenieurwissenschaften und Didaktik. Damit zeigt die NORDAKADEMIE, dass sie eine forschungsstarke Hochschule ist, bei der die Anwendungsorientierung im Mittelpunkt steht – ganz im Sinne der Forschungsphilosophie Forschung mit der Wirtschaft, für die Wirtschaft und mit den Studierenden im Mittelpunkt. Verbunden mit der Darstellung von Forschungsergebnissen ist auch die Aufforderung an Sie, liebe Leserinnen und Leser, uns zu fordern. In unseren Kompetenzfeldern sind wir gerne Partner für die Wirtschaft und entwickeln gemeinsam mit Ihnen Lösungen für Ihre spezifischen Probleme.

In diesem Heft stellen wir Ihnen Ergebnisse aus den Bereichen Ingenieurwesen und Informatik vor. Dabei stellen alle Autorinnen und Autoren einen direkten Bezug zur Anwendung in der Wirtschaft her. Im letzten Teil dieser ersten Ausgabe werden dann aktuelle Aspekte aus der Lehre aufgenommen. In den Aufsätzen werden neue Ansätze der Didaktik dargestellt, die bereits in der Lehre an der NORDAKADEMIE im Einsatz sind. Damit wird ein wesentlicher Bezug zwischen Lehre und Forschung hergestellt. Es ist notwendig, dass Forschung unmittelbaren Eingang in die Lehre findet. Damit wird sichergestellt, dass die Lehre sowohl inhaltlich-fachlich als auch didaktisch auf aktuellem und höchstem Niveau ist. Dies ist unser Anspruch, den wir auch mit dem NORDBLICK erfüllen wollen. Wir freuen uns über Anregungen, Kritik und natürlich auch Lob.

Ich wünsche Ihnen eine interessante Lektüre!

Prof. Dr. habil. Stefan Behringer

Präsident

PROZESSE FÜR DIE KUNDENINDIVIDUELLE PRODUKTENTWICKLUNG



Prof. Dr.-Ing. Bernhard Meussen
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: Mit zunehmender Komplexität von Produkten steigt auch die Komplexität der Prozesse, mit denen in Unternehmen diese Produkte entwickelt werden müssen. Gleichzeitig verkürzen sich die am Markt realisierbaren Entwicklungszyklen. Die Herausforderungen der Digitalisierung der Wertschöpfungsketten verschärft diese Entwicklung. Im Folgenden wird ein Beitrag zur Problembeschreibung gegeben und es werden Ansätze zur Beherrschung dieser Herausforderungen in der Produktentwicklung aufgezeigt.

Keywords: Digitalisierung der Wertschöpfungsketten, Industrie 4.0, Produktentstehungsprozess, Produktentwicklungsprozess, kundenindividuelle Produktentwicklung, Produktdefinition, Produktspezifikation

1. EINLEITUNG

Unter dem Schlagwort „Industrie 4.0“ wird heute im Allgemeinen die Digitalisierung des Produktlebenszyklus zur Sicherstellung der Wettbewerbsfähigkeit der Wertschöpfungsketten verstanden (BITKOM et al. 2015). Neben vielen weiteren Aspekten der Digitalisierung spielen die Potentiale der Produktion von Produkten mit kundenspezifischen Eigenschaften durch die Technologien der 4. industriellen Revolution eine große Rolle. Unter Produkten mit kundenindividuellen Eigenschaften wird hier nicht nur die vorgedachte Konfiguration von Varianten, sondern auch die kundenspezifische Ausprägung eines Produkts mit kundenspezifischen Entwicklungs- und Produktionsschritten betrachtet. In diesem Beitrag werden Überlegungen zur Gestaltung produktindividueller Entwicklungsprozesse und deren produktionstechnischer Realisation dargelegt. In der Produktion werden dabei neben der Anwendung der Smart Products in der Smart Factory (siehe hierzu z. B. Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft (Hrsg.) 2013: 24 ff.) und Methoden der additiven Fertigung (siehe hierzu z. B. Feldhusen & Grote 2013: 48ff.) berücksichtigt. In der Smart Factory wird die durchgängige Digitalisierung der Wertschöpfungsketten durch Nutzung sogenannter „Cyber-Physical Systems“ (CPS) genutzt.

Bisher wird die wissenschaftliche Diskussion hierzu stärker durch technik- als nutzengetriebene Sichtweisen auf die zweifelsohne großen Potentiale der Vision „Industrie 4.0“ (Bauernhansl 2014: 8) geprägt. Die technikgetriebene Sichtweise führt dazu, dass geeignete Beispiele für die Anwendung der Vision „Industrie 4.0“ häufig fehlen oder Umsetzungen, z. B. durch Demonstratoren, nur einzelne Aspekte zeigen (Meussen 2015a). Folgerichtig hat die „Plattform Industrie 4.0“ in ihrem Ergebnisbericht als Empfehlung für die Kernbausteine der Umsetzung der Vision Industrie 4.0 für die nächsten Jahre die „Methoden für Geschäftsmodelle“ an den Anfang gesetzt (BITKOM et al. 2015: 15ff.).

Prof. Dr.-Ing. Bernhard Meussen hat seit 2013 eine Professur für Maschinenbau an NORDAKADEMIE inne. Nach dem Studium des Maschinenbaues mit dem Schwerpunkt Konstruktionstechnik hat er zu einem Thema aus der Kontinuumsmechanik promoviert. Nach der Promotion war er in verschiedenen technischen Leitungs-funktionen tätig. Von 2004 bis 2013 war er für die Entwicklung, Arbeitsvorbereitung und Qualitätssicherung verschiedener Werke eines weltweit führenden Herstellers mobiler Arbeitsmaschinen verantwortlich. Sein Forschungsgebiet umfasst die Produktentwicklung und die numerische Simulation.
E-Mail: bernhard.meussen@nordakademie.de

Geschäftsmodelle dienen der Umsetzung der Unternehmensstrategie und im Sinne produzierender Unternehmen ist dies die Vermarktung des zu verkaufende Produkts. Das Produkt kann selbst ein CPS sein. Bauernhansl beschreibt den Nutzen und die Herausforderungen für die Produktentwicklung und identifiziert als einen wichtigen Nutzen der Vision „Industrie 4.0“ die Fähigkeit zur Produktindividualisierung und den damit möglichen neuen, disruptiven Geschäftsmodellen (Bauernhansl 2014: 11). Der Forschungsbedarf ergibt sich hierbei durch die Tendenz mittelständischer Unternehmen z. B. des Maschinenbaus, die Digitalisierung bisher insbesondere zur sog. digitalen Veredelung ihrer Produkte zu verwenden und disruptive Produktentwicklungen zu vernachlässigen (Bauernhansl et al. 2015: 37ff.).

Hier soll dieser Beitrag ansetzen. Im Folgenden wird der Begriff des individuellen Produkts insbesondere aus der Sicht der Produktentwicklung definiert. Die Anforderungen der Vision „Industrie 4.0“ und der wandlungsfähigen Fertigung (Steegmüller et al. 2014: 103ff.) als produktionstechnische Umsetzung eines individuellen Produkts werden erörtert. Geeignete Produktentstehungsprozesse werden betrachtet und ein Vorschlag für einen solchen Produktentwicklungsprozess auf Basis erster Überlegungen gemacht (Meussen 2015).

2. PRODUKTE MIT KUNDENSPEZIFISCHEN EIGENSCHAFTEN

Lindemann sieht als Konsequenz der zunehmenden Fokussierung der Unternehmen auf den Kundennutzen die individualisierte Massenproduktion als strategisches Konzept an (Lindemann et al. 2006: 7ff.). Bauernhansl sieht durch die zunehmende Globalisierung der Märkte bei steigender Regionalisierung der Produkte ebenfalls einen Trend zur Produktindividualisierung (Bauernhansl 2014: 10). Dabei werden zwei Ansätze zur Produktindividualisierung unterschieden:

- variantenreiche Produkte und
- kundenspezifische Produkte.

Nach Lindemann entstehen individualisierte Produkte durch individuelle Gestaltung in vorgeplanten Strukturen (Lindemann et al. 2006: 9). Bauernhansl sieht die Vision Industrie 4.0 als eine Voraussetzung für die Nutzung der Potentiale individualisierter Produkte. Dabei kann die Individualisierung auch durch Komplexitätserhöhung durch Varianten erreicht werden. Variantenreiche Produkte sind aus vielen Bereichen bekannt, weit verbreitet sind die Konfiguratoren z. B. im Automobilbau, bei dem sich Kunden aus einer Vielzahl von Kombinationsvarianten ein individuelles Fahrzeug zusammenstellen können. Die variantenreichen Produkte führen zu großer Produktkomplexität, da die möglichen Varianten alle vorentwickelt werden. Komplexitätsmanagement durch Variantenmanagement ist eine Methode zur Beherrschung solch variantenreicher, komplexer Produkte (siehe z. B. Leichnitz & Eilmus 2013; Ehrlenspiel & Meerkamm 2013: 714ff.). Bekanntes Beispiel ist der Querbaukasten der Volkswagen AG (Volkswagen AG 2012).

Alternativ kann untersucht werden, wie kundenindividuelle Produkte entstehen können, die ohne die in allen Einzelheiten vorgedachte Struktur der variantenreichen Produkte auskommen. Vorbild kann hier der Anlagen- und Sonderbau sein (Lindemann et al. 2006; Meussen 2015b).



Abbildung 1: Trend zu kundenindividuellen Produkten in Anlehnung an (Lindemann et al. 2006)

3. ANFORDERUNGEN AN DIE KUNDENSPEZIFISCHE PRODUKTENTWICKLUNG

Die Produktentwicklung als Teil des Produktentstehungsprozesses hat die Bereitstellung der zur Erfüllung des Geschäftsmodells erforderlichen Produkte zum Ziel. Dabei wird durch den Produktentwicklungsprozess sichergestellt, dass

- Funktion,
- Qualität,
- Gestalt und
- Kosten

des Produkts anforderungsgerecht umgesetzt werden (Ehrlenspiel & Meerkamm 2013: 162). Diese Anforderungen müssen natürlich auch dann umgesetzt werden, wenn Teile des Produkts kundenindividuell, d. h. vom Kunden bestimmt, ausgeführt werden. Ansätze zur Beteiligung des Kunden in der Produktplanung finden sich im Konzept der Open Innovation wieder (Reichwald & Piller 2006: 95ff.). Grundlegende Überlegungen findet man auch in (Baumberger 2007).

Dabei muss z. B. für den Geltungsbereich der Maschinenrichtlinie (Europäisches Parlament 2006) die Produktsicherheit gewährleistet werden. Die Maschinenrichtlinie schreibt für die Produkte eine entsprechende Produktdokumentation vor, die die Nachweise der konformen Produkteigenschaften enthält. Wie können diese für zunächst unbekannte, da kunden-spezifische Teile des Produkts erbracht werden, wenn nicht alle Produktvarianten im Vorwege ausgeplant werden?

Weiterhin müssen marktfähige Kosten, Qualitäten und Erstellungszeiten realisiert werden. Dazu sind die Unternehmensprozesse entsprechend zu gestalten. Insbesondere in der Fertigung müssen Verfahren verwendet werden, die den kundenindividuellen Anteil eines Produkts realisieren können. Die Vision Industrie 4.0 ermöglicht solche Verfahren, die zur Umsetzung der Anforderungen kundenindividueller Produkte genutzt werden können. Einige dieser Elemente werden im Folgenden diskutiert.

4. ANFORDERUNGEN AN DIE SMART FACTORY UND DAS SMART PRODUCT

In der Vision Industrie 4.0 spielt die sog. „Smart Factory“ eine zentrale Rolle (Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft 2013). Die Smart Factory besteht aus „Cyber-Physical Production Systems“ (CPPS), die eine flexible Produktion der sich selbst steuernden CPS als Produkte (Smart Products) ermöglichen. Um als Smart Product in der Smart Factory agieren zu können, müssen die CPS-Produkte Funktionalitäten aufweisen, die einer standardisierten Wechselwirkung mit der Umgebung der Smart Factory genügen, z. B. dem Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) (VDI e.V. et al. 2015).

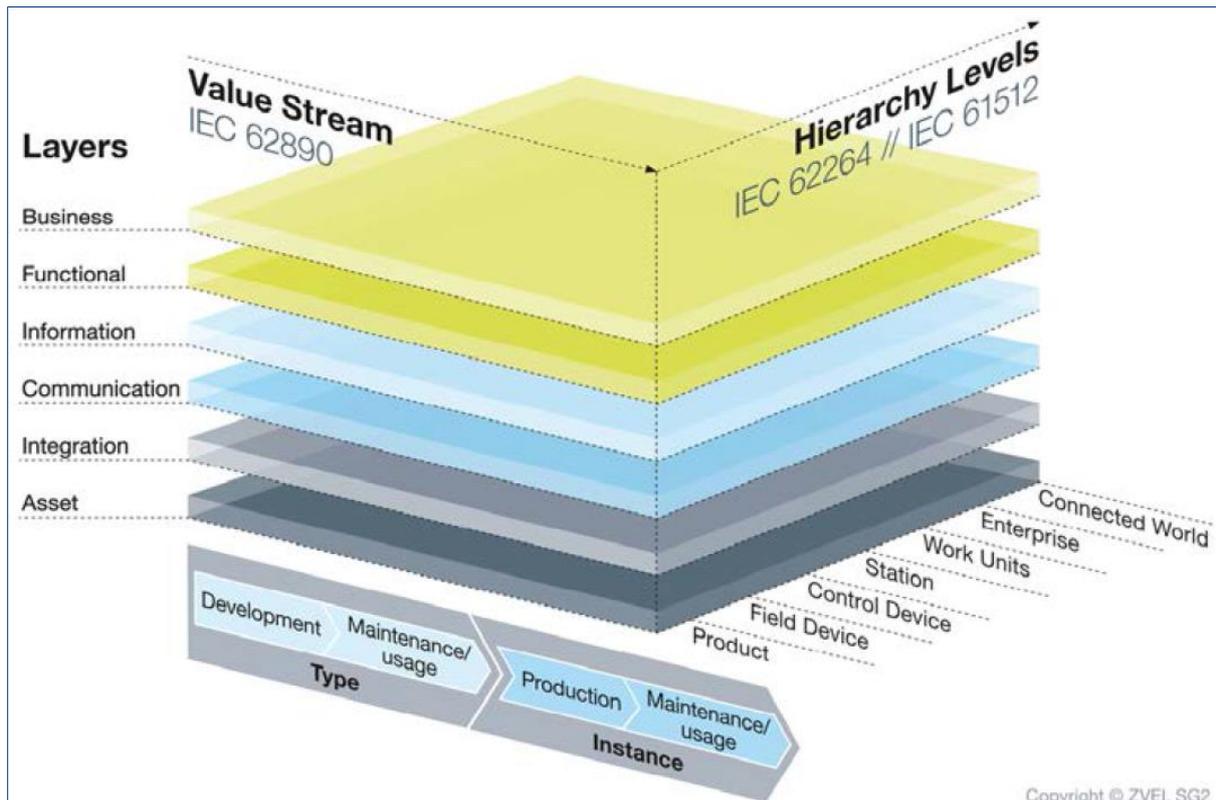


Abbildung 2: Referenzarchitekturmodell Industrie 4.0 nach (VDI e.V. et al. 2015)

Da die Produktionstechnik der Smart Factory auf eine möglichst wandlungsfähige Fertigung ausgelegt ist, in der z. B. fahrerlose Transportsysteme Logistikaufgaben übernehmen und kollaborierende Roboter in der Produktion mit Menschen eingesetzt werden (Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft 2013), spielen in der Fertigung dann Verfahren zur Herstellung von produktspezifischen Zeichnungsteilen ohne spezielle Werkzeuge eine große Rolle. Großes Potential haben hier die Verfahren der additiven Fertigung. Additive Fertigung wird in den Bereichen

- des Rapid Prototyping,
- des Rapid Tooling und
- der Rapid Production

eingesetzt (Feldhusen & Grote 2013: 77). Die additive Fertigung erzeugt Bauteile mit geringen geometrischen Einschränkungen, allerdings variieren die Bauteileigenschaften abhängig vom Werkstoff bei unterschiedlichen Verfahren und die Eigenschaften der Bauteile weichen von denen des Grundwerkstoffs teilweise deutlich ab. Auch gibt es spezielle Einschränkungen, z. B. auf Grund teilweise nötiger sog. Stützkonstruktionen oder anderer verfahrensspezifischer Besonderheiten, die eine besondere Berücksichtigung der additiven Fertigung in der Produktentwicklung erfordern (Feldhusen & Grote 2013: 97).

Das Smart Product als CPS ist ein Produkt aus den Domänen

- Maschinenbau,
- Elektrotechnik und
- Informationstechnologie (Hard- und Software).

Daher muss die Produktentwicklung den besonderen Anforderungen dieser Domänen genügen. Dies wird häufig durch Anwendung des sog. V-Modells realisiert (siehe z. B. in Seidler 2013: 91ff.) oder einer Variante dieses Modells zur Berücksichtigung des Systembetriebs und damit dem Übergang zu einer kontinuierlichen Betrachtung über Produktlebenszyklen hinweg (siehe Ahrens & Hieronymus 2015).

Neben den Anforderungen aus dem RAMI4.0 als Industrie 4.0-Komponente (Bitkom e.V. et al. 2015) und der IT-Security spielen die Anforderungen der Produktsicherheit (Safety) eine große Rolle. Hierzu sind die geforderten Risikobetrachtungen durchzuführen und zu dokumentieren (Europäisches Parlament 2006). Da CPS als Smart Products kundenindividuelle Ausprägungen und aufgrund ihrer Vernetzungsfähigkeit in unterschiedlichen Kontexten verwendet werden, sind entsprechende Überlegungen bei der Risikobeurteilung zu berücksichtigen (Liggesmeyer et al. 2014: 433ff.).

5. PRODUKTENTWICKLUNGSPROZESSE FÜR KUNDENINDIVIDUALISIERTE PRODUKTE

Wie oben beschrieben können individuelle Produkte als kundenspezifische Sonderanfertigungen auftragsbezogen wie z. B. im Anlagenbau hergestellt werden. Entsprechende Prozessbeschreibungen liegen vor, allerdings lassen sich diese Lösungen nicht auf die Serienproduktion oder gar auf die kundenindividuelle Produktion im Sinne der Vision Industrie 4.0 übertragen, da diese Prozesse keine marktkonformen Ergebnisse für z. B. Konsumprodukte liefern können (Bauernhansl et al. 2014; Bitkom e.V. et al. 2015).

Individualisierte Produkte können durch Erhöhung der äußeren Varianz bei optimaler innerer Varianz erzeugt werden (Komplexitätserhöhung, s. o.). Die hier verwendeten Prozesse sehen ungeplante, kundenindividuelle Konstellationen nicht vor und sind für eine solche kundenspezifische Produktentwicklung schlecht geeignet (Lindemann et al. 2006).

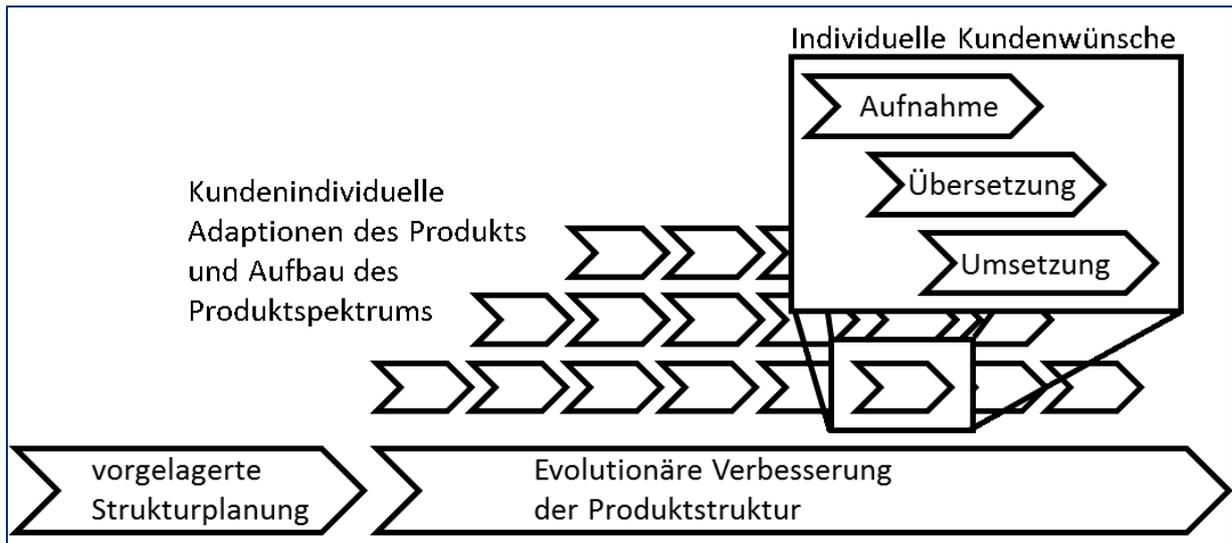


Abbildung 3: Kundenspezifische Produktentwicklung nach (Lindemann et al. 2006)

Lindemann schlägt eine Kombination aus strukturellen Maßnahmen zur Einschränkung kundenindividueller Anteile mit komplexitätsbeherrschenden Maßnahmen vor. Dabei wird in den Entwicklungsprozess die direkte Kundeninteraktion mit aufgenommen.

In der vorgelagerten Strukturplanung werden die grundlegenden Produkteigenschaften beschrieben und die kundenindividuellen Anteile definiert. Im Laufe des Produktlebens wird durch Aufnahme der individuellen Kundenwünsche in die Produktstruktur die Kundenindividualität dargestellt. Parallel wird die entstehende Produktstruktur evolutionär verbessert. Vorteile dieses Vorgehens liegen in der sparsamen Verwendung von Entwicklungskapazitäten im vorgelagerten Entwicklungsprojekt, Nachteile liegen in der Notwendigkeit der kontinuierlichen aufwändigeren Serienbetreuung (oder Produktadaption) (Lindemann et al. 2006: 130).

Die Vision Industrie 4.0 geht von der durchgängigen Digitalisierung des Produktlebenszyklus aus. Diese Digitalisierung betrifft nicht nur die Vorgänge im Produkt als CPS, sondern auch seine Interaktion mit der Umwelt, und damit auch mit den CPPS der Smart Factory (s. o.). Es lassen sich dann sowohl die Nutzung durch den Kunden als auch die Produktionsprozesse modellhaft beschreiben. Unter Nutzung der dann notwendigen Semantik können weitgehend freie Kundenanwendungen im Rahmen der Beschreibungsmodelle erlaubt werden. Diese Anwendungen umfassen nicht nur die Funktionen des Produkts, sondern in weiten Bereichen durch Nutzung additiver Fertigung auch deren Gestalt. Ein Produktentwicklungsprozess für kundenindividuelle Produkte beinhaltet also

- eine vorgelagerte Strukturplanung (Lindemann et al. 2006),
- eine modellbasierte Systembeschreibung (VDI e.V. et al. 2015),
- eine gezielte Kundeninteraktion (Lindemann et al. 2006, Reichwald & Piller 2006),
- eine Berücksichtigung der Anforderungen der Smart Factory (Bauernhansl et al. 2014),
- die Einhaltung der gesetzlichen Rahmenbedingungen (Europäisches Parlament 2006) und
- eine Begleitung über den gesamten Produktlebenszyklus (Ehrlenspiel 2013).

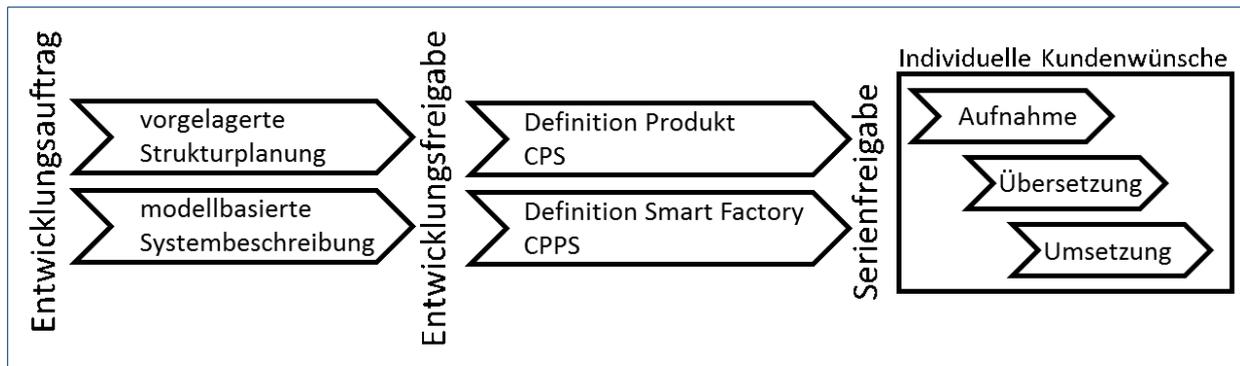


Abbildung 4: Entwicklungsprozess bei kundenindividuellen Produkten nach (Meussen 2015b)

Der wesentliche Aspekt der kundenspezifischen Produktentwicklung liegt in der Beschreibung der kundenindividuellen Aspekte eines Produkts. Nach der Industrie 4.0-Vision können diese individuellen Produkte als CPS in der Smart Factory von den CPPS erzeugt werden. Damit schwindet der Zwang zur Produktstandardisierung und es eröffnen sich neue Felder für Produktideen. Die Auswahl der vom Kunden individuell zu beeinflussenden Aspekte eines Produkts hat direkten Einfluss auf die Komplexität des Entwicklungsprozesses, da die modellbasierte Systembeschreibung entsprechend komplex ausfällt. Daher ist es für eine praktische Ausprägung des hier nur in Grundzügen dargestellten Entwicklungsprozesses erforderlich, konkrete Geschäftsmodelle umzusetzen und den Entwicklungsprozess auf die Erfordernisse des Geschäftsmodells zuzuschneiden (siehe z. B. Meussen 2013; Baumberger 2007: 5ff.).

An dieser Stelle unterliegt man dem Dilemma der gesamten Diskussion zu Industrie 4.0: Die Potentiale erscheinen verlockend, die Aufwände hoch und die konkrete Umsetzung noch ungewiss (Bitkom e.V. et al. 2015). Um nicht in diesem Dilemma gefangen zu bleiben, wird im Folgenden ein nutzenbasiertes Vorgehen vorgeschlagen, welches den Aspekt des kundenindividuellen Produkts berücksichtigt und daher die Möglichkeit geben soll, die beschriebenen Prozesse beispielhaft auszuprägen.

6. NUTZENBEISPIEL FÜR KUNDENINDIVIDUELLE PRODUKTENTWICKLUNG

Vorgeschlagen wird im Prinzip ein CPPS als flexible Fertigungszelle. Ausgehend von einem CAD-System werden automatisch Stücklisten aufgelöst, um aus einem automatischen Lager Norm- und Kleinteile zu erhalten. In einem weiteren Schritt werden Zeichnungsteile durch Verfahren der additiven Fertigung erstellt und durch kollaborative Roboter bereitgestellt bzw. vormontiert. Diese Konstellation stellt den Aspekt der Smart Factory dar, der in einem vorgegebenen Rahmen vollständig individuelle Produkte erlaubt (Böttcher et al. 2016).

Ausgehend von den Überlegungen zu Wertschöpfungsketten (VDI e.V. und VDE e.V. 2014) wird ein Nutzenbeispiel als Grundlage für weitere Betrachtungen zur nutzenorientierten Erforschung der Produktentwicklung kundenspezifischer Produkte erarbeitet (Böttcher et al. 2016).

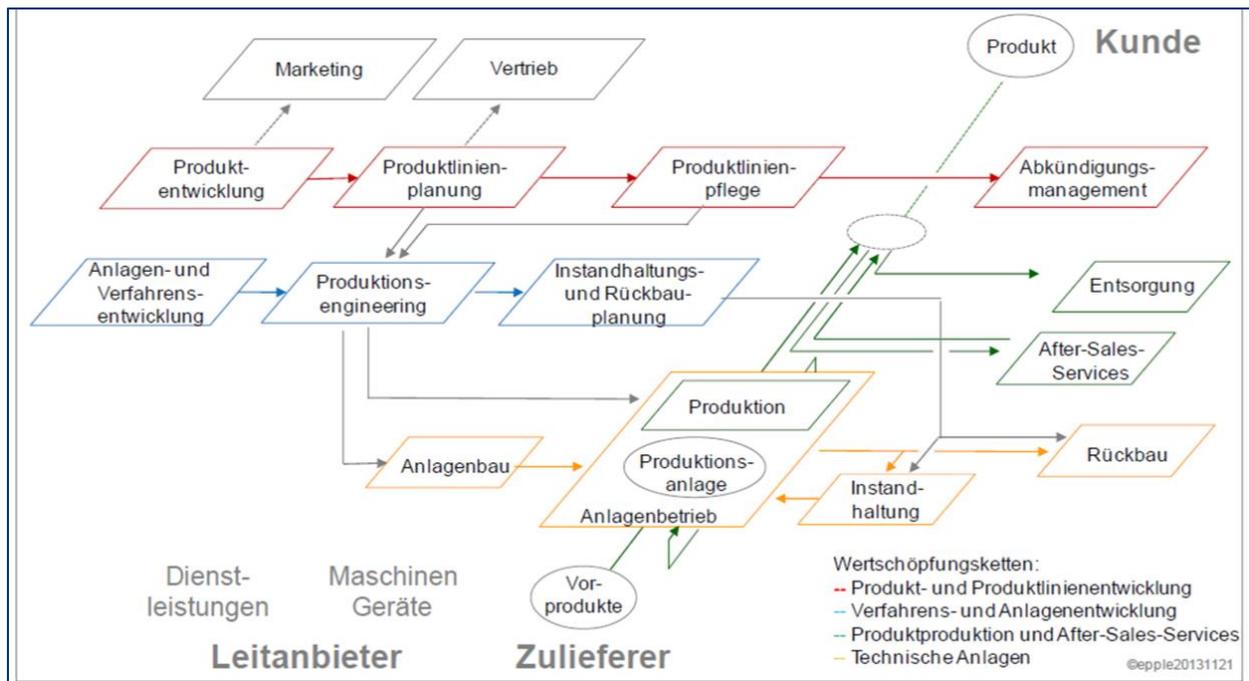


Abbildung 5: Wertschöpfungsketten der Industrie 4.0-Landschaft nach (VDI e.V. und VDE e.V. 2014)

Dabei wurden die wesentlichen Motivationen für eine kundenindividuelle Produktgestaltung untersucht:

- Biometrik,
- Geschmack und
- Funktion.

Mittels geeigneter Kreativitätstechniken wurde ein Geschäftsmodell entwickelt, welches alle drei Aspekte befriedigen kann und für das man auch einen Business Case aufstellen kann (Details hierzu in Böttcher et al. 2016). Hierbei geht es um elektrische Bedienelemente wie Lichtschalter und Netzsteckdosen, die folgende wesentliche Eigenschaften haben, um als Untersuchungsobjekt für kundenindividuelle Produktentwicklung geeignet zu sein:

- Verwendung von mechanischen Bauteilen als
 - Zeichnungsteile und
 - Standardbauteile
- Verwendung von elektronischen Komponenten und
- Erweiterungsmöglichkeit für Sensoren, Aktoren und eingebettete Systeme um zum CPS erweitert zu werden.

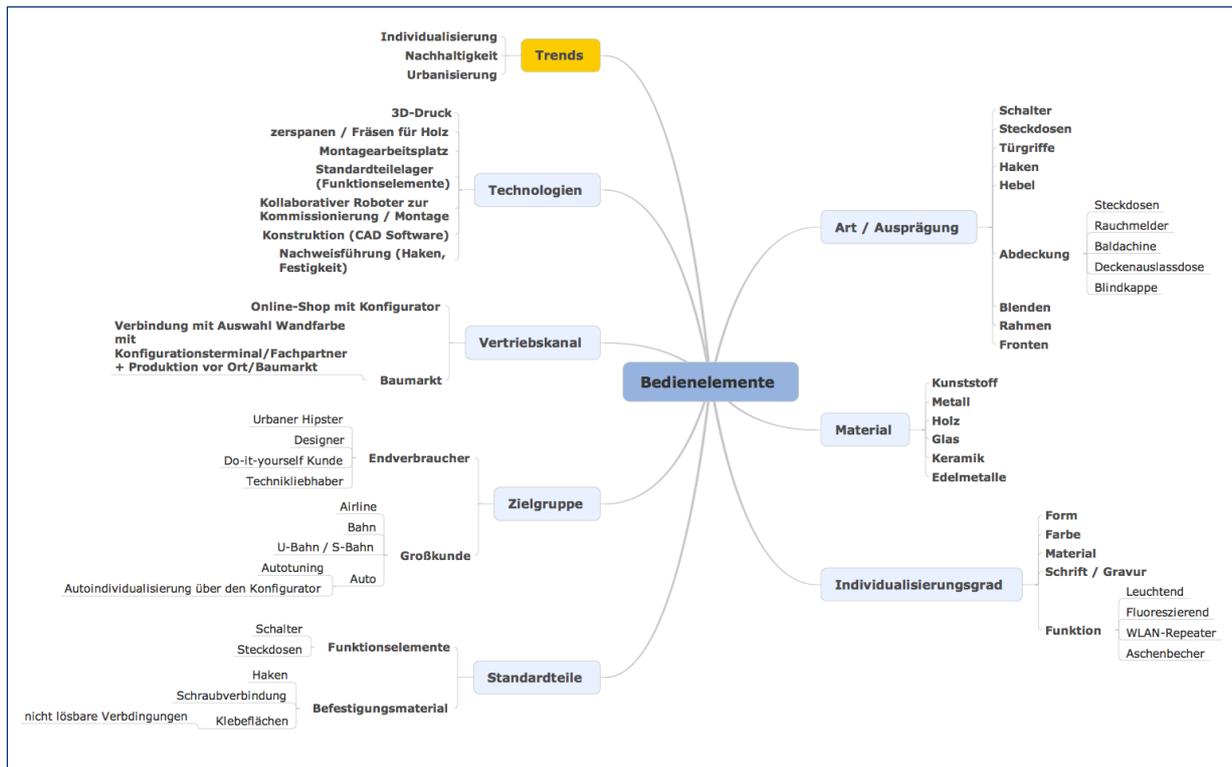


Abbildung 6: Mindmap zum Geschäftsmodell „Bedienelemente“ (Böttcher et al. 2016)

Als Beispiel für eine Individualisierung sei hier die Aufnahme des Gebäudedesigns in die Gestaltung der Lichtschalter oder die Abbildung eines Grundrisses auf dem Lichtschalter oder der Netzsteckdose gezeigt.



Abbildung 7: Schalter zur Wiedergabe einer typischen Gebäudegestalt (Böttcher et al. 2016)

7. BESCHREIBUNG EINES BEISPIELHAFTEN KUNDENINDIVIDUELLEN PRODUKTENTSTEHUNGSPROZESSES

In (Böttcher et al. 2016) wird der in Abbildung 8 gezeigte Prozessablauf zur Erstellung des kundenindividuelle Bedienelementes beschrieben.

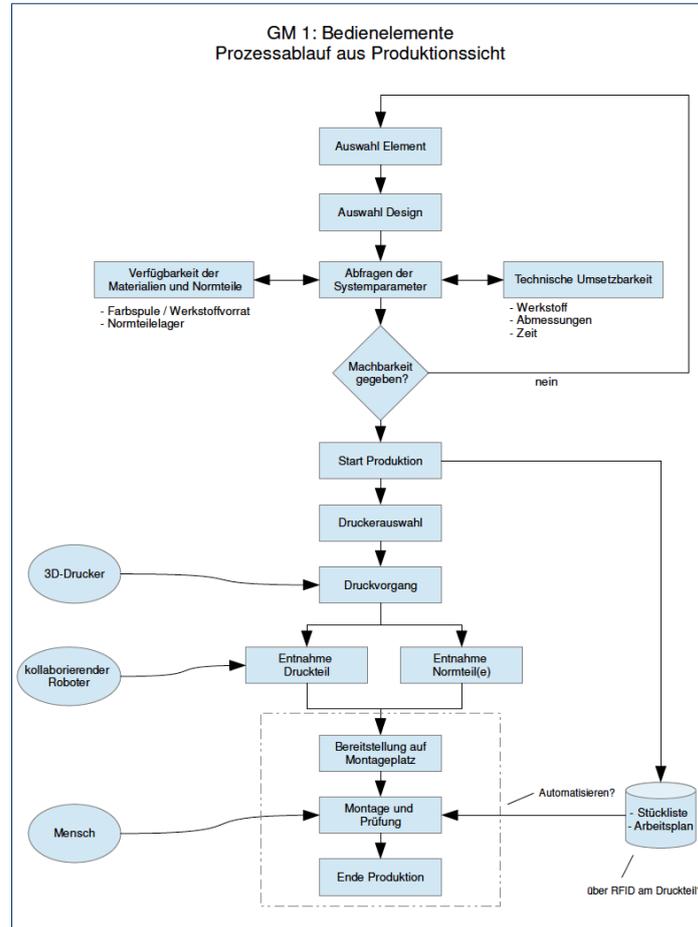


Abbildung 8: Prozessablauf bei der Erstellung eines kundenindividuellen Bedienelementes (Böttcher et al. 2016)

Zwei Aspekte der kundenindividuellen Produktentwicklung, die auch Lindemann et al. (2006: 7ff.) beschreibt, fallen auf:

- die Erfordernis der Prüfung der technischen Machbarkeit und
- die Möglichkeit der evolutionären Weiterentwicklung des Produkts durch Nutzung der kundenindividuellen Entwicklungsinhalte.

Letzteres kann mit den Ideen der Open Innovation (siehe Reichwald & Piller 2006) einhergehen. Von größerem Interesse sind hier die Erfordernisse, die in Abbildung 8 zur Prüfung der Machbarkeit herangezogen werden müssen.

Vor diesem Hintergrund wird ein Entwicklungsprozess vorgeschlagen, der die klassischen Entwicklungsprozesse des Maschinenbaus (siehe Ehrlenspiel & Meerkamm 2013 und Feldhusen & Grote 2013) mit den Überlegungen von Lindemann et al. (2006) und den Erfordernissen aus Böttcher et al. (2016) und Meussen (2013) verknüpft.

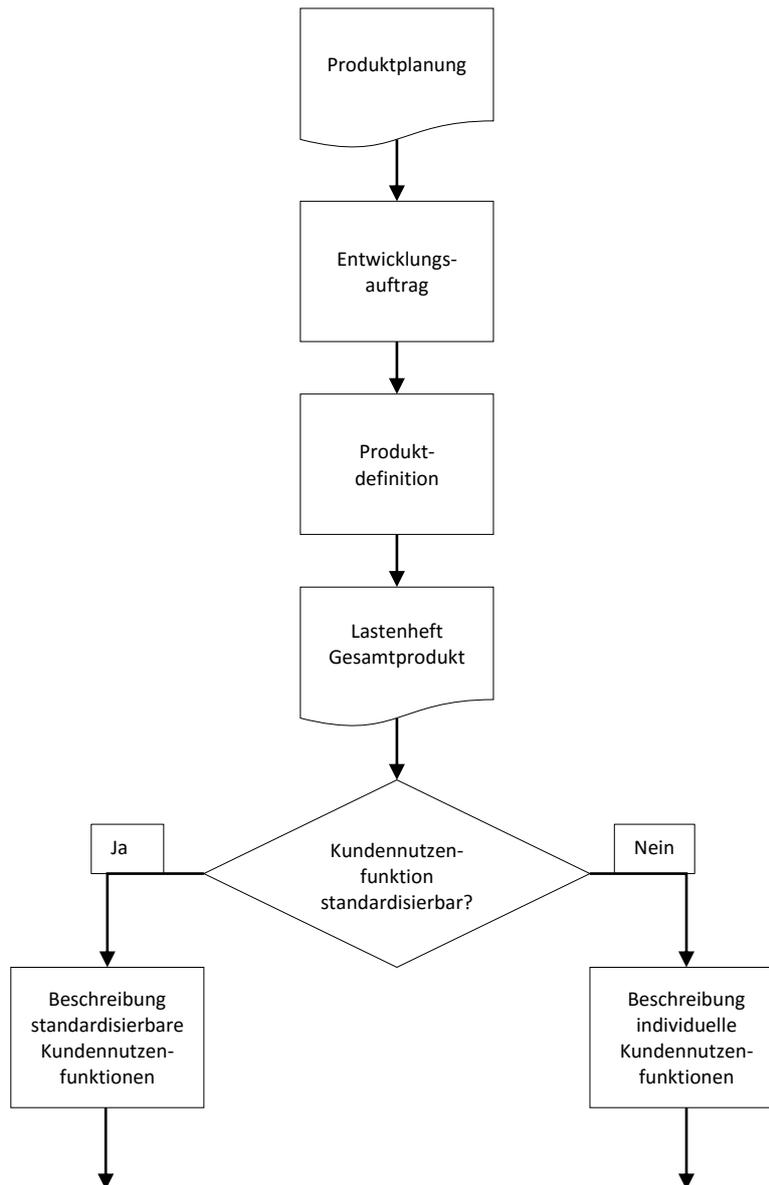


Abbildung 9: Produktentwicklungsprozess für kundenindividuelle Produktentwicklung (Teil 1)

Ausgehend von der revolvierenden *Produktplanung* (Abbildung 9) (siehe Ehrlenspiel & Meerkamm 2013) wird im Rahmen der Geschäftsstrategie der *Entwicklungsauftrag* für die Entwicklung des Produkts mit kundenindividuellen Entwicklungsinhalten gegeben. Dem klassischen Entwicklungsprozess folgend wird hierfür zunächst eine *Produktdefinition* in einem *Lastenheft Gesamtprodukt* erstellt, in dem alle relevanten Kundennutzenfunktionen beschrieben werden.

Im zweiten Schritt wird beurteilt, welche *Kundennutzenfunktionen* standardisierbar sind und welche nicht standardisierbar sind. An dieser Stelle teilt sich der Prozess in einen Pfad für die standardisierbaren Kundennutzenfunktionen und einen Pfad für die individuellen Kundennutzenfunktionen auf (Abbildung 10).

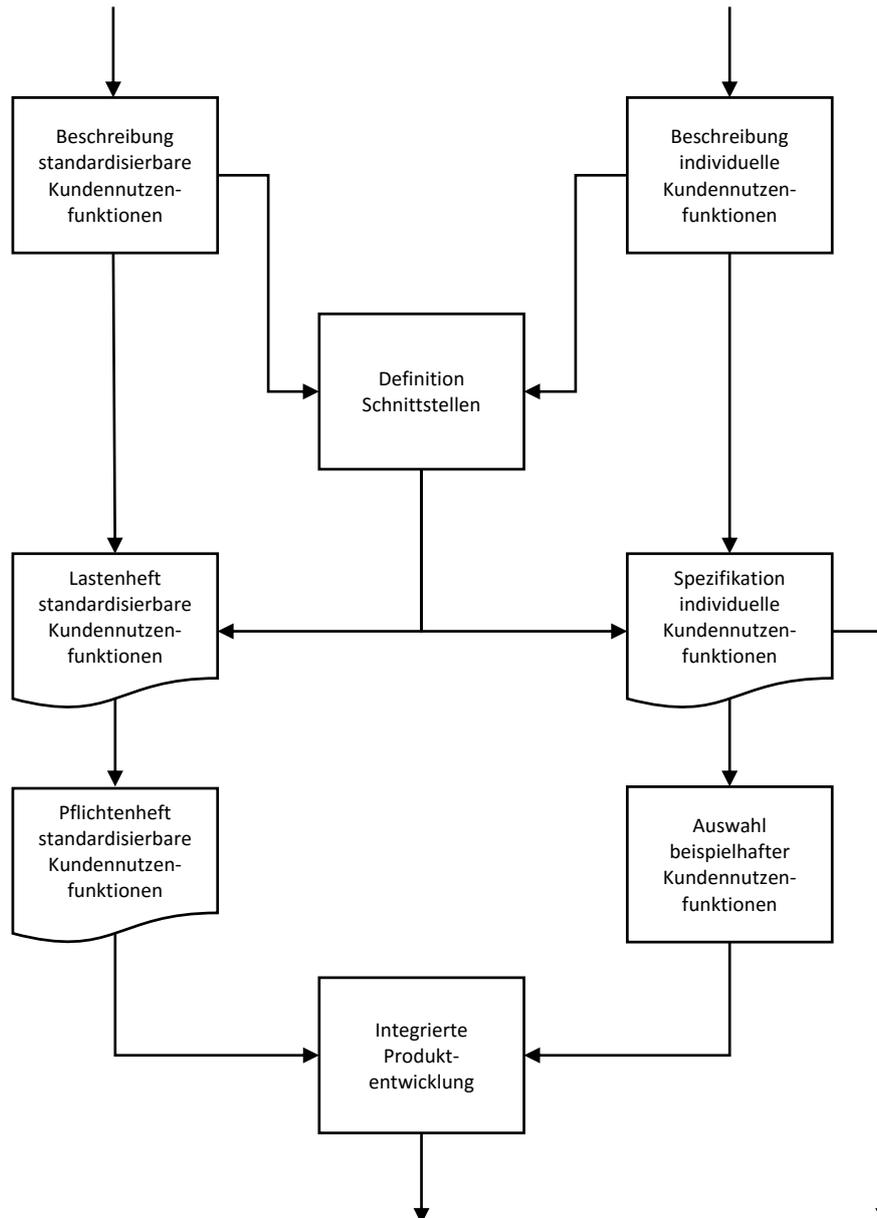


Abbildung 10: Produktentwicklungsprozess für kundenindividuelle Produktentwicklung (Teil 2)

Nach der Unterscheidung von standardisierbaren und individuellen Kundennutzenfunktionen müssen die *Schnittstellen* dieser Kundennutzenfunktionen untereinander festgelegt werden. Für die standardisierbaren Kundennutzenfunktionen können dann klassische *Lasten- und Pflichtenhefte* erstellt werden. Die individuellen Kundennutzenfunktionen werden in einer *Spezifikation* über ihre Eigenschaften beschrieben.

Aus dieser Spezifikation werden *beispielhafte individuelle Kundennutzenfunktionen* ausgewählt und als beispielhafter Anteil der jetzt weiter klassischen, z. B. *integrierten Produktentwicklung* nach (Ehrlenspiel & Meerkamm 2013) entwickelt.

Es folgt die Erstellung der *Prototypen* mit den beispielhaften kundenindividuellen Kundennutzenfunktionen sowie deren Nachweise (Abbildung 11). Im Rahmen der Produktentwicklung werden dann die Eigenschaften des Produktes sowie der erforderlichen Produktionsschritte modellhaft beschrieben. Diese Modelle dienen der Darstellung der Digitalisierung der Wertschöpfungskette Produktentwicklung und Produktion (siehe Bauernhansl et al. 2014 und VDI e.V. und VDE e.V. 2014 und VDI e.V. et al. 2015). Aus den Modellen werden für die Produktion der standardisierten und individuellen Produktion die Produktionsschritte und die Produktionsparameter ermittelt. Die Nullserie weist die Serienreife der Prozesse und der standardisierten und beispielhaft ausgewählten individuellen Anteile des Produkts und des Prozesses für die Serienfreigabe nach.

Mit dem Produktionsstart muss im Rahmen der Auftragsabwicklung die Entwicklung der kundenindividuellen Anteile erfolgen. Welche Entwicklungsschritte im Rahmen dieser Prozesse erforderlich sind, hängt von der Art der individuellen Kundennutzenfunktion ab. In Anlehnung an Lindemann et al. (2006) und Meussen (2013) sind diese Schritte im Rahmen der Definition der individuellen Kundennutzenfunktionen im Entwicklungsprozess im Rahmen der Spezifikation der individuellen Kundennutzenfunktionen festzulegen.

8. ZUSAMMENFASSUNG UND AUSBLICK

Die Entwicklung kundenspezifischer Produkte hat zum Ziel, für das anbietende Unternehmen einen Wettbewerbsvorteil durch bessere Anpassung des eigenen Produktes an die Kundennutzenfunktionen des Kunden zu erzielen. Dieser Wettbewerbsvorteil kann sich in höheren Preisen oder höheren Marktanteilen ausdrücken. Grundsätzlich gibt es zwei Möglichkeiten, die Produktindividualität zu erzeugen:

- Anwendung des sog. Mass Customization durch Berücksichtigung aller relevanten Kundenwünsche im Produktentwicklungsprozess durch Optimierung der inneren Produktvarianz für die erforderliche äußere Produktvarianz (siehe z. B. Reichwald & Piller 2006).
- Implementierung kundenspezifischer Produktentwicklungsschritte im Rahmen der Wertschöpfungskette Produktion zur Minimierung der im Vorwege im Rahmen der Produktentwicklung notwendigen Variantenbetrachtungen (siehe z. B. Lindemann et al. 2006).

Bisher war letzteres allein durch die notwendige Vorplanung von Produktionsschritten nicht möglich. Mit Hilfe der Vision Industrie 4.0 kann nun durch die Anwendung von CPPS und additiver Fertigung in der Smart Factory ein kundenindividuelles Produkt erzeugt werden, welches sich entsprechend den Produktionserfordernissen seinen Weg durch die Produktion selbst weisen kann.

Mit Hilfe des hier gemachten Vorschlags zur kundenspezifischen Produktentwicklung im Sinne von Lindemann et al. (2006) kann nun untersucht werden, welche Geschäftsmodelle im Rahmen welcher Wertschöpfungsketten Vorteile bieten kann.

Auf Basis des von Böttcher et al. (2016) gemachten Vorschlags für ein beispielhaftes Geschäftsmodell wird dieser Frage im Rahmen weiterer Forschungsarbeiten nachgegangen werden.

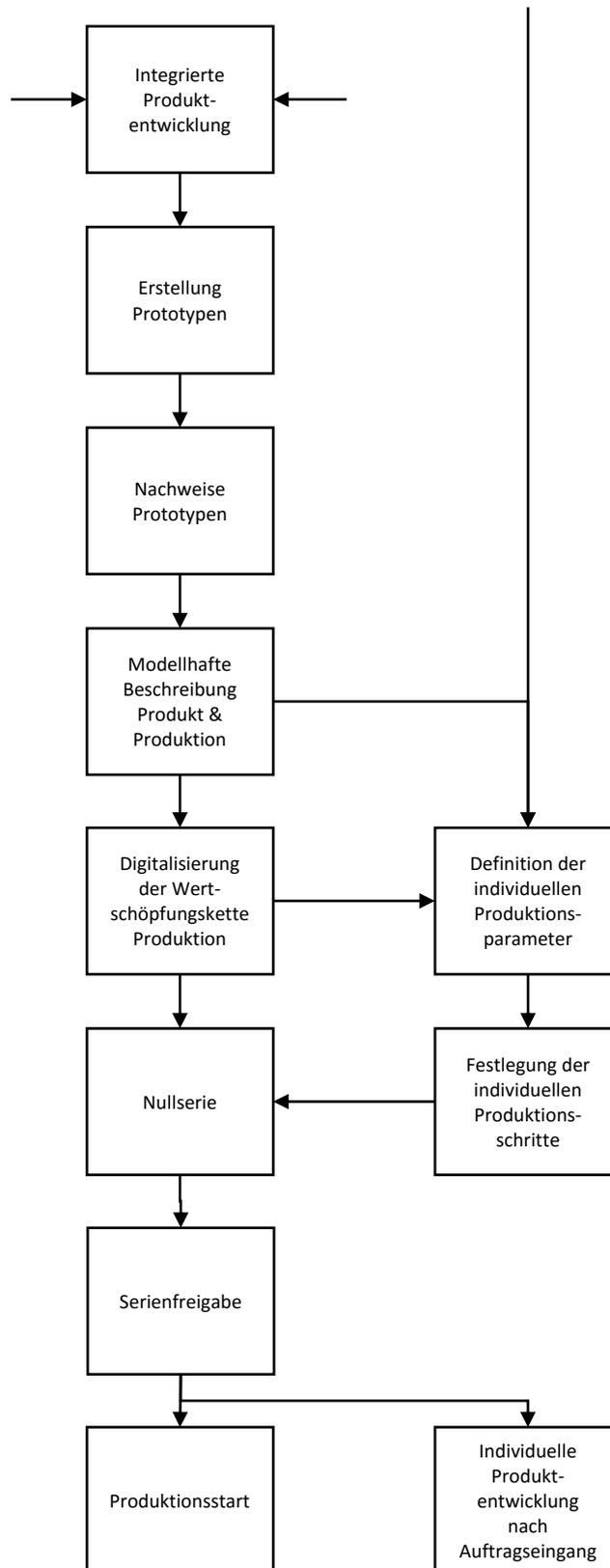


Abbildung 11: Produktentwicklungsprozess für kundenindividuelle Produktentwicklung (Teil 3)

9. QUELLENANGABEN

- Ahrens, V.; Hieronymus, M. (2015): Das Kreismodell als Grundlage der Entwicklungsmethodik für cyber-physikalische Systeme, in: Plate, G. (Hrsg.): Forschung für die Wirtschaft 2014, Göttingen, Cuvillier, S. 181 – 191.
- Baumberger, G. C. (2007): Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten, Dissertation, Technische Universität München.
- Bauernhansl, T.; Döbele, M.; Emmerich, V.; Paulus-Rohmer, D.; Schatz, A.; Weskamp, M. (2015): Geschäftsmodell-Innovation durch Industrie 4.0, Dr. Wieselhuber & Partner GmbH, München.
- Bauernhansl, T. (2014): Die vierte industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma, in: Bauernhansl, T.; ten-Hompel, M.; Vogel-Heuser, B. (Hrsg.) (2014): Industrie 4.0 in Produktion, Automatisierung und Logistik, Wiesbaden, Springer Fachmedien, S. 5 – 35.
- Bitkom e.V., VDMA e.V., ZVEI e.V. (Hrsg.) (2015): Umsetzungsstrategie Industrie 4.0 – Ergebnisbericht der Plattform Industrie 4.0, Bitkom e.V..
- Böttcher, S; Krone, E; Meussen, B.; Schnauer, M.; Rittemann, S. (2016): Die flexible Fertigungszelle als ein Cyber-Physikalisches Produktionssystem (CPPS), Arbeitspapiere der NORDAKADEMIE.
- Ehrlenspiel, K.; Meerkamm, H. (2013): Integrierte Produktentwicklung, München, Hanser.
- Europäisches Parlament (2006): Richtlinie 2006/42/EG des europäischen Parlaments und des Rats vom 17. Mai 2006. Amtsblatt der Europäischen Union, S. L157/24ff.
- Feldhusen, J.; Grote, K.-H. (Hrsg.) (2013): Konstruktionslehre, Berlin/Heidelberg, Springer Vieweg.
- Leichnitz, J.; Eilmus, S. (2013): Methoden für die Entwicklung modularer Produkte bei der Jungheinrich AG, Design for X, Beiträge zum 24. DfX-Symposium, Hamburg, S. 153 – 161.
- Liggesmeyer, P.; Trapp, M. (2014): Safety: Herausforderungen und Lösungsansätze, in: Bauernhansl, T.; ten-Hompel, M.; Vogel-Heuser, B. (Hrsg.) (2014): Industrie 4.0 in Produktion, Automatisierung und Logistik, Wiesbaden, Springer Fachmedien, S. 433 – 450 .
- Lindemann, U.; Reichwald, R.; Zäh, M. (Hrsg.) (2006): Individualisierte Produkte: Komplexität beherrschen in Entwicklung und Produktion, Berlin/Heidelberg, Springer.
- Meussen, B. (2013): Optimierung von Produktentwicklungsprozessen, in: Plate, G. (Hrsg.): Forschung für die Wirtschaft 2013, Göttingen, Cuvillier, S. 289 – 303.
- Meussen, B. (2015a): Anwendung von Industrie 4.0, in: Forschung und Praxis, Arbeitspapiere der NORDAKADEMIE.
- Meussen, B. (2015b): Ein Beitrag zur individuellen Produktentwicklung für die Industrie 4.0, Forschungsberichte der Hochschule Mittweida, IWKM 2015, S. 9 – 12.
- Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft (Hrsg.) (2013): Umsetzungsempfehlung für das Zukunftsprojekt Industrie 4.0, Deutsche Akademie der Technikwissenschaften e.V..
- Reichwald, R.; Piller, F. (2006): Interaktive Wertschöpfung: Open Innovation, Individualisierung und neue Formen der Arbeitsteilung, Wiesbaden, Gabler.
- Seidler, U. (Hrsg.) (2013): Industrie 4.0: Beherrschung der industriellen Komplexität mit SysLM, Berlin/Heidelberg, Springer Vieweg.
- Stegmüller, D.; Zürn, M. (2014): Wandlungsfähige Produktionssysteme für den Automobilbau der Zukunft, in: Bauernhansl, T.; ten-Hompel, M.; Vogel-Heuser, B. (Hrsg.) (2014): Industrie 4.0 in Produktion, Automatisierung und Logistik, Wiesbaden, Springer Fachmedien, S. 103 – 119.
- VDI e.V.; VDE e.V. (2014): Industrie 4.0 Statusreport: Wertschöpfungsketten, VDI e.V..
- VDI e.V.; VDE e.V.; ZVEI e.V. (2015): Statusreport: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), VDI e.V..
- Volkswagen AG (2012): Baukastenprinzip: Vielfalt durch einheitliche Standards, Viavision Nr. 2, März 2012.

MERKMALE VON OPEN-SOURCE-ENTWICKLUNGSPROZESSEN IN BIBLIOTHEKEN



Prof. Dr. Matthias Finck
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: Am Beispiel dreier großer Open-Source-Entwicklungsprojekte im Bereich webbasierter Bibliothekssoftware im deutschsprachigen Raum – der Software für Digitalisierungsprojekte goobi, dem offenen Discoverysystem vufind und der App für den Bibliothekskatalog BibApp – werden typische Merkmale von Open-Source-Entwicklungsprozessen in diesem Anwendungsfeld identifiziert und hinsichtlich ihrer Bedeutung für das Bibliotheksumfeld bewertet. Es werden u. a. die Strukturen der Entwicklungsgemeinschaften verglichen und die Erfolgsfaktoren benannt. Außerdem wird der Belgische Kreisel – eine Strategie aus dem Radsport – als Metapher für die prägnante Verbildlichung dieser typischen Merkmale vorgestellt.

Keywords: Open-Source, Webbasierte Systeme, Bibliotheken, Metaphern

1. EINLEITUNG

Hinter dem Begriff Open Source steht mehr als die Verfügbarmachung von quelloffener Software. Er steht für verschiedene mit dieser Quelloffenheit kausal verbundene Phänomene, die Auswirkungen auf die unterschiedlichsten Forschungs- und Wissenschaftsgebiete haben – wie z. B. die Rechtswissenschaft oder die Betriebswirtschaftslehre, aber auch Innovationsforschung oder Informationswissenschaften – und somit auch auf Bibliotheksprozesse.

1.1 Open-Source-Softwareentwicklungsprozesse

Open-Source-Entwicklungen werden häufig eine ganze Reihe von Eigenschaften zugesprochen, die bei einer näheren Betrachtung eher der Welt der Mythen und Missverständnisse zuzuordnen sind (Finck & Bleek 2006). Weder sind es im Normalfall weltweit kollektive und kooperative Entwicklungsprozesse, noch wird Open-Source-Software in der Regel von männlichen, jugendlichen Hackern entwickelt, die von altruistischen Motiven angetrieben werden.

Altruismus ist zwar ein identifizierbares Motiv, das aber in seiner Bedeutung überschätzt wird und mittlerweile deutlich hinter anderen Motiven wie z. B. Reputationsgewinn, Eigenbedarf oder finanziellen Motiven zurücksteht (vgl. u. a. Luthiger 2004; Lakhani & Wolf 2003; Mair et al. 2015). Außerdem umfassen die meisten Open-Source-Projekte weniger als zehn Entwickler, die häufig in einem engen lokalen Zusammenhang arbeiten (Finck 2007).

Die drei Merkmale, die die meisten Open-Source-Entwicklungsprozesse ausmachen und eindeutig beschreiben, sind die folgenden drei (vgl. Finck 2007: 73 ff):

Offenheit: Sowohl in der Nutzung als auch in der Entwicklung muss ein gewisses Maß an Offenheit vorhanden sein. Offenheit in der Nutzung besagt, dass die in der Entwicklung intendierte Nutzung nicht auf einen Kontext beschränkt ist und Offenheit in der Entwicklung bedeutet, dass es Dritten frei steht, sich in den Entwicklungsprozess einzubringen. Dabei bedeutet Offenheit in der Entwicklung nicht, dass einzelne im Entwicklungsprozess verankerte Projekte offen sind, aber es ermöglicht die grundsätzliche Beteiligung, sodass viele Projekte sich zwangsläufig stärker öffnen, um z. B. Parallelentwicklungen zu vermeiden.

Verteiltheit: Dieses Merkmal weist weniger auf einer örtliche (wohlmöglich weltweite) Verteiltheit der beteiligten Personen hin, sondern vielmehr auf eine zeitliche und vor allem organisatorische Verteiltheit. Typisch für Open-Source-Entwicklungsprozesse sind parallel oder synchron verlaufende Entwicklungsaktivitäten im Rahmen konkreter Projekte unterschiedlicher, am Entwicklungsprozess beteiligter Organisationen, die dann in Einklang gebracht werden müssen.

Agilität: Open-Source-Entwicklungsprozesse werden oft mit agilen Prozessen verglichen (vgl. Angioni et al. 2005; Gandomani et al. 2013). Flache Teamstrukturen, recht klein geschnittenen Aufgaben, gepaart mit einer „ad-hoc“-Verteilung, und relativ kurze Releasezyklen sind typische Eigenschaften von Open-Source-Entwicklungsprozessen und sprechen für eine Einordnung in agile Entwicklungsprozesse. Die durchaus schwankende Teamgröße und die Verteiltheit sprechen grundsätzlich zwar eher dagegen – insgesamt beschreibt der Agilitätsbegriff die Form der Zusammenarbeit recht passend, sodass er als identifizierendes Merkmal durchaus tauglich ist.

1.2 Bedeutung von Open-Source-Entwicklungen in Bibliotheken

Auch wenn im deutschsprachigen Bibliothekssektor Open-Source-Systeme noch nicht flächendeckend zum Einsatz kommen (Bauknecht 2009), so kann der Einsatz von Open-Source-Software mittlerweile „auch im bibliotheksspezifischen Bereich auf namhafte und erfolgreiche Projekte zurückblicken“ (Lohmeier & Seige 2014, Folie 7). Die Häufigkeit und Intensität, mit der das Thema auf bedeutenden Bibliothekskongressen aufgegriffen und diskutiert wird, zeigt, dass sich sehr intensiv mit dem Thema auseinandergesetzt wird (vgl. Mittelbach 2011). So wurden z. B. allein auf dem Deutschen Bibliothekartag 2015¹ fünf Vorträge mit Open Source im Vortragstitel gehalten..

Open-Source-Software wird als wichtiger Baustein für „Innovation und Wettbewerbsfähigkeit im Bibliotheksmarkt“ angeführt (Lohmeier & Seige 2014, Folie 3). Als Gründe für den Einsatz und die Entwicklung von Open-Source-Software in Bibliotheken werden vor allem die Offenheit, die Integrationsfähigkeit in einer sehr heterogenen Systemlandschaft und die Flexibilität in der schnellen Umsetzung innovativer Bibliothekslösungen angeführt (Flimm 2007; Lohmeier & Seige 2014).

Grundsätzlich scheinen dabei Open-Source-Entwicklungsprozesse an Bibliotheken gut vergleichbar mit bekannten, sehr stark Community-geprägten Open-Source-Projekten wie Ubuntu oder TYPO3 zu sein (vgl. Meier 2012). Es lassen sich also dieselben typischen Merkmale von Offenheit im Prozess sowie Verteiltheit und Agilität in der Entwicklung identifizieren.

Anhand dreier konkreter Fallstudien wird nachfolgend betrachtet, in welcher Ausprägung diese Merkmale in typischen bibliothekarischen Entwicklungsprozessen vorhanden sind und welche Auswirkungen diese Merkmale auf die konkreten Entwicklungsprojekte haben.

¹ http://react-profile.org/ebook/DBT2015_Hauptprogramm/

2. DREI FALLSTUDIEN

2.1 BibApp: Der Bibliothekskatalog als App

Die Universitätsbibliotheken Hildesheim und Lüneburg haben ab 2013 in einem gemeinsamen Projekt eine App für mobile Endgeräte entwickelt, die die Recherche und Ausleihe aus den Bibliothekskatalogen umsetzt (vgl. Goltz 2014). Die App wurde als Open-Source-Software geplant und nach der initialen Projektphase aktiv anderen Bibliotheken zur Nachnutzung zur Verfügung gestellt. Mittlerweile nutzen ca. 15 Bibliotheken die App nach.

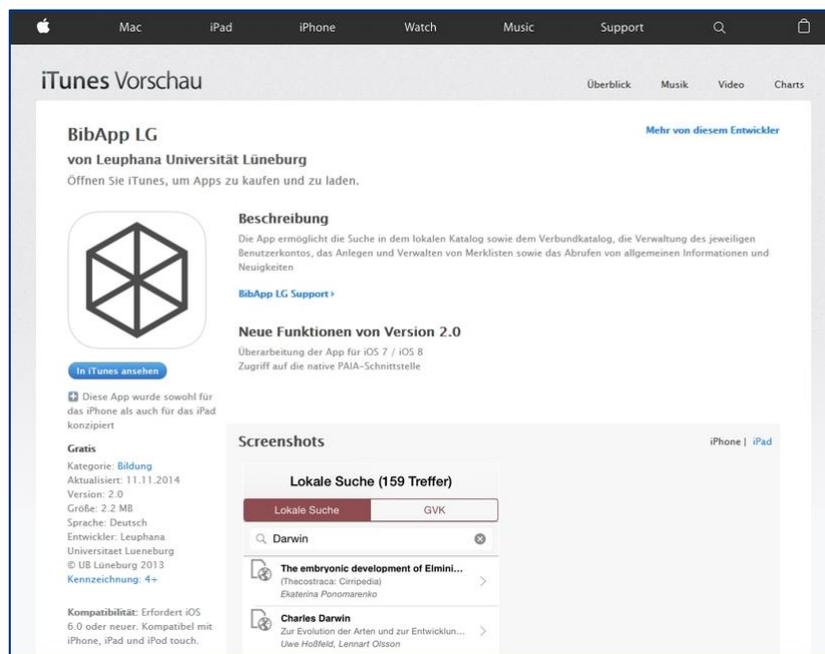


Abbildung 1: Screenshot der BibApp der Universität Lüneburg im Apple Store

Als Erfolgsfaktoren für die schnelle Verbreitung der App haben sich die Wahl einer Open-Source-Lizenz zur kostenfreien Nachnutzung und die Etablierung eines offenen Entwicklungsprozesses nach Veröffentlichung der ersten Version erwiesen.

Um eine lebendige Anwender-Community zu etablieren, wurde das Projekt über das Wiki des gemeinsamen Bibliotheksverbundes publiziert und so den weiteren Bibliotheken vorgestellt². Über diese Website erfolgt auch der konkrete Aufruf zur Nachnutzung und zur aktiven Mitarbeit am Entwicklungsprozess. Neben einem Mailverteiler zur Kommunikationsunterstützung werden jährliche Anwendertreffen organisiert und alle Protokolle dieser Treffen sowie alle Entwicklungswünsche werden über das Wiki transparent zur Verfügung gestellt.

Eine Besonderheit dieses Entwicklungsprozesses ist das Finanzierungsmodell. Die App ist strategisch so konzipiert worden, dass für notwendige Anpassungsarbeiten zur Nachnutzung relativ geringe initiale Kosten von ca. 3.000 € – 5.000 € entstehen. Außerdem gibt es eine verbindliche Zusage eines technischen Dienstleisters, die Anpassungen zu diesen Konditionen durchzuführen. Diese Kombination von geringen Kosten und einfacher Umsetzung hat wesentlich dazu beigetragen, dass die BibApp in relativ kurzer Zeit so zahlreich nachgenutzt wurde.

² <https://www.gbv.de/wikis/cls/BibApp>

Außerdem haben sich alle nachnutzenden Bibliotheken einverstanden erklärt, dass sie 500 € pro Jahr in einen gemeinsamen Topf zur Finanzierung der grundlegenden Betreuungs- und Entwicklungsaufgaben bereitstellen. Durch diese Crowdfunding ähnliche Form der Mikrofinanzierung des Entwicklungsprozesses (vgl. Hemer 2011) steht ein auskömmliches Grundbudget zur weiteren Pflege der BibApp zur Verfügung, ohne dass die Kosten für den einzelnen Standort zu hoch werden.

2.2 vufind: Ein quelloffenes Discovery-System

Im Rahmen eines gemeinsamen Projektes von sechs wissenschaftlichen Bibliotheken in Hamburg wurde seit 2007 unter Federführung der Staats- und Universitätsbibliothek Hamburg am Aufbau einer neuen Rechercheplattform mit dem Titel beluga gearbeitet (Christensen 2010: 320). Mit beluga entstand dabei eine mit Web 2.0-typischen Funktionen ausgestattete Rechercheplattform, die im Vergleich zum herkömmlichen Bibliothekskatalog die Recherche in über 8 Millionen Büchern, Zeitschriften und Aufsätzen in den Hamburger Bibliotheken vereinfachen und zudem die virtuellen Lernräume und sozialen Netzwerke mit Literaturinformationen und digitalen Texten versorgen sollte (Christensen 2010: 320).

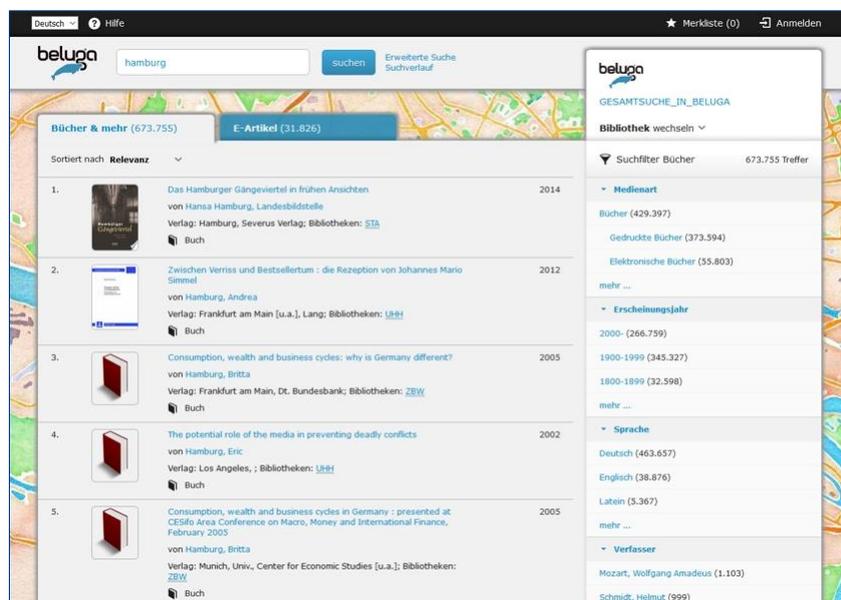


Abbildung 2: Screenshot der vufind-Installation „beluga“ der SUB Hamburg

Schon beim Bau des ersten Prototypen entschied sich die SUB Hamburg aus strategischen Gründen, auf die Offenheit des Systems, die Flexibilität und Agilität des Entwicklungsprozesses sowie die Nachnutzbarkeit für die Verwendung von Open-Source-Technologien (Christensen 2010) zu setzen. Seit 2011 kommt die Open-Source-Software vufind als technische Grundlage für beluga zum Einsatz, weil vufind sich als Open-Source-Alternative zu den proprietären Katalogsystemen international und auch deutschlandweit etabliert hatte (vgl. Breeding 2009; Schmidt & Stehle 2010).

Seit 2015 wird auch in diesem Entwicklungsprozess ein Konzept verfolgt, das darauf abzielt, möglichst vielen Bibliotheken eine niedrigschwellige Nachnutzung zu ermöglichen. Unter dem Arbeitstitel belugaCore wurde von der Staats- und Universitätsbibliothek Hamburg in Zusammenarbeit mit den Universitätsbibliotheken in Lüneburg, Hildesheim und Braunschweig ein technisches Grundgerüst von beluga entwickelt, welches möglichst aufwandsarm durch andere Bibliotheken nachnutzbar ist.

Es wird dabei bewusst nicht auf vufind selbst, sondern auf der weiterentwickelten beluga-Version aufgesetzt, weil diese den Anforderungen wissenschaftlicher Bibliotheken aus Deutschland deutlich besser entspricht.

Wie im BibApp-Projekt ist es so den nachnutzenden Bibliotheken möglich, das System mit vergleichsweise geringem finanziellen Aufwand nachzunutzen und sich gleichzeitig aktiv in die Entwicklung einzubringen.

2.3 Goobi: Eine Open-Source-Software für Digitalisierungsprojekte

Goobi ist eine Software-Lösung zur Produktion und Präsentation von Digitalisaten. Sie wird federführend von Bibliotheken selbst entwickelt und eignet sich besonders gut für die arbeitsteilige Massendigitalisierung (vgl. Bonte 2012). Die Software wird mittlerweile von zahlreichen Bibliotheken nachgenutzt – insgesamt in über 40 Bibliotheken, Archiven und Museen – in Deutschland, Großbritannien, den Niederlanden, Österreich und Spanien.



Abbildung 3: Screenshot der Website der Goobi-Anwendervereins Goobi e.V.

Als Gründe für die Entscheidung, Digitalisierungsprojekte mit Goobi durchzuführen, werden meist, neben der technischen Offenheit und Anpassbarkeit, eben auch die Lizenzkostenfreiheit und die Möglichkeit, sich aktiv in die Entwicklung einzubringen, genannt (vgl. z.B. Neuböck 2014).

Im Gegensatz zu den beiden vorherigen Fallbeispielen hat sich die Organisation der Anwender- und Entwickler-Community rund um Goobi deutlich stärker professionalisiert. Im Interesse eines verbesserten Rechts- und Investitionsschutzes sowie einer höheren Organisationskraft der Anwendergemeinschaft haben 12 Bibliotheken und Unternehmen aus Deutschland und Österreich 2012 den Verein "Goobi. Digitalisieren im Verein" gegründet, der das Release-Management organisiert und die Kommunikation zwischen den einzelnen Mitgliedern koordiniert (vgl. Neuböck 2013: 184). Mittlerweile sind in dem Verein mehr als 30 Bibliotheken und professionelle Dienstleister organisiert, um den Entwicklungsprozess von Goobi voranzutreiben.

Wie in dem BibApp-Projekt leisten alle Anwender über einen Mitgliedsbeitrag im Verein ihren Beitrag zur Finanzierung der Weiterentwicklung. Darüber hinaus gibt es aber regelmäßig projektfinanzierte Initiativen einzelner Vereinsmitglieder, die so größere Entwicklungsarbeiten finanzieren, die letztlich auch wieder der Anwender-Community zur Verfügung stehen.

3. DER BELGISCHE KREISEL ALS METAPHER

3.1 Gemeinsame Merkmale

Die drei Fallbeispiele zeigen, dass sich im deutschsprachigen Bibliothekswesen Open-Source-Entwicklungsprozesse ganz unterschiedlicher Größe und Ausprägung erfolgreich etabliert haben. Diese reichen von sehr flach organisierten Anwendergemeinschaften mit dem Ziel der kostengünstigen Pflege und Entwicklung einer nachgenutzten Software (BibApp-Projekt) bis hin zu Projekten mit professionell organisierten Strukturen mit langfristigen Perspektiven und Plänen (Goobi-Projekt).

Angelehnt an den drei typischen Merkmalen von Open-Source-Projekten im Allgemeinen (Offenheit, Verteiltheit, Agilität) lassen sich diese Merkmale in Bezug auf die Bibliotheksentwicklungen weiter spezifizieren und ausdifferenzieren:

Offenheit für die Nachnutzung

Die Offenheit ist auch in diesen Entwicklungsprozessen ein wichtiges Merkmal. Allerdings findet sie sich in den Bibliotheksprojekten vor allem in einer speziellen Ausprägung wieder. Primär steht in allen drei Projekten der Wunsch im Vordergrund, die zunächst selbst entwickelte Software anderen Bibliotheken zur Nachnutzung zur Verfügung zu stellen. Das lässt sich auch daran erkennen, dass die Organisation einer Anwender-Community in allen Projekten im Mittelpunkt der Aktivitäten steht und sich auch in allen Fällen gut etabliert hat.

Die Unterstützung einer echten Entwickler-Community wird zwar auch mehr oder weniger gefördert, aber dies steht offensichtlich nicht im Vordergrund. Im BibApp-Projekt, wie auch bei der Nachnutzung von beluga, erfolgt die Entwicklung fast vollständig durch eine Firma. Von den beteiligten Anwendern wird das nicht als ein so großer Nachteil empfunden, dass Maßnahmen zur Verbreiterung der Entwicklungsunterstützung eingeleitet wurden. Nur in der Goobi-Entwicklung existieren deutlich mehr direkt an der Entwicklung Beteiligte.

Organisatorische Verteiltheit

Noch weniger als bei typischen Open-Source-Entwicklungen steht bei den betrachteten bibliothekarischen Entwicklungsprozessen die örtliche oder zeitliche Verteiltheit im Vordergrund. Die eingesetzten Maßnahmen – wie jährliche Treffen oder Mailverteiler – dienen der Unterstützung organisatorischer Verteiltheit. Alle Projekte zeichnen sich durch eine Vielzahl von beteiligten Organisationen aus – vorwiegend Bibliotheken und Firmen als Entwicklungspartner.

Finanzielle Flexibilität

Flexibilität ist ein wesentlicher Faktor in den Fallbeispielen. Allerdings nicht in Bezug auf die Prozessorganisation. Flexibilität im Sinne agiler Softwareentwicklung steht bei den Bibliotheksentwicklungen nicht im Vordergrund. Die Entwicklungshierarchien sind zwar flach, aber die Entwicklungszyklen sind z. B. in allen Fällen deutlich zu lang, um wirklich als agil bezeichnet zu werden.

Die Flexibilität bezieht sich vielmehr auf die Entwicklungskosten als auf die Entwicklungsdurchführung. Ein entscheidender Faktor für die Beteiligung der nachnutzenden Bibliotheken an dem Entwicklungsprozess ist die hohe finanzielle Flexibilität. Die erstmalige Nachnutzung ist mit deutlich niedrigeren Fixkosten verbunden als die Anschaffung einer proprietären Alternative und bei der Entwicklung selbst bringt jeder Partner, neben einem sehr niedrigen Jahresbeitrag, jeweils so viel in den Prozess ein, wie ihm zur Verfügung steht.

3.2 Der Belgische Kreisel als Metapher für die Entwicklungsprozesse

Um die Merkmale und Besonderheiten der bibliothekarischen Entwicklungsprozesse möglichst anschaulich zu beschreiben, kann man sich einer Metapher bedienen. Die Nutzung von Metaphern ist in der Informatik aufgrund der Immaterialität des Gegenstandsbereichs ein etabliertes Mittel, um durch den Vergleich zwischen einem wenig und einem gut bekannten Sachverhalt ein besseres Verständnis des unbekannteren Sachverhalts aufzubauen (vgl. Madsen 1994; Preim 1999).

Um die Merkmale von Open-Source-Entwicklungsprozessen in Bibliotheken möglichst bildlich zu beschreiben, eignet sich aus meiner Sicht eine Metapher aus der Welt des Radrennsports: der Belgische Kreisel.

Radrennfahrer fahren i. d. R. in Gruppen, um Energie zu sparen. Ein Radrennfahrer, der hinter dem führenden Fahrer fährt, benötigt für dieselbe Geschwindigkeit nur 64% der Leistung des Führenden (Bock & Bracke 2014: 1). Als Team wird dann die beste Rennzeit erreicht, wenn der Führende regelmäßig seine Position verlässt und sich in einer in Reihe fahrender Rennfahrer am Ende wieder einsortiert. Dies ist das Prinzip des Belgischen Kreisels (Bock & Bracke 2014). Im klassischen Belgischen Kreisel wechselt die Führung immer nach gleicher Zeit, sodass die Positionen durchgehend wechseln. In heutigen Radrennen hat sich die Abwandlung als besonders erfolgreich durchgesetzt, in der jeder Führende solange in dieser Position verbleibt, wie es ihm nach individueller Leistungsstärke möglich ist. So kann sich jedes Teammitglied entsprechend seiner Leistungsfähigkeit unterschiedlich stark einbringen und damit die optimale Gesamtleistung erbracht werden.



Abbildung 4: Das Klapprad-Power-Team beim Belgischen Kreisel³

Diese Ausprägung der verbreiteten Radrennstrategie eignet sich gut als Metapher, um die Open-Source-Entwicklungsprozesse in Bibliotheken zu beschreiben:

Mit dem Ziel, möglichst ressourcenschonend notwendige Software zu entwickeln, tun sich Bibliotheken zu einer „Renngruppe“ zusammen. Während sich ein oder zwei Bibliotheken mittels einer Anschubfinanzierung vor die Gruppe setzen, können die anderen Bibliotheken „im Windschatten“ mithalten. Durch die Crowdfunding-ähnlichen Finanzierungsstrategien (z. B. Mitgliedsbeiträge oder freiwillige jährliche Zahlungen) behält die Gruppe eine konstante „Grundgeschwindigkeit“ bzw. Grundfinanzierung.

³ <http://world-klapp.de/>

Wenn einer Bibliothek aus organisatorischer Notwendigkeit zur Softwareweiterentwicklung heraus oder aufgrund sich ergebender finanzieller Spielräume – z. B. durch Drittmittelprojekte – vorübergehend mehr Ressourcen zur Verfügung stehen, dann kann diese für diese Zeit in die Führungsposition gehen und so zur höheren Gesamtleistung der Gesamtgruppe beitragen. Diese flexible Form der wechselnden Führung je nach finanziellen Ressourcen passt sehr gut zu der identifizierten finanziellen Flexibilität der Entwicklungsprozesse.

Weitere Bibliotheken können – wie im Radrennen – dynamisch die Gruppe erweitern oder auch wieder verlassen. Das passt gut zu der Offenheit für die Nachnutzung der Entwicklungsprozesse.

Die größte Herausforderung beim Belgischen Kreisel ist, sich unter den Fahrern abzustimmen, wer wann für wie lange in Führungsposition geht. Außerdem sollten keine Fahrer durch eine zu hohe Grundgeschwindigkeit abgehängt werden, weil dies zwar kurzfristig zu einer höheren Geschwindigkeit, über das gesamte Rennen aber zu einer geringen Leistung führt, weil im späteren Rennverlauf die unterstützenden Kräfte fehlen. Genau dieser Herausforderung müssen sich Bibliotheken in den gemeinsamen Entwicklungsprozessen auch stellen. Es gilt, alle mitzunehmen und keine in der Grundgeschwindigkeit zu überfordern, selbst wenn einzelne kurzfristig zu einer höheren Leistung in der Lage wären.

Im Belgischen Kreisel haben die Fahrer den großen Vorteil, die übrigen Teammitglieder in Hör- und Rufweite zu haben, sodass mögliche Probleme wie z. B. das Aufreißen einer Lücke unmittelbar identifiziert werden können. Die potentiellen Probleme in der Anwender-Community wie z. B. unterschiedliche Geschwindigkeiten in der Entwicklung oder Technologieaneignung möglichst zeitnah zu identifizieren, ist dabei eine typische Herausforderung, der in den Open-Source-Entwicklungsprozessen durch den Einsatz diverser Kommunikationskanäle (Treffen, Mailinglisten, Wikis, etc.) begegnet wird. Mit diesen Anforderungen an die Abstimmung werden typische Aspekte der organisatorischen Verteiltheit beschrieben.

4. FAZIT UND AUSBLICK

Open-Source-Entwicklungsprozesse in Bibliotheken als Radrennen zu verstehen, das sich mit der Strategie des Belgischen Kreisels am erfolgreichsten bewältigen lässt, hilft als Metapher, um plastisch zu beschreiben, welchen Herausforderungen sich Bibliotheken in diesen Prozessen stellen müssen – Offenheit in der Nachnutzung, organisatorische Verteiltheit und finanzielle Flexibilität.

Die drei Fallstudien zeigen, mit welchen Mitteln diesen Herausforderungen begegnet wird. Eine Grundfinanzierung durch Mitgliedsbeiträge, eine hohe kommunikative Abstimmung innerhalb der Anwender-Community und eine große Offenheit gegenüber nachnutzenden Bibliotheken sind typische Muster.

Die herausgearbeiteten Merkmale in dieser Arbeit können nun genutzt werden, um die in der Praxis angewendeten Methoden noch weiter zu systematisieren und zu schärfen. Außerdem sollten weitere Open-Source-Entwicklungsprozesse im Bibliothekswesen untersucht werden, um die Merkmale zu bestätigen.

5. LITERATUR

- Angioni, M.; Raffaella S.; Soro, A. (2005): Defining a distributed agile methodology for an open source scenario, in: Scotto, M., Succi, G. (Hrsg.): Proceedings of the 1st International Conference on Open Source Systems, July 11-15, Genova, Italy, Springer Verlag, S. 209 – 214.
- Bauknecht, C. (2009): Open Source Bibliothekssysteme: eine Betrachtung der Systeme BiblioteQ, Koha und OpenBiblio hinsichtlich Installationsaufwand und Leistungsfähigkeit. Bachelorarbeit. Hochschule der Medien Stuttgart, zugegriffen über: <http://hdms.bsz-bw.de/frontdoor/index/index/docId/606> am 30.01.2016.
- Bock, W.; Bracke, M. (2014): Teamcycling–Optimales Teamtraining im Radsport, in: Maas, J.; Siller, H.-J. (Hrsg.): Neue Materialien für einen realitätsbezogenen Mathematikunterricht, Wiesbaden, 2. Springer Fachmedien, S. 1 – 9.
- Bonte, A. (2012): Fünf Jahre Goobi – das Rückgrat der Massendigitalisierung der SLUB, in: BIS – Das Magazin der Bibliotheken in Sachsen. 5. Jg., Heft 2, S. 84 – 86.
- Breeding, M. (2009): Open Discovery Interface, in: American Libraries, 40 Jg., Heft 6/7, S. 40 – 41, zugegriffen über: <http://americanlibrariesmagazine.org/columns/open-discovery-interfaces> am 16.03.2016.
- Christensen, A. (2010): Katalog 2.0 im Eigenbau: Das beluga-Projekt der Hamburger Bibliotheken, in: Bergmann, J.; Danowski, P. (Hrsg.) Handbuch Bibliothek 2.0, De Gruyter Verlag, zugegriffen über: <http://www.degruyter.com/viewbooktoc/product/43790> am 30.01.2016.
- Finck, M.; Bleek, W.-G. (2006): Mythen, Märchen, Missverständnisse – Eine nüchterne informatische Betrachtung von Open-Source-Entwicklungsprozessen, in: Lutterbeck, B.; Bärwolff, M.; Gehring, R. A. (Hrsg.) Open Source Jahrbuch 2006 – Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin, Lehmanns Media, S. 207 – 219.
- Finck, M (2007): Usability-Engineering in der Open-Source-Softwareentwicklung – Perspektiven, Vorgehensweisen und Techniken, Dissertation am Department Informatik der Universität Hamburg, Göttingen, Sierke Verlag.
- Flimm, O. (2007). Die Open-Source-Software OpenBib an der USB Köln–Überblick und Entwicklungen in Richtung OPAC 2.0, in: Bibliothek Forschung und Praxis, 31. Jg., Heft 2, S. 185 – 192.
- Gandomani, T.; Zulzalil, H.; Ghani, A.; Sultan, A. (2012): A Systematic Literature Review on Relationship Between Agile Methods and Open Source Software Development Methodology, in: International Review on Computers and Software, 7. Jg., Heft 4, S. 1602 – 1607.
- Goltz, J. A. (2014): Mobile Applikationen für Bibliotheken im deutschsprachigen Raum, in: ZIB-Report 14-06, zugegriffen über: <https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/4693> am 16.03.2016.
- Hemer, J. (2011) Crowdfunding und andere Formen informeller Mikrofinanzierung in der Projekt- und Innovationsfinanzierung, Stuttgart, Fraunhofer Verlag.
- Lakhani, K., R.; Wolf, R. G. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects, in: Feller, J.; Fitzgerald, B.; Hissam, S. (Hrsg.): Perspectives on Free and Open Source Software, Boston, MIT Press, S. 3 – 23.
- Lohmeier, F., Seige, L. (2014): Open Source Bibliotheksmanagement. Vortrag auf der Konferenz der Leiterinnen und Leiter der sächsischen Hochschulbibliotheken, zugegriffen über: <http://de.slideshare.net/f.lohmeier/open-source-bibliotheksmanagement-mit-dswarm-amsl> am 27.01.2016
- Luthiger, B. (2004). Alles aus Spaß? Zur Motivation von Open-Source-Entwicklern, in: Lutterbeck und Gehring (Hrsg.): Open Source Jahrbuch 2004, Berlin, Lehmanns, S. 93 – 106.
- Madsen, K. (1994): A guide to metaphorical design, in: Communications of the ACM, 37. Jg., Heft 12, S. 57 – 62.
- Mair, P.; Hofmann, E.; Gruber, K.; Hatzinger, R.; Zeileis, A.; Hornik, K. (2015): Motivation, values, and work design as drivers of participation in the R open source project for statistical computing, in: PNAS - Proceedings of the National Academy of Sciences of the United States of America. 112. Jg., Heft 48, S. 14.788-14.792; vorveröffentlicht zum Print am 09.11.2015, doi:10.1073/pnas.1506047112.

- Meyer, S. (2010): Goobi.org. Die Informationsplattform zur Digitalisierung, in: BIS – Das Magazin der Bibliotheken in Sachsen, 3. Jg., Heft 1, S. 12 – 14.
- Meyer, S. (2012): Dos and Don'ts der kooperativen Software-Entwicklung: Das Goobi Release Management. Vortrag auf dem 101. Deutscher Bibliothekartag. Hamburg 2012, zugegriffen über: <https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/1097> am 27.01.2016.
- Mittelbach, A. (2011): Zur Zukunft von Bibliothekssoftware, in: BIS – Das Magazin der Bibliotheken in Sachsen, 4. Jg., Heft 1, S. 26 – 28.
- Neuböck, G (2013). Oberösterreichische Landesbibliothek: Neues Leben für alte Bücher–oder wie man ein Digitalisierungsprojekt auf Schiene bringt!, in: Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen & Bibliothekare, 66. Jg., 1. Heft, S. 179 – 191.
- Neuböck, G. (2014): David gegen Goliath–Digitalisierung in Regionalbibliotheken. Bibliothek Forschung und Praxis, 38. Jg, Heft 3, S. 439 – 443.
- Preim, B. (1999): Metaphern in der Mensch-Computer-Interaktion, in: Preim, B. (Hrsg.): Entwicklung interaktiver Systeme, Berlin/Heidelberg, Springer, S. 165 – 179.
- Schmitt, J. & Stehle, M. (2010). Der OPAC aus dem Baukasten. Realisierung eines Katalog 2.0 unter Einbeziehung der Community. Überarbeitete Version 1.0 (Bachelorthesis). Hamburg Hochschule für Angewandte Wissenschaften, Hamburg, zugegriffen über: <http://edoc.sub.uni-hamburg.de/haw/volltexte/2011/1143/> am 30.01.2016.

INDIKATORENBASIERTE AUDITS – EIN STRINGENTER ANSATZ ZUR PRÜFUNG VON GESCHÄFTSPROZESSEN?



StB Prof. Dr. Nick Gehrke
NORDAKADEMIE Hochschule der Wirtschaft, Elmshorn

Abstract: Die Prüfung von Geschäftsprozessen im Hinblick auf Ordnungsmäßigkeits- und Complianceaspekte ist ein verbreitetes Vorgehen in größeren Organisationen. In diesem Beitrag wird im Sinne der gestaltungsorientierten Wirtschaftsinformatik ein Vorgehen entwickelt, wie Prozessprüfungen methodisch unterstützt werden können, sodass eine Toolunterstützung weitestgehend ermöglicht wird. Das Vorgehen stützt sich auf die Daten im ERP-System des Unternehmens und geht in folgenden Schritten vor. Mittels Process Mining werden zunächst die relevanten Geschäftsprozessinstanzen rekonstruiert. In einem zweiten Schritt werden die Belege des ERP-Systems mit Indikatoren, die auf Schwächen oder Complianceverstöße hinweisen, markiert und mit den Geschäftsprozessinstanzen verknüpft. Zum Schluss werden alle Geschäftsprozessinstanzen und deren Verknüpfungen zu relevanten Indikatoren einer statistischen Faktorenanalyse unterzogen, um zu berechnen, welche systematischen Prozessschwächen, repräsentiert durch die Indikatoren, in den Geschäftsprozessen des gesamten Unternehmensdatensatzes verborgen sind. Der Prüfer kann dann einerseits die systematischen Prozessschwächen oder „Schwächeprofile“ erkennen und andererseits konkrete Geschäftsprozessinstanzen prüfen, die den ermittelten „Schwächeprofilen“ entsprechen.

Keywords: Auditing, Audit, Revision, Process Mining, Big Data, Enterprise Ressource Planning

1. EINLEITUNG UND MOTIVATION

Große Organisationen wickeln ihre Geschäftsprozesse oftmals in einem oder mehreren integrierten betriebswirtschaftlichen IT-Systemen (Enterprise Ressource Planning (ERP) Systeme) ab. Dabei werden in diesen ERP Systemen auch die resultierenden Buchungen im Rechnungswesen elektronisch erfasst, welche in ihrer Gesamtheit mindestens am Jahresende für Zwecke der Finanzberichtserstattung in Form des Jahresabschlusses für Investoren und Finanzbehörden aufbereitet werden. Der Jahresabschluss und damit alle im Betrachtungszeitraum erfassten Buchungen unterliegen gesetzlichen Prüfungspflichten (§316 HGB), um Investoren, Gläubiger und Behörden vor Falschdarstellungen der Ertrags-, Finanz- und Vermögenslage zu schützen. In großen und/oder transaktionsintensiven Organisationen kann der Prüfer aufgrund der Datenmasse nicht mehr sinnvoll einzelne Geschäftsvorfälle auf Stichprobenbasis prüfen, um die Ordnungsmäßigkeit des Jahresabschlusses festzustellen. Aufgrund dieses Umstandes schreiben internationale und nationale Prüfungsstandards eine systemische Prüfung vor, welche eine Prüfung der IT-Systeme, der Geschäftsprozesse und des sogenannten internen Kontrollsystems, welches die Geschäftsprozesse organisationsintern führt und kontrolliert, beinhaltet. Hierbei wird auf den Zusammenhang zurückgegriffen, dass wohlkontrollierte Geschäftsprozesse auch zu einer ordnungsmäßigen Abbildung der Geschäftsprozesse in der Finanzbuchhaltung führen. Im internationalen Prüfungswesen geben die International Standards on Auditing (ISA) die Richtlinien zur Prüfung des

StB Prof. Dr. Nick Gehrke ist seit 2011 Professor für Wirtschaftsinformatik und Leiter des Masterstudiengangs Financial Management and Accounting an der NORDAKADEMIE. Nach dem Studium der Betriebswirtschaft an der Universität Hamburg war er als wissenschaftlicher Mitarbeiter am Institut für Wirtschaftsinformatik an der Universität Göttingen tätig. Danach war er fünf Jahre bei der Wirtschaftsprüfungs- und Beratungsgesellschaft PwC beschäftigt. In dieser Zeit legte er das Steuerberaterexamen ab. Schwerpunkte seiner Forschungstätigkeit sind IT-Management und Datenbanken.

E-Mail: nick.gehrke@nordakademie.de

Rechnungswesens vor. ISA 315 fordert dabei eine Prüfung der IT und der Geschäftsprozesse: „The auditor should obtain an understanding of the information system, including the related business processes, relevant to financial reporting (...)“ ISA 315.81. Weiterhin ist eine solche Prüfung der IT und der Geschäftsprozesse jährlich wieder notwendig, soweit Änderungen wahrscheinlich sind: „when the auditor intends to use information about the entity and its environment obtained in prior periods, the auditor should determine whether changes have occurred that may affect the relevance of such information in the current audit“ ISA 315.12 (International Federation of Accountants (IFAC-I) 2008). Neben der üblichen Prüfung rechnungswesenrelevanter Geschäftsprozesse soll der Prüfer auch Prüfungshandlungen zur Entdeckung von Betrug und dolosen Handlungen (Fraud) durchführen (ISA 240). Besonders relevant in Bezug auf die Prüfung von Geschäftsprozessen und internen Kontrollsystemen sind dabei Unternehmen, die dem amerikanischen Sarbanes-Oxley-Act (PUBLIC LAW 107–204—JULY 30 2002) unterliegen. Dies können deutsche Firmen sein, die an der amerikanischen Wertpapierbörse gelistet sind. Gleichermaßen trifft dies aber auch amerikanische Firmen, die ein deutsches Tochterunternehmen betreiben. Insbesondere der Sarbanes-Oxley-Act Section 404 „Management Assessment of Internal Controls“ schreibt dabei explizit die Einrichtung und interne Prüfung eines internen Kontrollsystems entlang der Geschäftsprozesse vor.

Auch nationale Prüfungsstandards schreiben eine entsprechende Prozessprüfung vor. Maßgeblich sind hier die Prüfungsstandards des Instituts der Wirtschaftsprüfer (IDW). Die im HGB geforderten Prüfungen werden durch den Standard IDW RS FAIT 1 (IDW RS FAIT 1 2002) (IDW Stellungnahme zur Rechnungslegung: Grundsätze ordnungsmäßiger Buchführung bei Einsatz von Informationstechnologie) konkretisiert. Im Besonderen die Textziffern 107 und 108 weisen auf die Notwendigkeit von Prozessprüfungen hin. Weiterhin wird in den IDW Prüfungsstandards 261 (IDW PS 261 2006) (Feststellung und Beurteilung von Fehlerrisiken und Reaktionen des Abschlussprüfers auf die beurteilten Fehlerrisiken; Textziffer 37ff) durch „Der Abschlussprüfer hat sich bereits im Rahmen der Prüfungsplanung einen Überblick über das zu prüfende Unternehmen zu verschaffen. Dies umfasst u. a. den Umgang des Managements mit den Geschäftsrisiken und die Organisation der Geschäftsprozesse im Unternehmen und insbesondere den Überblick über das interne Kontrollsystem.“, sowie in IDW PS 330 (IDW PS 330 2002) (Abschlussprüfung bei Einsatz von Informationstechnologie; Textziffer 84ff) „Prüfung IT-gestützter Geschäftsprozesse“ eine Prozessprüfung als Teil der vorgeschriebenen Jahresabschlussprüfung deklariert.

Auch die interne Revision hat ein Interesse an Prozessprüfungen. Die Rolle der internen Revision wurde seit dem Bilanzrechtsmodernisierungsgesetz 2009 (BilMoG) weiter aufgewertet. So sieht § 107 (3) S. 2 AktG vor:

Er (=der Aufsichtsrat) kann insbesondere einen Prüfungsausschuss bestellen, der sich mit der Überwachung des Rechnungslegungsprozesses, der Wirksamkeit des internen Kontrollsystems, des Risikomanagementsystems und des internen Revisionssystems sowie der Abschlussprüfung, hier insbesondere der Unabhängigkeit des Abschlussprüfers und der vom Abschlussprüfer zusätzlich erbrachten Leistungen, befasst.

Der Gesetzgeber wollte dem Aufsichtsrat offenbar hiermit ein Instrument nahelegen, um der Überwachungsfunktion professionell nachkommen zu können. Diese interne Überwachung auf operativer Ebene fällt regelmäßig der internen Revisionsabteilung zu, die für die Prüfung von Rechnungslegungsprozessen, Kontrollsystemen und Risikomanagementsystemen somit auch auf entsprechende methodische und IT-technische Unterstützung angewiesen ist. Für Richtlinien innerhalb der internen Revision existieren Dokumente, die vom Berufsverband der internen Revisoren herausgegeben werden (international das „Institute of Internal Auditors (IIA)“; in Deutschland „Institut für interne Revision (IIR)“). Im Zusammenhang mit Geschäftsprozessen, welche von ERP-Systemen unterstützt werden, sind insbesondere die Global Technology Audit Guides (GTAGs) relevant. Insbesondere GTAG 8: „Auditing Application Controls“ (Bellino & Hunt 2007) und GTAG 13: „Fraud Prevention and Detection in an Automated World“ (Askelson et al. 2009) sind für eine Prozessprüfung überaus relevant. Sogenannte „Application Controls“ sind dabei Kontrollen im Rahmen des internen Kontrollsystems, welche in IT-Systemen eingebettet sind, Geschäftsprozesse führen und

Integrität bewahren. Hierzu gehören z. B. Passwortkontrollen, Zugriffsrechte, Definition von Mussfeldern bei der Dateneingabe, zwingende Referenz auf vorangegangene Geschäftsvorfälle (Bestellungen, Wareneingänge etc.), Vier-Augen-Prinzip bei der Freigabe von Stammdaten oder systemgestützte Genehmigungsschritte. Auch der wohl umfassendste und bekannte de-facto Standard für IT-Governance CobiT (Control Objectives for Information and Related Technology) hebt die Wichtigkeit von Application Controls bei automatisierten Geschäftsprozessen hervor und gibt umfangreiche Definitionen und Hinweise zur Implementierung und Prüfung von dieser Art von internen Kontrollen (ISACA 2009).

Das Ergebnis von Audits ist letztendlich das Ergebnis der Würdigung des Prüfers („Professional Judgement“) im Hinblick auf den untersuchten Prüfungsgegenstand und hat somit stets einen gewissen subjektiven Charakter. Dieses „Professional Judgement“ kann nicht automatisiert, sehr wohl aber durch geeignete Methoden und Tools unterstützt werden. Dieser Beitrag macht deshalb im Sinne einer gestaltungsorientierten Wirtschaftsinformatik einen Vorschlag (Design of Artifacts), wie eine solche methodische Unterstützung für Prozessprüfungen im Sinne einer „sinnhaften Vollautomation“ (Mertens 1995) aussehen könnte. Entwicklung sowie Evaluation der dafür notwendigen Artefakte findet dabei durch prototypische Implementierung bzw. Anwendung statistischer Methoden statt und soll somit strengen wissenschaftlichen Maßstäben standhalten. In einer Case Study anhand echter Unternehmensdaten werden die entwickelten Artefakte angewendet und somit einer ersten Evaluation unterzogen.

2. METHODISCHES VORGEHEN

Eine Prozessprüfung findet in der betriebswirtschaftlichen Praxis für gewöhnlich derart statt, dass der Prüfer sich zunächst Klarheit über den oder die Prozessabläufe verschafft (Schritt 1, Prozessverständnis). Dies geschieht durch die Inaugenscheinnahme vorhandener Prozessdokumentationen oder, soweit nicht vorhanden, durch eine eigene Prozessdokumentation, die durch vorangegangene Interviews erhoben wurde. In einem zweiten Schritt identifiziert der Prüfer in den Prozessen (prozessintegrierte Kontrollen) oder auch in der Aufbauorganisation (prozessunabhängige Kontrollen) interne Kontrollen, die die Ordnungsmäßigkeit der Prozessergebnisse sicherstellen sollen (Schritt 2, interne Kontrollen identifizieren). Die identifizierten internen Kontrollen werden dann auditiert im Hinblick auf das Design (Aufbauprüfung) und die tatsächliche Wirksamkeit (Funktionsprüfung) (Schritt 3, interne Kontrollen prüfen).

Denkt man über eine methodische Unterstützung dieses Prüfprozesses nach, um eine sinnhafte Vollautomatisierung zu ermöglichen, so stellt sich die Frage, wie sich die beschriebenen Schritte des Prüfers in eine automatisierbare Methodik „übersetzen“ lassen. Dies soll im Folgenden beschrieben werden.

Schritt 1: Prozessverständnis

Die Ermittlung der Abläufe in dem betrachteten Unternehmen geschieht vollständig aufgrund der tatsächlichen Datenlage (empirisch induktives Vorgehen). Alle Abläufe sind implizit im ERP-System des Unternehmens gespeichert. ERP-Systeme, im Besonderen auch SAP (welches für diese Studie betrachtet wurde), sind „Belegerzeugungssysteme“, was schlicht bedeutet, dass jeder relevante Prozessschritt einen Beleg im System hinterlässt (z. B. im Rechnungswesen, aber auch im Einkauf Bestellungen oder im Verkauf Aufträge und Lieferungen). Mit Methoden des Process Mining, hier aufgrund des Revisionsbezugs das eigens entwickelte Financial Process Mining, werden Prozessinstanzen ermittelt, sodass als Ergebnis eine Menge von Prozessinstanzen, auch genannt Sequenzen, entsteht. Eine Sequenz ist dabei jeweils eine Liste mit Belegen, die eine Instanz eines Ablaufs bilden. Die Ordnung in der Liste ist determiniert durch die Chronologie der enthaltenen Belege.

Schritt 2: Interne Kontrollen identifizieren

Die entwickelte Methode identifiziert nicht, wie zunächst möglicherweise nahe liegt, interne Kontrollen aus dem Datenbestand des Unternehmens. Es ist zu vermuten, dass ein solches Vorgehen impraktikabel wäre, denn es muss davon ausgegangen werden, dass sich interne Kontrollen nicht einfach aus einem ERP-System auslesen lassen, schon gar nicht deren Wirksamkeit im Hinblick auf konkrete Prozessinstanzen. Deshalb untersucht die entwickelte Methode den Buchungsstoff des Unternehmens im Hinblick auf Datenkonstellationen, die darauf hinweisen, dass ein Mangel an interner Kontrolle vorliegen könnte. Ein solcher genau definierter Hinweis wird im folgenden Indikator genannt. Durch dieses Vorgehen wird sichergestellt, dass auch die Betrachtung des internen Kontrollsystems vollständig empirisch induktiv betrieben wird. Gleichwohl wird die Betrachtungsweise umgekehrt: Während der Auditor klassischerweise wirksame interne Kontrollen (unter)sucht, geht die entwickelte Methode gewissermaßen „invers“ vor: Das Vorhandensein von Indikatoren bei Belegen im ERP-System ist eine Indikation für eine Schwäche im internen Kontrollsystem (bzw. eine Indikation für Abwesenheit von interner Kontrolle). Beispiele für Indikatoren sind: Rechnungen wurden zu spät überwiesen, sodass Skonto verloren ging; Belege wurden durch einen Administrator gebucht; eine Rechnung wurde vor der dazugehörigen Bestellung gebucht, ...

Schritt 3: Interne Kontrollen prüfen

Analog zu Schritt zwei werden im Rahmen der entwickelten Methode nicht interne Kontrollen geprüft, sondern Belege bzw. Sequenzen, die bzgl. der definierten Indikatoren auffällig sind. Auch hier gilt also die „inverse“ Sicht: Statt die Wirksamkeit von Kontrolle zu prüfen, wird das Vorhandensein von Mängeln (Anwesenheit von Indikatoren bei Belegen/ Sequenzen = Abwesenheit von Kontrolle) betrachtet. Gleichwohl zeigt sich bei ersten Anwendungen der beschriebenen Methode in der Praxis, dass zwar der Zusammenhang zwischen Sequenzen und Indikatoren als Informationseinheit vorliegt, der Prüfer aber ob des Datenvolumens Schwierigkeiten hat zu entscheiden, welche Abläufe einer genaueren Betrachtung wert sind, etwa im Rahmen von Stichprobenprüfungen oder Test of Details. Es ist deshalb von Nutzen, durch statistische Methoden herauszufinden, welche „Schwächeprofile“ in den Abläufen des Unternehmens stecken. Ein „Schwächeprofil“ soll ausdrücken, welche systematischen Kombinationen von Indikatoren (= Mängel/Schwächen) in den Abläufen des Unternehmens verborgen sind. Der Auditor kann somit durch eine Menge von „Schwächeprofilen“ die unternehmensindividuellen Prozessprobleme erfassen und sehr zielgerichtet und organisationsindividuell darauf eingehen. Die Berechnung der „Schwächeprofile“ wird durch eine explorative Faktorenanalyse umgesetzt.

Tabelle 1 fasst die entwickelte Methode zusammen.

Schritt	„klassisches manuelles“ Prüfvorgehen	Designed Artifact(s) der entwickelten Methode	Output der Designed Artifact(s)
1	Prozessverständnis	Financial Process Mining Algorithmus	Menge von <i>Sequenzen</i> bzgl. des gesamten Unternehmensdatenbestandes eines Geschäftsjahres
2	Interne Kontrollen identifizieren	Menge von Indikatoren, die mit den Sequenzen verknüpft werden	<i>Zuordnung</i> von <i>Indikatoren</i> zu den <i>Sequenzen</i>
3	Interne Kontrollen prüfen	Explorative Faktorenanalyse zur Identifikation von Prozessschwächen („Schwächeprofile“)	Extrahierte Faktoren, die „Schwächeprofile“ der Prozesse darstellen und die es dem Auditor erlauben, konkrete Sequenzen aufzugreifen.

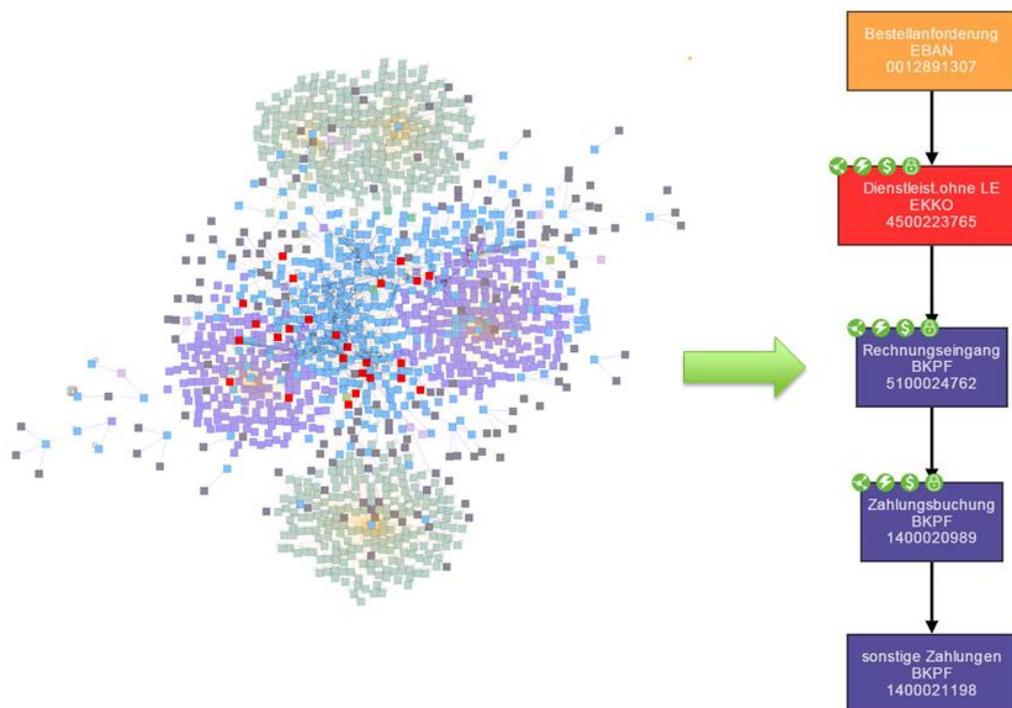
Tabelle 1: Artefakte der entwickelten Methode und deren Outputs

3. FINANCIAL PROCESS MINING

Die Ermittlung von Prozessinstanzen, folgend auch Sequenzen genannt, wird durch den Financial Process Mining-Algorithmus durchgeführt. Dabei wird ausgenutzt, dass die Belege im ERP-System miteinander verknüpft sind bzw. entsprechende Referenzen von einem Beleg auf einen anderen existieren. Der Algorithmus schreitet diese Referenzen ab und erzeugt dadurch eine Menge von Sequenzen oder Prozessinstanzen. Maßgeblicher Ansatzpunkt sind dabei die Belege des Rechnungswesens, da die Abläufe aus Sicht der Revision betrachtet werden und somit die integrierte Sichtweise aus Prozessfluss und Wertfluss von höchster Bedeutung ist. Der Algorithmus nutzt dabei die Belegreferenzen aus, die aufgrund der Offene-Posten-Buchhaltung vorliegen. Offene Posten in der Buchhaltung stellen sicher, dass zusammengehörige Buchungen sich gegenseitig ausgleichen. Offene-Posten-geführte Konten findet man in aller Regel im Bereich der Nebenbücher (Debitoren, Kreditoren etc.), aber auch im Hauptbuch (Wareneingangs-/Verrechnungskonto, Rückstellungskonten etc.). Letztendlich kann man sich die Verknüpfung von Belegen als einen Graphen vorstellen, der nach bestimmten Kriterien abgeschritten wird. Der Graph wird gewissermaßen in einzelne Sequenzen „zerschnitten“. Abbildung 1 zeigt schematisch einen solchen Graph von Buchungsbelegen.

Mit Hilfe eines Regelwerks, welches Startbelege im Graph und die Regeln zum Abschreiten der Kanten festlegt, werden dann alle Prozessinstanzen bzw. Sequenzen ermittelt. Eine einzelne Sequenz ist beispielhaft in Abbildung 1 dargestellt.

Die exakte Funktionsweise des Mining-Algorithmus ist an dieser Stelle nicht Gegenstand dieses Beitrages. Vielmehr wird an dieser Stelle auf bereits vorhandene Vorarbeiten verwiesen (Gehrke & Müller-Wickop 2010a; Gehrke & Müller-Wickop 2010b; Gehrke & Werner 2013; Werner et al. 2012).



4. INDIKATOREN

Für diese Studie wurden Indikatoren aus dem Bereich der ordnungsgemäßen Zugriffsrechte ausgewählt. Tabelle 2 zeigt die 19 ausgesuchten Indikatoren, auf deren Schwächen die Sequenzen des Unternehmensdatensatzes hin untersucht werden. Bis auf die Indikatoren 1 und 2 sind alle Indikatoren Risiken, die aus mangelnder Funktionstrennung resultieren. Diese Indikatoren schlagen in einer Sequenz an, wenn derselbe Nutzer zwei unterschiedliche, gemäß eines Regelwerks zuvor definierte, Prozessschritte durchgeführt hat. Zu der definierten Kombination der beiden Prozessschritte ist ein Risiko zugeordnet, welches in Tabelle 2 beschrieben ist. Es handelt sich um in der Revisionspraxis üblicherweise anzutreffende Risiken aus dem SAP-Umfeld.

Beispiel: Im Regelwerk ist definiert, dass der Prozessschritt „Kreditorenrechnung erfassen“ und „Zahlungslauf starten“ ein Risiko darstellt (z. B. „Fiktive Lieferantenrechnung anlegen und Zahlung dafür initiieren“). Wenn in einer Sequenz derselbe Nutzer beides durchgeführt hat, dann schlägt der Indikator mit diesem Risiko bei der Sequenz an.

Es sei angemerkt, dass das geschilderte Risiko des Indikators nicht bedeutet, dass tatsächlich etwas „Schlimmes“ oder gar Betrug vorgefallen ist. Vielmehr ist es ein Hinweis (Indikation) darauf, dass das Risiko in einer solchen Situation eingetreten sein kann und dieses als Mangel an Ordnungsmäßigkeit angesehen werden sollte.

No	Indikator	Hinweis
1	OPERATIONS DONE BY POWERFUL USERS	Markiert Buchungen, die mit Superuser- oder ähnlichen Zugriffsrechten gebucht wurden
2	ONE USER DID THE COMPLETE SEQUENCE	Markiert die komplette Sequenz (alle Belege), wenn diese durch denselben Nutzer bearbeitet / gebucht wurde
3	Adjust the subsidiary balance using the vendor invoice entry and then cover it up using journal entries	Nebensaldo mithilfe der Kreditorenrechnungserfassung korrigieren und Vorgang mit Journalbuchungen verschleiern
4	Create fictitious vendor invoice and initiate payment for it	Fiktive Lieferantenrechnung anlegen und Zahlung dafür initiieren
5	Change the customer master file and modify cash received	Kundenstammdatei ändern und empfangenes Bargeld manipulieren
6	Sales documents entered and released by the same person	Von derselben Person erfasste und freigegebene Verkaufsbelege
7	Maintain PO and accept the services through service acceptance	Bestellung bearbeiten und die Dienstleistungen per Leistungsabnahme annehmen
8	Adjust the AR subsidiary balance using AR payments and then conceal with journal entries	Nebensaldo der Debitorenbuchhaltung mithilfe von Zahlungen der Debitorenbuchhaltung anpassen und Vorgang mit Journalbuchungen verschleiern
9	Maintain a fictitious vendor and direct disbursements to it	Fiktiven Lieferanten bearbeiten und Auszahlung dorthin leiten
10	Maintain a sales doc and generate a billing doc for it	Verkaufsbeleg bearbeiten und einen Fakturabeleg dafür generieren
11	Purch unauth items and hide by not fully receiving order	Nicht autorisierte Positionen kaufen und durch unvollständigen Empfang des Auftrags verbergen
12	Create fictitious vendor and initiate payment to the vendor	Fiktiven Lieferanten anlegen und Zahlung an den Lieferanten initiieren
13	Change customer master and enter inappropriate invoice	Kundenstamm ändern und unangemessene Rechnung eingeben
14	Settle expenses from an unauthorized order	Aufwendungen aus einem nicht autorisierten Auftrag abrechnen
15	Maintain customer master records and post fraudulent payments	Kundenstammsätze bearbeiten und in betrügerischer Absicht erstellte Zahlungen buchen
16	Adjust the AR subsidiary balance using cash application and then conceal with journal entries	Nebensaldo der Debitorenbuchhaltung mithilfe des Zahlungsvorgangs anpassen und Vorgang mit Journalbuchungen verschleiern
17	Maintain an invoice and enter or change payments against it	Rechnung bearbeiten und dazugehörige Zahlungen eingeben oder ändern
18	Cover up shipment by maintaining a fictitious sales doc	Transport durch Bearbeiten eines fiktiven Verkaufsbelegs verschleiern
19	Requisition an item and create a PO from that req	Position anfordern und eine Bestellung anhand dieser Anforderung anlegen

Tabelle 2: Die in dieser Studie verwendeten Indikatoren

5. ZUSAMMENHANG ZWISCHEN INDIKATOREN UND PROZESSINSTANZEN

Nachdem Sequenzen berechnet wurden, müssen nun diese Sequenzen mit Indikatoren verknüpft werden, die in einer Sequenz angeschlagen haben. Erst nach dieser Verknüpfung ist es möglich, signifikante Prozessschwächen mit statistischen Methoden zu erkennen. Bezüglich der in diesem Beitrag verwendeten Indikatoren geschieht dies wie folgt:

- Indikator 1 (OPERATIONS DONE BY POWERFUL USERS): An einen Beleg, welcher durch einen Superuser gebucht wurde, wird dieser Indikator geheftet.
- Indikator 2 (ONE USER DID THE COMPLETE SEQUENCE): An alle Belege einer Sequenz wird der Indikator geheftet, wenn alle Belege einer Sequenz durch denselben User gebucht wurden.
- Indikator 3-19 (Funktionstrennungsrisiken): Der entsprechende Indikator wird jeweils an das Pärchen von Belegen in einer Sequenz geheftet, wenn diese Belege gemäß Regelwerk in Konflikt stehen.

6. STATISTISCHE ANALYSE DER SCHWÄCHEN DER GRUNDGESAMTHEIT DER PROZESSINSTANZEN

Die statistische Analyse in Form einer explorativen Faktorenanalyse basiert auf den ermittelten mit Indikatoren markierten Sequenzen. Folgend wird ein Indikator, der an einem bestimmten Beleg in einer Sequenz angeschlagen hat, ein Issue genannt (Instanz eines Indikators). Ein Issue hat eine eindeutige ID und gehört zu einem bestimmten Indikator und ist an einen bestimmten Beleg „geheftet“. Um die Kombination von Indikatoren in Sequenzen zu untersuchen wird für jede Sequenz die Kombination von Issues ermittelt. Ist bei zwei verschiedenen Sequenzen die Kombination der Issues gleich, so wird diese Issuekombination nur einmal betrachtet (merge). Tabelle 3 zeigt beispielhaft die Ermittlung der Kombinationen von Issues als Vorbereitung für die explorative Faktorenanalyse.

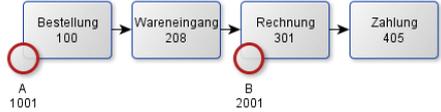
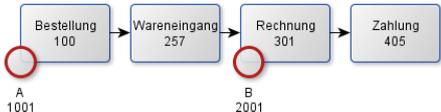
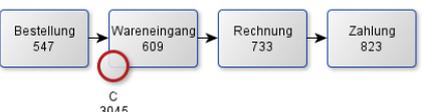
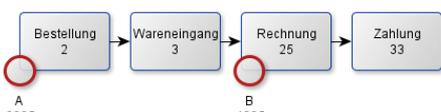
Sequenz	Beteiligte Belege und Indikatoren	Indikatoren	Kombinationen von Issues	Merge
1		A,B	1001,2001	1001,2001
2		A,B	1001,2001	
3		C	3045	Kein Merge
4		A,B	3035,4005	Kein Merge

Tabelle 3: Sequenzen und Kombinationen von Issues

In Tabelle 3 sind vier verschiedene Sequenzen zu sehen. Die Zahlen in den Belegen sind eindeutige Belegnummern. Die runden Symbole stellen an die Belege geknüpfte Issues dar des Indikators A, B bzw. C. Die Zahl unter den Issues ist die eindeutige ID des Issues. Man

beachte, dass sich die Sequenzen 1 und 2 nur durch den Beleg des Wareneingangs unterscheiden, der Rest der Belege der Sequenzen 1 und 2 sind dieselben. Dies bedeutet, dass es für den Ablauf in Sequenz 1 und 2 zwei verschiedene Wareneingänge gab, jedoch nur eine Bestellung, Rechnung und Zahlung. Entsprechend sind an die Bestellung und an die Rechnung auch dieselben Issues geheftet. Da die Kombination der Issues dieselbe ist, wird diese Kombination nur einmal berücksichtigt und es findet ein Merge statt. Sequenz 3 enthält gänzlich andere Belege und damit auch ein anderes Issue eines weiteren Indikators C (diesmal am Wareneingang). Genauso ist es mit Sequenz 4.

Für die explorative Faktorenanalyse ergibt sich aufgrund des Beispiels dann folgende Datenmatrix wie in Tabelle 4 gezeigt.

Fallkombination (Nummer der Datenzeile)	Indikator A	Indikator B	Indikator C
1	1	1	0
2	0	0	1
3	1	1	0

Tabelle 4: Abgeleitete Datenmatrix als Input für die folgende explorative Faktorenanalyse

Tritt ein Indikator in der Zeile auf, erhält die Zeile in der Spalte eine „1“. Ist der Indikator abwesend, hat der Datensatz in der Spalte eine „0“. Natürlich ist zu beachten, dass in echten Unternehmensdatensätzen die Anzahl der Zeilen die Anzahl der Spalten bei weitem übersteigt.

Die Datenmatrix wird nun einer explorativen Faktorenanalyse zugeführt. Die Faktorenanalyse nutzt die Korrelation der beobachteten Variablen (hier: Indikatoren) aus, um auf zugrundeliegende sogenannte latente (nicht direkt beobachtbare) Variablen zu schließen, die Faktoren genannt werden (Backhaus et al. 2011). Wenn – wie im Beispiel in Tabelle 4 – die Indikatoren A und B häufig (aber nicht zwingend) zusammen auftauchen, dann würde die Faktorenanalyse hervorbringen, dass es einen Faktor gibt, der starke „Anteile“ von Indikator A und B beinhaltet. Die berechneten Faktoren sollen im Rahmen der vorgestellten Methode „Schwächeprofile“ genannt werden. Sie zeigen auf, welche Mängel oder Schwächen systematisch als Kombination von Indikatoren im Unternehmensdatensatz vorherrschen. Die Faktorenanalyse hat die Eigenschaft, dass die extrahierten Faktoren im geometrischen Sinne orthogonal zueinander stehen, was bedeutet, dass die Faktoren sich klar voneinander unterscheiden und somit auch verschiedene unterschiedliche „Schwächeprofile“ im Datensatz erkannt werden können.

7. CASE STUDY

Im Folgenden wird eine Case Study präsentiert mit einem echten Unternehmensdatensatz. Es handelt sich um ein Produktionsunternehmen, bei dem SAP im Einsatz ist. Der tatsächliche Name kann aus Vertraulichkeitsgründen nicht genannt werden. Der Datensatz hat bereits die Schritte 1 und 2 durchlaufen und liegt insoweit aufbereitet vor. Als „Endergebnis“ soll nun die explorative Faktorenanalyse durchgeführt werden, um die „Schwächeprofile“ zu berechnen. Die involvierten Indikatoren wurden bereits oben beschrieben. Zur Durchführung der Faktorenanalyse wurde SPSS verwendet. Die Faktorenanalyse wurde durchgeführt auf Basis der Korrelationsmatrix der beobachtbaren Variablen. Das Ergebnis der Analyse ist die sogenannte rotierte Faktorenmatrix, die dann auch die beschriebenen „Schwächeprofile“ darstellt. Als Rotationsmethode wurde die übliche Varimax-Methode (Kaiser 1958) verwendet. Als Extraktionsmethode wurde die Hauptkomponentenanalyse angewendet. Die explorative Faktorenanalyse ist aufgrund ihres explorativen Charakters hypothesengenerierend und nicht, wie die konfirmatorische Faktorenanalyse, hypothesenprüfend. D. h. bei der explorativen Faktorenanalyse ist dem Nutzer unklar, wie die Faktoren aufgrund theoretischen Vorwissens gestaltet sind bzw. wie viele Faktoren überhaupt existieren (sollten). Aufgrund dieses Mangels

an Wissen wird die explorative Faktorenanalyse bemüht. Bzgl. der Bestimmung der Anzahl der zu extrahierenden Faktoren gibt es in der Literatur verschiedene Kriterien. An dieser Stelle wird sich des klassischen Kaiser-Guttman-Kriteriums (Guttman 1954; Kaiser & Dickman 1959) bedient, d. h. es werden so lange Faktoren extrahiert, bis der sogenannte Eigenwert des Faktors unter 1 fällt und somit der Faktor nicht mehr hinreichend viel der Varianz des Datensatzes erklären kann.

Komponente	Anfängliche Eigenwerte			Summen von quadrierten Faktorladungen für Extraktion			Rotierte Summe der quadrierten Ladungen		
	Gesamt	% der Varianz	Kumulierte %	Gesamt	% der Varianz	Kumulierte %	Gesamt	% der Varianz	Kumulierte %
1	2,234	11,757	11,757	2,234	11,757	11,757	1,866	9,821	9,821
2	2,021	10,639	22,396	2,021	10,639	22,396	1,825	9,606	19,426
3	1,760	9,264	31,661	1,760	9,264	31,661	1,721	9,060	28,486
4	1,526	8,030	39,691	1,526	8,030	39,691	1,560	8,209	36,695
5	1,315	6,920	46,611	1,315	6,920	46,611	1,448	7,619	44,314
6	1,276	6,716	53,328	1,276	6,716	53,328	1,390	7,316	51,630
7	1,178	6,199	59,527	1,178	6,199	59,527	1,351	7,108	58,738
8	1,133	5,963	65,490	1,133	5,963	65,490	1,283	6,752	65,490
Extraktionsmethode: Hauptkomponentenanalyse.									

Tabelle 5: Beitrag der Faktoren zur Erklärung der Varianz des Datensatzes

Tabelle 5 zeigt den Beitrag der Faktoren zur Varianz des Datensatzes. Gemäß dem Kriterium von Kaiser-Guttman werden 8 Faktoren extrahiert bei 19 anfänglichen Observablen, d. h. der Datensatz wird um 11 Dimensionen reduziert. Faktor 9 erklärt nur noch 5,18 % der Varianz, was weniger als 100/19 ist und deshalb wird auf alle Faktoren ab Faktor 9 verzichtet. Somit werden insgesamt ca. 65,5 % der Varianz erklärt. Ca. 34,5 % bleiben als „weißes Rauschen“ unerklärt.

	Komponente							
	1	2	3	4	5	6	7	8
OPERATIONS_DONE_BY_POWERFUL_USERS	-,067	-,855	-,242	,089	-,155	-,129	-,078	-,116
ONE_USER_DID_THE_COMPLETE_SEQUENCE	,088	,136	,118	-,025	,008	-,048	,050	,764
Adjust_the_subsidary_balance_using_the_vendor_invoice_entry_an	-,026	,463	-,106	,160	-,151	-,175	,005	,044
Create_fictitious_vendor_invoice_and_initiate_payment_for_it	-,020	,839	-,074	-,084	,156	,128	-,036	-,030
Change_the_customer_master_file_and_modify_cash_received	,313	,040	,026	-,012	-,053	-,011	,678	-,038
Sales_documents_entered_and_released_by_the_same_person	,017	,049	,195	,799	,003	-,036	-,035	-,015
Maintain_PO__accept_the_services_through_svc_acceptance	,007	,037	-,009	,845	-,012	-,054	-,005	-,002
Adjust_the_AR_subsidary_balance_using_AR_paymentsÄ_and_then_co	-,027	-,061	-,117	,397	,008	,083	,020	-,005
Maintain_a_fictitious_vendor_and_direct_disbursements_to_it	,003	,144	-,003	-,020	,815	,060	-,012	-,027
Maintain_a_sales_doc_and_generate_a_billing_doc_for_it	-,013	-,008	,875	-,019	-,011	-,008	,103	,014
Purch_unauth_items_and_hide_by_not_fully_receiving_order	-,003	-,162	-,011	,073	-,064	,842	,011	,034
Create_fictitious_vendor_and_initiate_payment_to_the_vendor	,003	-,020	-,019	,027	,826	-,034	,010	,033
Change_customer_master_and_enter_inappropriate_invoice	,713	-,033	-,041	,013	,090	,005	,265	,069
Settle_expenses_from_an_unauthorized_order	-,033	-,042	-,056	,003	-,002	,030	-,031	,816
Maintain_customer_master_records_and_post_fraudulent_payments	-,046	-,006	,063	,005	,040	-,001	,892	,048
Adjust_the_AR_subsidary_balance_usingÄ_cash_application_and_th	,688	,041	,045	-,026	-,057	-,015	,010	-,010
Maintain_an_invoice_and_enter_or_change_payments_against_it	,876	-,015	-,016	-,001	,003	,006	-,018	,012
Cover_up_shipment_by_maintaining_a_fictitious_sales_doc	,012	,008	,896	,014	-,013	-,015	-,016	,044
Requisition_an_item_and_create_a_PO_from_that_req	-,007	,310	-,019	-,032	,102	,773	-,025	-,057
Extraktionsmethode: Hauptkomponentenanalyse. Rotationsmethode: Varimax mit Kaiser-Normalisierung.								
a. Die Rotation ist in 6 Iterationen konvergiert.								

Tabelle 6: Extrahierte Faktoren oder „Schwächeprofile“

Tabelle 6 zeigt die extrahierten Faktoren aus der Analyse. Aufgabe des Nutzers (Auditors) der Faktorenanalyse ist es nun, diese Faktoren spaltenweise zu interpretieren. Dazu ist es sinnvoll, sich innerhalb eines Faktors die Variablen zu vergegenwärtigen, die stark in dem Faktor vertreten sind. Es ist hilfreich, die Interpretation der Faktoren anhand von zuvor gewählten Schwellenwerten zu bewerten, um einen Faktor bzgl. der in ihm enthaltenen Variablen zu interpretieren. Die Schwellenwerte sind dabei nicht wissenschaftlich ableitbar und unterliegen somit einer gewissen Willkür. An dieser Stelle werden die Schwellenwerte wie in Tabelle 7 abgebildet empfohlen.

Schwellenwert	Bezeichnung der Konstellation
Eine einzelne Variable $\geq 0,75$	„clear delinquent“
Mehrere Variablen $\geq 0,75$	„clear collaborators“
Eine oder mehrere Variable „stark negativ“ $\leq -0,3$	„steer clear“
Eine Variable mittelhoch $0,4 \leq x < 0,75$	„weak delinquent“
Mehrere Variablen mittelhoch $0,4 \leq x < 0,75$	„weak collaborating delinquents“

Tabelle 7: Schwellenwerte zur Interpretation von Faktoren / „Schwächeprofilen“

Im Lichte der empfohlenen Schwellenwerte wurden die Faktoren in Tabelle 6 bereits farblich markiert. Es ergeben sich dann folgende Einordnungen wie in Tabelle 8 dargestellt.

Faktor	„clear delinquent“	„clear collaborators“	„steer clear“	„weak delinquent“	„weak collaborating delinquents“
1	X				X
2	X		X	X	
3		X			
4		X			
5		X			
6		X			
7	X			X	
8		X			

Tabelle 8: Einordnung der „Schwächeprofile“ gem. Schwellenwerte

Die Interpretation der Faktoren soll nun anhand der zwei ausgewählten Faktoren 1 und 3 demonstriert werden:

Faktor 1 enthält die Einordnung „clear delinquent“, das heißt, in diesem Faktor dominiert eine Variable, nämlich das durch den Indikator ausgedrückte Risiko, dass eine Rechnung erstellt wird und die dazugehörige Zahlung gebucht wird („Maintain an invoice and enter or change payments against it“, Stärke 0,876). Dieses Phänomen fällt aufgrund der weiteren Einordnung „weak delinquent“ manchmal zusammen mit dem Risiko (Stärke 0,688), dass der Nebensaldo der Debitorenbuchhaltung mithilfe von Zahlungen der Debitorenbuchhaltung angepasst wird und der Vorgang mit Journalbuchungen verschleiert wird („Adjust the AR subsidiary balance using AR payments and then conceal with journal entries“). Faktor 1 lädt weiterhin als „weak delinquent“ (Risiko 0,713) auf „Change customer master and enter_inappropriate_invoice“.

Faktor 3 beinhaltet, wie auch die Faktoren 4, 5, 6 und 8, nur die Einordnung „clear collaborators“. D. h. mindestens zwei Indikatoren häufen sich jeweils in diesen Faktoren (bei allen diesen Fällen sind es jeweils genau zwei Indikatoren). Faktor 3 beinhaltet also das häufige Zusammentreffen der Risiken „Maintain a sales doc and generate a billing doc for it“ (Stärke 0,875) und „Cover up shipment by maintaining a fictitious sales doc“ (Stärke 0,896). Dies lässt darauf schließen, dass es im Verkauf mangelnde Funktionstrennung gibt, denn scheinbar sind die drei zentralen Tätigkeiten des Verkaufs Auftrag bearbeiten (sales doc), Lieferung (shipment) und Faktura (billing) häufig nicht getrennt, sodass es hier zu dieser Häufung der angesprochenen Risiken kommt.

Interessant ist auch, wenn ein Faktor eine stark negative Variable enthält („steer clear“) z. B. in Faktor 2 bei „Operations done by powerful users“ (Stärke -0,855). Dies bedeutet, dass bei Vorhandensein anderer Risiken (in Faktor 2 z. B. „Create fictitious vendor invoice and initiate

payment for it') dieser Indikator regelmäßig abwesend ist, sich also gewissermaßen „abstoßend“ verhält. Dies ist insofern eine gute Nachricht, als dass sich vermuten lässt, dass systematisch keine Superuser für die beschriebenen Schwächen in diesem Faktor verantwortlich sind.

8. HANDLUNGSEMPFEHLUNGEN FÜR DEN AUDITOR

Im Sinne der vorgestellten Methode sollte der Auditor im Rahmen seiner Prüfungshandlungen mit den Ergebnissen der Faktorenanalyse wie folgt verfahren:

- 1) *Empfehlung 1: Bewertung des „Designs“ der „Schwächeprofile“:* Interpretation der Faktoren/„Schwächeprofile“ und Bewertung im Hinblick auf das im „Schwächeprofil“ steckende Risiko. Hier ist das ‚Professional Judgement‘ des Auditors gefragt. Die prozentuale Aufklärung der Varianz des Faktors liefert immerhin ein erstes quantitatives Kriterium zur Bewertung eines „Schwächeprofils“.
- 2) *Empfehlung 2: Prüfung der systematischen Risiken im Unternehmensdatensatz:* Prüfung einer Stichprobe von Sequenzen für jedes (als interessant erachtetes) „Schwächeprofil“.
- 3) *Empfehlung 3: Prüfung der unsystematischen Risiken im Unternehmensdatensatz:* Prüfung von Sequenzen, die zwar viele Issues besitzen (wo also viele Indikatoren angeschlagen haben), die aber zu keinem der Faktoren/„Schwächeprofile“ passen.

Die Interpretation der Faktoren wurde bereits im vorangegangenen Kapitel beispielhaft erläutert.

Zur stichprobenhaften Prüfung der systematischen Risiken (Empfehlung 2) sollten Sequenzen gezielt ausgewählt werden, die dann einer Einzelfallprüfung unterzogen werden (Test of Details). Diese Sequenzen sollten dem jeweils betrachteten Faktor möglichst „nahekommen“. Dafür wird jede existierende Sequenz im Hinblick auf den spezifischen Faktor durch Berechnung eines Scores bewertet. Der Score berechnet sich aus der Summe der Gewichte der Indikatoren des Faktors (siehe die Spalten der Faktoren in Tabelle 6). Ein Gewicht wird in den Score/die Summe einbezogen, wenn bei der Sequenz der entsprechende Indikator angeschlagen hat. Sequenzen mit hohen Scores/Summen entsprechen dann besonders dem „Schwächeprofil“ des Faktors und kommen somit in die Stichprobe.

Zur Prüfung der unsystematischen Risiken (Empfehlung 3) sollten Sequenzen geprüft werden, bei denen

- viele verschiedene Indikatoren angeschlagen haben und
- die sehr verschieden von den (allen) extrahierten Faktoren/„Schwächeprofilen“ sind, also zu keinem der Faktoren gut passen.

Die Auswahl solcher Sequenzen kann auf verschiedene Art und Weise geschehen, z. B. wie folgt:

- 1) Berechne für jede Sequenz wie in Empfehlung 2 für jeden Faktor den Score und speichere als Score für die Sequenz den maximalen Score (über alle betrachteten Faktoren).
- 2) Adjustiere den Score aus 1) für jede Sequenz um die Anzahl der verschiedenen Indikatoren, die in der Sequenz angeschlagen haben, z. B. wie folgt:

$$\text{Adjusted Score} = \frac{\text{Score}}{\ln(1 + N)}$$

N ist die Anzahl der verschiedenen angeschlagenen Indikatoren in der Sequenz. Die Sequenz wird nur betrachtet, wenn $N > 0$.

Nach dieser Prozedur sind für eine Stichprobe die Sequenzen interessant, die einen besonders kleinen adjusted Score haben. Weiterhin ist von Interesse, wie die zu untersuchende Stichprobe auf die verschiedenen Faktoren/„Schwächeprofile“ aufgeteilt werden sollte. Da die Faktoren/„Schwächeprofile“ unterschiedlich umfangreich zur Erklärung der Varianz der Daten beitragen, könnte dies als Aufteilungsmaßstab für die Stichprobe dienen. Faktoren, die viel Varianz erklären, könnten als „systematischer“ als andere aufgefasst werden und deshalb sollte auch ein größerer Stichprobenanteil zu diesem Faktor alloziert werden. Legt der Auditor eine Gesamtstichprobe von z. B. 100 zu prüfenden Sequenzen fest, so würde die Stichprobe nach dem „Varianzerklärungsausmaß“ gemäß Tabelle 5 in der Case Study wie in Tabelle 9 alloziert.

Prüfung	Faktor	Anteil erklärter Varianz des Faktors	Größe der Teilstichprobe (Anzahl Sequenzen)
Systematisches Risiko	1	11,76%	12
	2	10,64%	11
	3	9,26%	9
	4	8,03%	8
	5	6,92%	7
	6	6,72%	7
	7	6,2%	6
	8	5,96%	6
Unsystematisches Risiko	/	34,51%	34
Summe			100

Tabelle 9: Risikoorientierte Verteilung der gesamten Stichprobe auf die extrahierten Faktoren

9. AUSBLICK

Der vorliegende Beitrag hat eine Methode vorgestellt, wie Audits von Geschäftsprozessen weitestgehend toolgestützt erfolgen können. Die Methode kommt dabei ohne Prämissen oder „Vorwissen“ bezüglich der Organisation und der Prozesse des Unternehmens aus. Es handelt sich deshalb um einen sehr stringenten Ansatz, weil durchgehend empirisch induktiv vorgegangen wird: aus der umfangreichen und für den Menschen nicht überschaubaren Grundgesamtheit der Unternehmensdaten und –prozesse wird auf „Schwächeprofile“ geschlossen, die durch eine explorative Faktorenanalyse dem Auditor sehr verdichtet präsentiert werden können. Dadurch können spezifische Unternehmensabläufe (Sequenzen) sehr zielgerichtet und orientiert an den spezifischen Schwächen des Unternehmens untersucht werden, um den Finger direkt „in die Wunde zu legen“, im Sinne eines effektiven und effizienten Audits.

Weiterer Forschungsbedarf ergibt sich durch die Notwendigkeit der Anwendung der Methode auf weitere Unternehmensdatensätze und das daraus resultierende Feedback, welches sinnvoll in die entworfene Methode zu integrieren wäre. Auch wird die Basis und Anzahl der Indikatoren erweitert werden, um noch weitere Schwächebereiche von Prozessen mit berücksichtigen zu können. Derzeit liegt noch ein Vorrat von weiteren 120 verschiedenen bereits implementierten Indikatoren vor. Eine explorative Faktorenanalyse mit dieser erweiterten Menge an Indikatoren steht noch aus.

10. LITERATUR

- Askelson, K.; Lanza, R., Millar, P.; Prosch, M.; Sparks, D. E. (2009): Global Technology Audit Guide (GTAG) 13: Fraud Prevention and Detection in an Automated World, The Institute of Internal Auditors (IIA), zugegriffen über: http://www.theiia.org/bookstore/downloads/freetomembers/0_2030.dl_gtag13.pdf am 14.07.2015.
- Backhaus, K.; Erichson, B.; Plinke, W.; Weiber, R. (2011): Multivariate Analysemethoden - Eine anwendungsorientierte Einführung, Berlin/Heidelberg, Springer, Berlin.
- Bellino, C.; Hunt, S. (2007): Global Technology Audit Guide (GTAG) 8: Auditing Application Controls, The Institute of Internal Auditors (IIA), zugegriffen über: http://www.ii.nl/SiteFiles/IIA_leden/Praktijkgidsen/GTAG8.pdf am 14.05.2015.
- Gehrke, N.; Müller-Wickop, N. (2010a): Rekonstruktion von Geschäftsprozessen im Finanzwesen mit Financial Process Mining, in: Lecture Notes in Informatics, Proceedings der Jahrestagung Informatik 2010, IT-supported Service Innovation and Service Improvement, Leipzig.
- Gehrke, N.; Müller-Wickop, N. (2010b): Basic Principles of Financial Process Mining, Proceedings of the 16th Americas Conference on Information Systems, Lima, Peru.
- Gehrke, N.; Werner, M. (2013): Process Mining, in: WISU – Wirtschaft und Studium, 2013 Jg., Heft 7, S. 934 – 943.
- Guttman, L. (1954): Some necessary conditions for common factor analysis, in: Psychometrika, Jg. 19, Heft 2, S. 149 – 161.
- IDW PS 330 (2002): Abschlussprüfung bei Einsatz von Informationstechnologie (Quelle: WPg 21/2002, S. 1167 ff., FN-IDW 11/2002, S. 604 ff.) vom 24.09.2002.
- IDW RS FAIT 1 (2002): Grundsätze ordnungsmäßiger Buchführung bei Einsatz von Informationstechnologie (in: WPg 21/2002, S. 1157 ff., FN-IDW 11/2002, S. 649 ff.) vom 24.09.2002.
- IDW PS 261 (2006): Feststellung und Beurteilung von Fehlerrisiken und Reaktionen des Abschlussprüfers auf die beurteilten Fehlerrisiken (in: WPg 22/2006, S. 1433 ff., FN-IDW 11/2006, S. 710 ff., WPg Supplement 4/2009, S. 1 ff., FN-IDW 11/2009, S. 533 ff.) vom 09.09.2009.
- International Federation of Accountants (IFAC-I) (2008): Handbook of international auditing, assurance, and ethics pronouncements 2008 Edition Part I, New York.
- ISACA (2009): CobiT® and Application Controls: A Management Guide, zugegriffen über: <http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/COBIT-and-Application-Controls-A-Management-Guide.aspx> am 14.07.2015.
- PUBLIC LAW 107–204—JULY 30 (2002): 107th Congress, USA, zugegriffen über: <http://www.sec.gov/about/laws/soa2002.pdf> am 14.07.2015.
- Kaiser, H. F. (1958): The varimax criterion for analytic rotation in factor analysis, in: Psychometrika, 23 Jg., Heft 3, S. 187 – 200.
- Kaiser, H. F.; Dickman, K. (1959): Analytic determination of common factors, in: American Psychologist, Heft 14, S. 425 – 438.
- Mertens, P. (1995): Wirtschaftsinformatik - Von den Moden zum Trend, in: König, W. (Hrsg.): Wirtschaftsinformatik '95, Heidelberg, Physica, 1995, S. 25 – 64.
- Werner, M.; Gehrke, N.; Nüttgens, M. (2012): Business Process Mining and Reconstruction for Financial Audits, in: Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS-45), Hawaii.

KOMPLEXITÄTSMANAGEMENT FÜR IT-ORGANISATIONEN



Prof. Dr. Arno Müller
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Prof. Dr. Hinrich Schröder
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: IT-Komplexität entwickelt sich zu einem zunehmenden Problem in Unternehmen, die sich den Herausforderungen der Digitalisierung stellen müssen. Im folgenden Artikel wird ein System zur Ermittlung und Messung der Komplexitätstreiber in IT-Organisationen vorgestellt, auf dessen Basis sich Handlungsempfehlungen zum Komplexitätsmanagement ableiten lassen. Dabei wird auf den Ansatz des Lean IT-Managements zurückgegriffen, in dem Prinzipien wie Selbstorganisation und flexible Teams gezielt zur Komplexitätsbeherrschung umgesetzt werden.

Keywords: Komplexität, IT-Organisation, Lean IT-Management

1. EINFÜHRUNG

Unternehmen sehen sich mit einer steigenden Komplexität in Prozessen und Strukturen konfrontiert. Diese verursacht direkte und indirekte Kostenwirkungen und beeinflusst damit langfristig die Wettbewerbsfähigkeit. Das Management von Komplexität und Variantenvielfalt ist mittlerweile eine Grundaufgabe nachhaltig erfolgreicher Unternehmensführung.

Für IT-Organisationen gilt dies in besonderem Maße: Die Kernaufgabe der Unternehmens-IT (IT-Abteilung / Corporate IT) liegt in der optimalen Unterstützung der Geschäftsprozesse des Unternehmens mit der dafür geeigneten IT-Infrastruktur und passenden IT-Applikationen. Zur Sicherung der Zukunftsfähigkeit der IT-Anwendungslandschaft ist es erforderlich, dass das IT-Management permanent technologische Veränderungen verfolgt und systematisch bewertet, um Innovationen, wenn diese reif für die kommerzielle Nutzung sind, möglichst zielgerichtet einzuführen. Aktuelle Instrumente der Digitalisierung wie Mobile IT, Cloud Computing, Social Media oder Big Data rücken dabei in zunehmendem Maße in den Fokus. Begleitend zu den technischen Veränderungen der IT verändern sich auch die Fähigkeiten und Anforderungen der IT-Anwender. Getrieben durch die IT-Durchdringung des privaten Bereichs und die damit verbundene zunehmende Erfahrung und IT-Affinität der Nutzer steigen die Erwartungen und Anforderungen an die Leistungen der Corporate IT.

Prof. Dr. Arno Müller ist seit 1994 Professor an der NORDAKADEMIE. Nach dem Studium an der TH Darmstadt, der Promotion an der Universität Passau und Tätigkeiten in Forschung und Beratung zu den Themen Logistik und Produktionsorganisation war er Leiter der Logistik bei einem Automobilzulieferanten. Seine Forschungsschwerpunkte sind die Auswirkungen von IT-Technologien auf die Geschäftsprozesse in Unternehmen und die Potentiale des Lean Managements. Er ist geschäftsführender Gesellschafter der bps – business process solutions GmbH Hamburg und Vorstand des e-ThinkTank e.V..
E-Mail: arno.mueller@nordakademie.de

Prof. Dr. Hinrich Schröder ist seit 2002 Professor der NORDAKADEMIE mit den Fachgebieten Betriebswirtschaftliche Anwendungssysteme, Projekt- und IT-Management und IT-Controlling im Fachbereich Informatik. Nach dem Studium der Betriebswirtschaftslehre an der Universität Göttingen und Promotion an der Universität Hamburg war er mehrere Jahre als Unternehmensberater tätig und verfügt über Erfahrungen als Projektleiter bei SAP-Einführungen und der Optimierung von Geschäftsprozessen. Seine Forschungsschwerpunkte liegen in der Analyse der wirtschaftlichen Wirkungen von IT-Systemen und im IT- und Prozessmanagement.
E-Mail: hinrich.schroeder@nordakademie.de

Weiterhin sind zunehmend Anforderungen an die Unternehmens-IT zu beobachten, die von anderen Stakeholdern bzw. aus dem Unternehmensumfeld abzuleiten sind. Insbesondere getrieben durch die Möglichkeiten des Cloud-Computing verändern sich die Betriebsmodelle für die IT. Neben Ansprüchen der unmittelbaren Geschäftspartner, der Kunden und Lieferanten, die durch die zunehmende Vernetzung immer häufiger und kurzfristiger Veränderungen in den IT-gestützten Geschäftsprozessen initiieren, resultieren daraus vor allem Compliance-Anforderungen unterschiedlichster Prägung, und das Thema IT-Sicherheit erhält eine zentrale Bedeutung.

All diese Faktoren sorgen dafür, dass das Managen einer Unternehmens-IT ein Management der damit verbundenen Komplexität bedingt. In diesem Beitrag werden nach einer Klärung des Komplexitätsbegriffs zunächst die Treiber der Komplexität in IT-Organisationen identifiziert und ein System zu deren Messung vorgeschlagen. Darauf aufbauend wird ein Self-Assessment zum Komplexitätsmanagement in IT-Organisationen vorgestellt.⁴

2. KOMPLEXITÄT

Komplexität wird als „das Zusammentreffen einer strukturierten Vielschichtigkeit, resultierend aus der Anzahl und Diversität der Elemente eines Systems sowie deren gegenseitigen Verknüpfungen und der dynamischen Veränderlichkeit der gegenseitigen Beziehungen der Systemelemente“ (Piller 2000: 179) bezeichnet. Im Gegensatz zum allgemeinen Sprachgebrauch, in dem mit dem Attribut „komplex“ häufig vermittelt werden soll, dass etwas schwierig oder kompliziert sei, ist hier vor allem die dynamische Veränderlichkeit des Systems bestimmend. Ulrich/Probst definieren in diesem Sinne Komplexität als „als Fähigkeit eines Systems, in einer gegebenen Zeitspanne eine große Zahl von verschiedenen Zuständen annehmen zu können“ (Ulrich & Probst 1991: 58).

Systeme können demnach einfach, kompliziert, relativ komplex oder äußerst komplex sein (vgl. Ulrich & Probst 1991: 58; Schuh 2005: 5). Ein komplexes System entzieht sich wegen seiner Eigendynamik der exakten Analyse und ist daher einer gezielten Beeinflussung auf direktem Wege nicht zugänglich und somit mit klassisch hierarchischen Methoden nur schwer zu managen. Zur Abgrenzung dienen folgende Indikatoren (vgl. Ulrich & Probst 1991: 61; Schuh 2005: 5; zu einer Übersicht weiterer Dimensionen vgl. Meyer 2007: 25):

Vielfalt des Systems (statische Betrachtung):

- die **Elementevielfalt** ist die Menge der Elemente (Bestandteile, Komponenten), gemessen anhand der Anzahl und der Unterschiedlichkeit der Elemente
- die **Beziehungsvielfalt** ist die Menge der Beziehungen zwischen den Elementen (die Wechselwirkungen), erfasst anhand der Anzahl und der Unterschiedlichkeit der Beziehungen

Veränderungsgrad des Systems (dynamische Betrachtung)

- das **Änderungspotenzial** bzw. die **Dynamik** ist die Menge möglicher Zustände, die die Elemente und die Beziehungen im Laufe der Zeit einnehmen können

Über diese Dimensionen lassen sich die folgenden Klassen von Systemen abgrenzen:

⁴ Beide Ansätze sind Ergebnisse eines von den Autoren unter Beteiligung von Fach- und Führungskräften aus mehreren Unternehmen organisierten Arbeitskreises zur „IT mit Zukunft“.

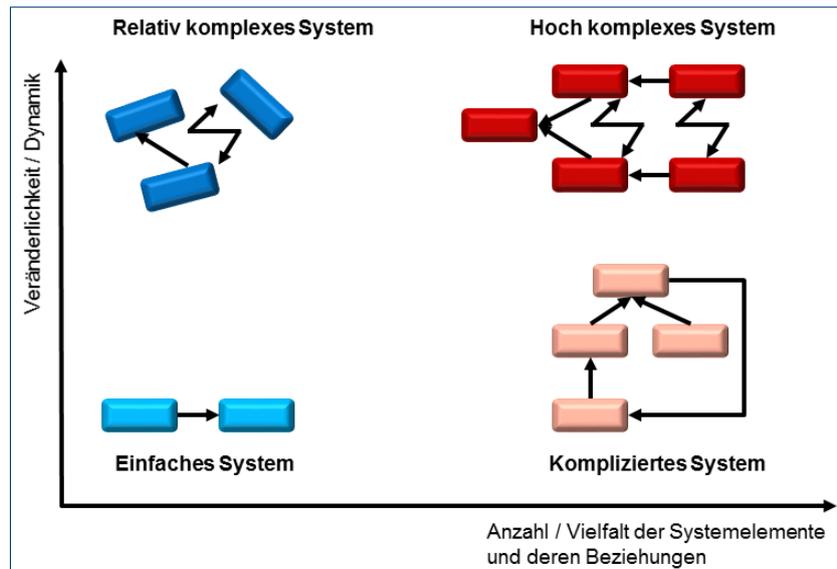


Abbildung 1: Kompliziertheit und Komplexität (in Anlehnung an Ulrich & Probst 1991: 61)

Ein einfaches System ist gekennzeichnet durch wenige Einflussfaktoren, stabile Beziehungen, schwache Verknüpfungen und klare Zusammenhänge von Ursache und Wirkung. Ein kompliziertes System hingegen hat viele Einflussfaktoren, enge Verknüpfungen, stabile Beziehungen, vorhersehbares Verhalten und bekannte Zusammenhänge zwischen Ursache und Wirkung. Geringe Komplexität weisen Systeme auf, die einige Einflussfaktoren, dynamische Beziehungen und unklare Wirkungszusammenhänge besitzen. Ein hoch komplexes System schließlich hat viele Einflussfaktoren, sehr dynamische Beziehungen, schwer erkennbare Muster und kaum erkennbare Zusammenhänge von Ursache und Wirkung.

Die IT im Unternehmen beschränkt sich nicht nur auf die Infrastruktur, sondern umfasst IT-Services auf allen Ebenen zur Unterstützung der Geschäftsprozesse und erfüllt damit die Merkmale eines hoch komplexen Systems. Eine Vielzahl an Elementen (z. B. technische Komponenten, Anwendungssysteme, Anwender, IT-Mitarbeiter) ist ebenso gegeben wie eine hohe Veränderlichkeit, bspw. durch sich ständig wandelnde externe Erwartungen der Kunden und Anwender sowie fortlaufende Veränderungen der Technologie, die sich durch die aktuellen IT-Trends nochmals deutlich erhöhen dürften (vgl. Abbildung 2).

Im Kern können hierbei vier Anforderungsdimensionen unterschieden werden (vgl. Müller & Schröder 2015: 162, 167)

- Business Innovation, d. h. die Möglichkeit mit Hilfe der IT neue Produkte und Leistungen zu entwickeln, bzw. bestehende Produkte und Leistungen merkbar weiterzuentwickeln.
- Prozessoptimierung, d. h. die Möglichkeit, mit Hilfe der IT betriebliche Prozesse nachhaltig zu verbessern.
- Serviceerbringung, d. h. die effiziente Bereitstellung der richtigen IT-Leistungen für den Kunden in der geforderten Qualität.
- Mitarbeiter, d. h. die Unterstützung der Zusammenarbeit in verteilten Teams, der Wissensarbeit und die Erhöhung der räumlichen und zeitlichen Flexibilität der Mitarbeiter mit Hilfe der IT.

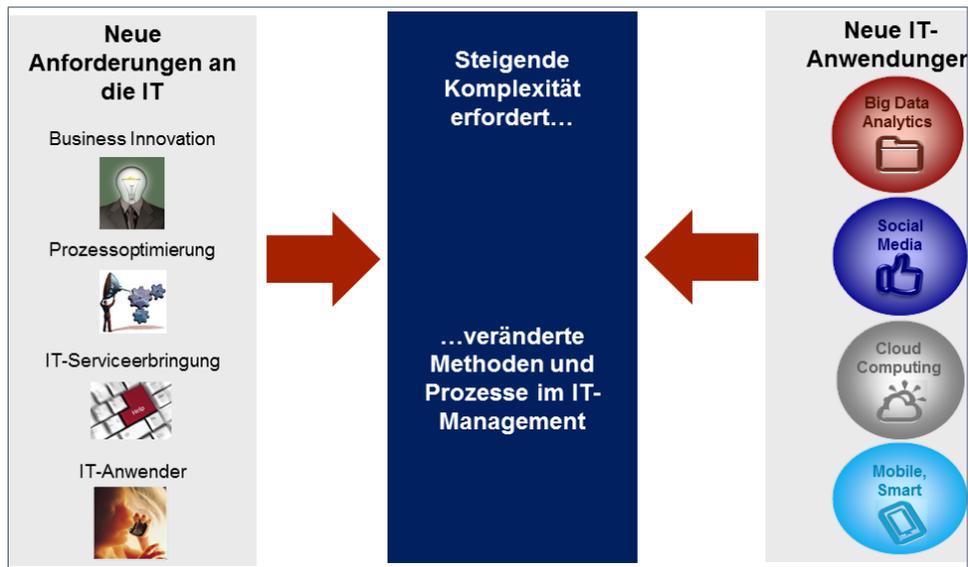


Abbildung 2: Komplexitätstreiber in der IT

3. MESSUNG UND PROGNOSE DER KOMPLEXITÄT IN IT-ORGANISATIONEN

Zum Themengebiet Komplexitätsmanagement und den damit verbundenen Problemen lassen sich unterschiedliche Studien heranziehen (vgl. z. B. Fraunhofer IPA 2015; speziell für IT-Organisationen vgl. Aumendo 2011, Maturity 2015). Im IT-Bereich werden als typische negative Auswirkungen einer hohen Komplexität Aspekte wie Kostenanstieg und Ressourcenengpässe durch Systemvielfalt, Qualitätsprobleme durch instabile Systemumgebungen sowie geringere Flexibilität durch eine große Anzahl nebeneinander existierender Systeme genannt (vgl. Aumendo 2011: 16). Auch in Bezug auf IT-Projekte wird ein unmittelbarer Zusammenhang zwischen Projektkomplexität und dem Risiko des Scheiterns eines Projektes festgestellt (vgl. InfoQ 2015).

Besonders problematisch ist es, wenn eine Negativspirale in Gang gesetzt wird: Neue Potenziale der IT, der Zwang zur Digitalisierung des Business oder Marktanforderungen und Preisdruck führen zu neuen Anforderungen an die Unternehmens-IT, diese reagiert mit neuen Anwendungen und Systemen, die einen Komplexitätsanstieg bewirken. Wenn dies wiederum zu erhöhten Kosten in der IT und im Business führt, werden neue Anforderungen mit dem Ziel der Kostenreduktion an die IT gestellt, was wiederum neue Anwendungen und Systeme zur Folge haben könnte – ein Teufelskreis, wenn komplexitätsbedingte Kosten mit mehr Komplexität bekämpft werden.

Im Produktionsmanagement wird ein ähnliches Szenario als „Komplexitätsfalle“ bezeichnet. Durch die markt- und kundengetriebene Forderung nach Diversifikation steigt die Variantenvielfalt und Produktkomplexität, was wiederum die Komplexität der betrieblichen Prozesse deutlich erhöht. Mit zeitlicher Verzögerung werden dann Investitionen in Informations-, Steuerungs- und Organisationssysteme getätigt, um die komplexitätsbedingten Probleme zu lösen. Dies führt zu überproportionalen Gemeinkostensteigerungen, die im Regelfall nicht durch die zusätzlichen Erlöse kompensiert werden können (Adam & Johannwille 1998: 5; Adam 1998: 49-50).

Der richtige Umgang mit der Komplexität in der IT stellt deshalb einen wichtigen Erfolgsfaktor dar. So stellt die Hackett Group in einer Studie fest, dass besonders erfolgreiche IT-Organisationen durch eine vergleichsweise geringe IT-Komplexität, gemessen u. a. an der Anzahl technischer Plattformen und Applikationen, gekennzeichnet sind. Deutlich wird dies an signifikant geringeren Kosten sowohl für die Technologie als auch für die durch IT unterstützten Prozesse in den Unternehmen, die eine geringere Anzahl an Applikationen betreiben. Die Effektivität der IT wird dagegen mit steigender Anzahl an Server-, Entwicklungs- und Applikationsplattformen geringer (vgl. Dorr & Holland 2014: 3-4).

Ausgangspunkt für ein Verhindern der komplexitätsbedingten Probleme sollte eine Identifikation der Komplexitätstreiber in der IT und deren Bewertung sein. Daraus abgeleitet können Handlungsfelder bestimmt und priorisiert werden, mit denen letztlich Maßnahmen zur Komplexitätsbeherrschung verbunden sind. Diese Roadmap (vgl. Abbildung 3) umfasst dabei die drei Ebenen Kosten und Nutzen, Methoden und Werkzeuge sowie Personal und Führung.

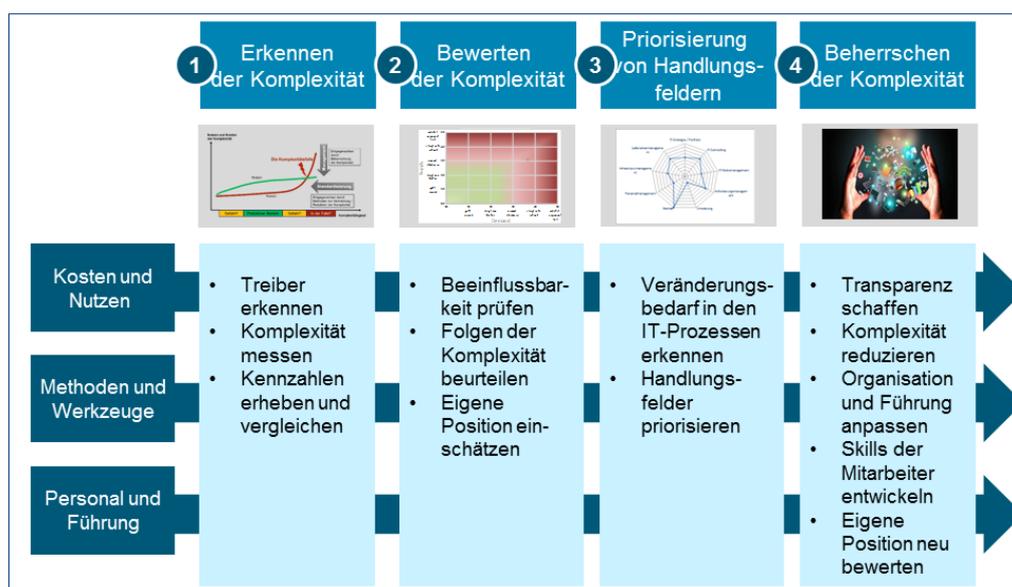


Abbildung 3: Roadmap des Komplexitätsmanagements

Eine hohe Komplexität in der IT ist die Folge von Maßnahmen, die der Steigerung des Unternehmenswertes dienen sollen. Cloud Systeme können Kosten senken, mobile Anwendungen erleichtern den Mitarbeitern die Arbeit, die Nutzung gleicher Anwendungen auf verschiedenen Endgeräten erhöht die Flexibilität des IT-Einsatzes. Individuellere Anpassungen an Kundenwünsche, Kostenreduktion beim Kunden durch Automatisierungen, höhere Kundenzufriedenheit durch kundenfreundlichere Applikationen, schnelle Reaktion auf neue Anforderungen sind positive Effekte der Veränderung des IT-Einsatzes. Die negative Wirkung ist jedoch, dass hierdurch die Komplexität und die damit verbundenen Kosten exponentiell steigen, wenn es nicht gelingt, diese zu beherrschen oder so weit als möglich zu vermeiden.

Im IT-Umfeld ist diese Gefahr insbesondere dadurch gegeben, dass der Nutzen meist im Business entsteht (hybride Produkte mit zunehmender IT-Unterstützung / Internet der Dinge, optimierte Prozesse, Integration der Kunden, verbesserte Wissensarbeit etc.), während die negativen Effekte die IT-Organisation betreffen (Kostenanstieg, Qualitätsreduktion, geringere Flexibilität, erhöhte Betriebsrisiken etc.) und auf diese Weise nicht immer erkannt und nicht in gebotenen Maße berücksichtigt werden.

Unternehmen sind gefordert, den „richtigen“, d. h. wirtschaftlich sinnvollen Komplexitätsgrad zu finden, was letztlich nur durch ein aktives Komplexitätsmanagement erreicht werden kann. Der oben beschriebenen Roadmap folgend sind dazu zunächst die Treiber festzustellen, die sich in Anlehnung an die gängige Aufteilung von IT-Organisationen in die Bereiche „Demand“ und „Supply“ aufteilen lassen.

Auf der Demand-Seite kann dann eine weitere Untergliederung in die Dimensionen *Funktionen*, *Schnittstellen*, *Daten* sowie *Dynamik* vorgenommen werden. Zur Ermittlung der funktionalen Komplexität sind bspw. die Zahl der Anwender, Anzahl unterschiedlicher Anwendungen je Anwender oder Zahl der Rollen der Anwender (Varietät) zu bestimmen. Die Schnittstellenkomplexität ergibt sich u. a. aus der Anzahl extern angebundener Anwendungen oder Geräte, die Datenkomplexität bspw. aus der Anzahl der Informationsobjekte und dem Speicherplatzbedarf. Die Dynamik schließlich wird über die Anzahl der Changes, Incidents oder neuer User ermittelt. Eine vollständige Übersicht zeigt Abbildung 4.

Komplexität im Demand	Funktionen	Schnittstellen	Daten	Dynamik
1. Funktional			3. Daten	
1.1	Zahl der Anwender – intern		3.1	Anzahl Informationsobjekte – Typen
1.2	Anzahl unterschiedlicher Anwendungen je Anwender		3.2	Anzahl Dateien
1.3	Zahl der Rollen der Anwender (Varietät)		3.3	Speicherplatzbedarf
1.4	Funktionsumfang der IT-Systeme – Anzahl Use Cases		4. Dynamik	
1.5	Zahl der Organisationseinheiten, die die IT bereut		4.1	Anzahl Changes – neue Rollen, Funktionen (Minor Changes)
1.6	Zahl der SLAs, die die IT vereinbart hat oder vereinbaren müsste (Serviceverträge mit den "Kunden")		4.2	Anzahl Changes – neue Rollen, Funktionen (Major Changes - Projekte mit Kunden)
2. Schnittstellen zu den Kunden / Anwendern der IT - NICHT IT Dienstleister			4.3	Anteil der Changes, der einen hohen Neuheitsgrad aufweist
2.1	Externe Schnittstellen - Anzahl		4.4	Neue Regularien und Vorgaben
2.2	Externe Schnittstellen - Anzahl der Typen / Protokolle		4.5	Anzahl Incidents
2.3	Extern angebundene Anwendungen		4.6	Anzahl neuer User
2.4	Extern angebundene User			
2.5	Extern angebundene Geräte			

Abbildung 4: Komplexitätstreiber Demand

Der Bereich Supply beinhaltet die Dimensionen *Applikationen*, *Outsourcing*, *Infrastruktur* und ebenfalls *Dynamik*. Während die Applikationskomplexität u. a. über die Anzahl der Anwendungssysteme und der dafür erforderlichen Skill-Bereiche zu ermitteln ist, bestimmt sich die Outsourcing-Komplexität über die Anzahl der Provider oder Cloud-Services. Die Komplexität der Infrastruktur wird bspw. durch die Anzahl der Configuration Items und deren unterschiedliche Typen beeinflusst, die Dynamik auf der Supply-Seite durch die Anzahl neuer Dienstleister pro Jahr oder die Mitarbeiterfluktuation im IT-Bereich. Eine entsprechende Übersicht zeigt Abbildung 5.

1. Applikationen 1.1 Anzahl der Anwendungen 1.2 Anzahl der redundanten Anwendungen (wenig Standardisierung) 1.3 Erforderliche Skill Bereiche in der IT 1.4 Anzahl der Mobility Services (Apps) 1.5 Anzahl der Social Media Solutions 1.6 Anzahl Anwendungen zur Datenanalyse (Big Data....)	3. Infrastruktur 3.1 Anzahl Infrastrukturelemente / Configuration Items 3.2 Anzahl unterschiedlicher Typen der CIs 3.3 Anzahl unterschiedlicher Service Level / Reaktionszeiten 3.4 Anzahl CIs außerhalb der eigenen Unternehmensgrenzen 3.5 Anzahl verschiedener Betriebssysteme / Basissysteme
2. Outsourcing 2.1 Anzahl externer Dienstleister 2.2 Anzahl von Externen unterstützte Systeme 2.3 Anzahl Prozessschnittstellen – zu DL 2.4 Anzahl der Cloud Services / SaaS 2.5 Anzahl der Cloud Dienstleister	4. Dynamik 4.1 Anzahl neuer Dienstleister pro Jahr 4.2 Anzahl neuer SLAs / OLAs 4.3 Anteil der Changes die externe Dienstleister betreffen bzw. einbeziehen 4.4 Anteil der Incidents die externe Dienstleister betreffen bzw. einbeziehen 4.5 Anteil neuer IT-Mitarbeiter pro Jahr (Wachstum / Fluktuation)

Komplexität im Supply
Applikationen
Outsourcing
Infrastruktur
Dynamik

Abbildung 5: Komplexitätstreiber Supply

Die zur Messung benötigten Daten lassen sich i. d. R. problemlos aus den gängigen Systemen zum IT-Servicemanagement extrahieren und in ein Messwerkzeug übernehmen. Dazu kann dann jeweils eine Abschätzung in einfacher Skalierung (geht zurück / steigt wie bisher / nimmt stärker zu / steigt sehr schnell / wächst exponentiell) vorgenommen werden. Als Ergebnis ergibt sich ein differenziertes Bild, das IT-Verantwortlichen die Möglichkeit bietet, den Komplexitätsgrad ihrer Organisation zu visualisieren und anschließend zu beurteilen. Eine mögliche Auswertung zeigt die Abbildung 6.

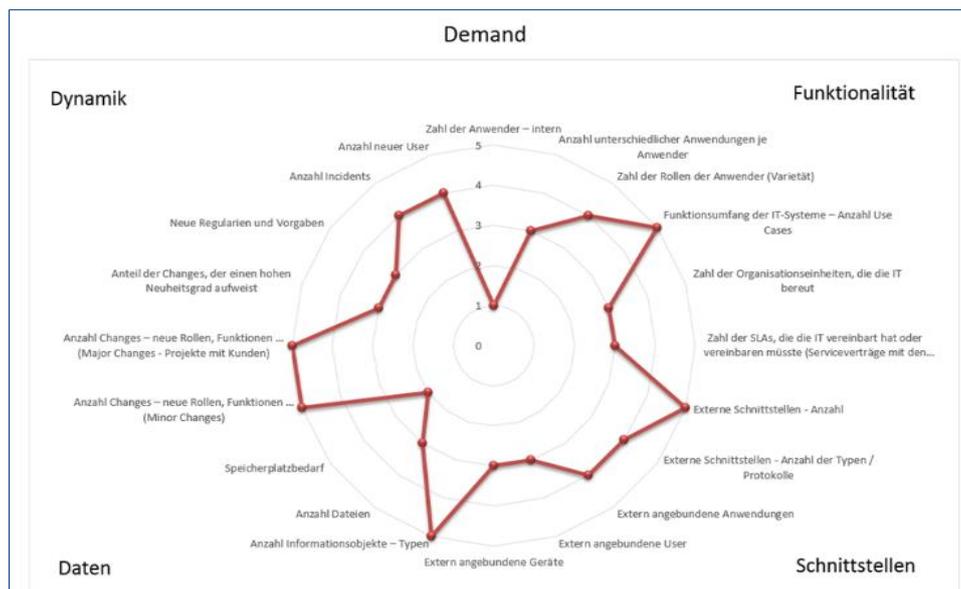


Abbildung 6: Auswertung der IT-Komplexität

Die so ermittelte Positionierung liefert Anhaltspunkte, um komplexitätsbedingte Probleme zu identifizieren und geeignete Maßnahmen zu entwickeln, auf die im folgenden Abschnitt eingegangen wird.

4. METHODEN UND SKILLS ZUM MANAGEMENT DER KOMPLEXEREN IT DER ZUKUNFT

In Anlehnung an die oben gezeigte Roadmap müssen sich nach einer Standortbestimmung Handlungsfelder und Maßnahmen des Komplexitätsmanagements anschließen. In Bezug auf die IT werden hierfür zahlreiche Empfehlungen genannt. So wird bspw. das Enterprise Architecture Management (EAM) als Schlüssel zur Komplexitätsbeherrschung herausgestellt, das dafür u. a. die folgenden Maßnahmen beinhalten sollte (vgl. Hanschke 2011: 68-69; Grebe & Danke 2013: 4; Tiemeyer 2013: 93):

- Erhöhung der Transparenz
- Eliminierung von Vielfalt und Heterogenität durch Standardisierung
- Szenario-basierte Rationalisierung von Applikationen
- Reduktion der Infrastruktur-Technologiemuster, Aufräumen der IT-Landschaft
- Reduktion der Anforderungen durch intelligentes Demand Management
- Erhöhung der Flexibilität durch Komponenten mit standardisierten Schnittstellen
- Shared-Services-Modell und optimiertes Sourcing

Aber reichen diese Empfehlungen wirklich aus? Reicht es, überflüssige Anforderungen und Systeme zu eliminieren, Leistungen und Prozesse zu standardisieren und Aufgaben an Externe auszulagern?

Als Strategien des Komplexitätsmanagements (vgl. dazu auch Wildemann 2013: 69) lassen sich aus den genannten Punkten vor allem die **Reduktion** (durch Eliminierung oder Standardisierung von Elementen) und **Vermeidung** (durch eine gezielte Auswahl neuer Elemente) ableiten (vgl. Abbildung 7). Damit wird der Versuch unternommen, aus einem hoch komplexen System ein nur noch relativ komplexes System zu machen, in dem die Vielzahl der Systemelemente verringert wird. Dies ist aber im IT-Bereich nur bis zu einem bestimmten Grad möglich, da Anwender- und Kundenanforderungen zu erfüllen sind, die eine gewisse Individualität und Systemvielfalt erfordern. Zudem wird durch eine Reduktion der Vielfalt und Anzahl der Systemelemente nicht das Kernproblem des Komplexitätsmanagement adressiert: die Dynamik und Veränderlichkeit des Systems bleiben weiterhin hoch. Komplexität in der IT ist vorhanden - es kommt im Management darauf an, diese zu erkennen und die richtigen Fähigkeiten zu entwickeln, damit umzugehen. Dieser Ansatz lässt sich als **Beherrschung** der Komplexität umschreiben (vgl. dazu grundlegend Malik 2006: 169 ff.)

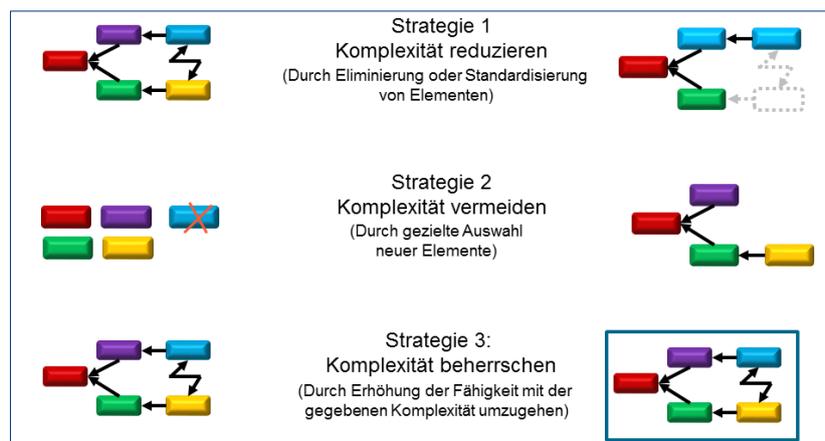


Abbildung 7: Strategien des Komplexitätsmanagement

Problematisch ist dabei, dass in komplexen Systemen Methoden und Techniken für komplizierte Systeme nicht zum Erfolg führen. Auf entsprechende Management- bzw. Problemlösungsfehler wird an unterschiedlichen Stellen hingewiesen (vgl. Ulrich & Probst 1991: 109-113 und 265-270; Probst & Gomez 1991: 5-6; Vester 2002: 36-39; Pfläging 2015: 18-19). Patentrezepte, die bei Bedarf hervorgeholt werden, eine zentrale Steuerung, das Verändern oder Optimieren einzelner Systemelemente, sind demnach nicht geeignet, komplexitätsbedingte Probleme zufriedenstellend zu lösen.

Übertragen auf IT-Organisationen bedeutet dies, dass viele weit verbreitete Maßnahmen im IT-Management nicht zielführend sind. So entspringen viele Methoden des Referenzsystems ITIL eher dem Managementansatz für komplizierte, als dem für komplexe Systeme. Beispielsweise wird dort eine direkte Kopplung zwischen Anwender und IT-Mitarbeiter durch einen systematischen Prozess über die Rollen Requirements Manager, Change Manager, Security Manager und Change Board ersetzt, was in einem stabilen Umfeld durchaus zielführend sein kann. Wenn die Anforderungen und der Systemzustand aber kurzfristig veränderlich sind, ist ein solcher zentralistischer Ansatz, der festen Regeln folgt, nicht in der Lage, die Komplexität zu beherrschen.

Ein Unternehmen sollte vielmehr so organisiert werden, dass es sich so weit wie möglich selbst organisieren und regulieren kann (Malik 2004: 315 ff.; Malik 2006: 489 ff.). Während komplizierte Systeme durch Methoden, Wissen, Lernen, Steuern und Regeln gemanagt werden können, sind für komplexe Systeme Ideen, Können, Üben, Führen und Prinzipien erforderlich (Schoeller 2009: 92-99). Flexibilisierung, Erhöhung der Verhaltensvarietät und Förderung der Kommunikation sind demnach angemessene Reaktionen auf die zunehmende Komplexität und Dynamik der Umwelt (Ulrich & Probst 1991: 270; Pfläging 2015: 17).

Ansätze des Komplexitätsmanagements in IT-Organisationen sollten sich daher nicht, wie häufig vorgeschlagen, auf die Technologie beschränken, sondern müssen vielmehr die Menschen, die die Technologie bereitstellen und nutzen, in den Fokus rücken. Selbstorganisation, Kooperation, Blick auf das Ganze, vernetztes Denken, Eigenverantwortung, Ergebnisorientierung, Kommunikation und Vertrauen sind Eigenschaften, die gezielt zu entwickeln sind.

Im Lean IT-Management (Müller, Schröder & von Thienen 2011) werden diese Anforderungen konsequent umgesetzt. Diesem Ansatz folgend kann mit Hilfe eines Self-Assessments eine Standortbestimmung durchgeführt werden, die sich konsequent auf alle Prozesse des IT-Managements bezieht. Zu jedem der für das IT-Management relevanten Prozesse sind dafür Empfehlungen zum Komplexitätsmanagement formuliert, die sich an den grundlegenden Prinzipien und Methoden des Lean IT-Management orientieren. In einem für das Self-Assessment entwickelten Werkzeug sind diese Empfehlungen als Leitfragen formuliert, die vom Unternehmen als Selbsteinschätzung beantwortet werden können, um den individuellen Reifegrad des Komplexitätsmanagements zu ermitteln.

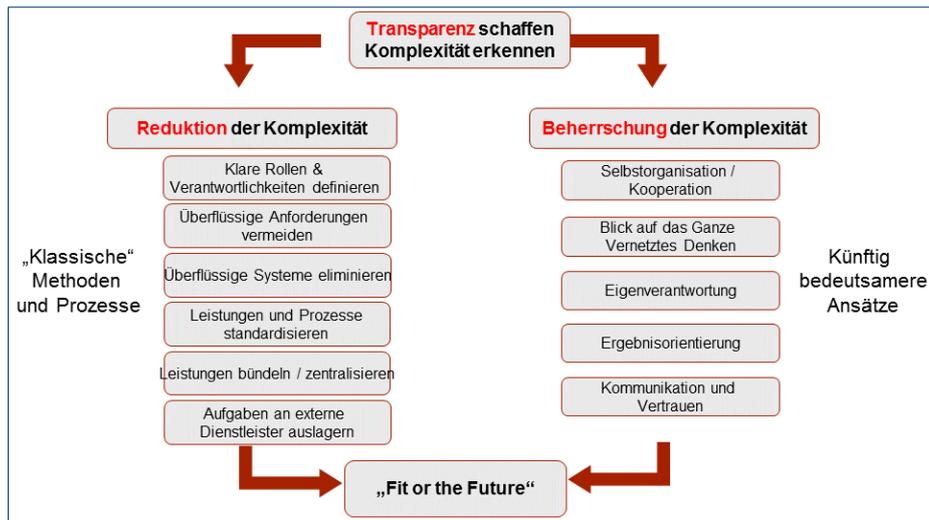


Abbildung 8: Empfehlungen für das IT-Management (in Anlehnung an Wiemeyer 2005: 5)

Die Leitfragen werden den vorherigen Überlegungen entsprechend in die drei Bereiche *Transparenz schaffen*, *Reduktion der Komplexität* und *Beherrschung der Komplexität* unterschieden (vgl. Abbildung 8). Mit dem ersten Schritt „Transparenz schaffen“ werden Empfehlungen verbunden, die dazu dienen, Komplexität zunächst überhaupt zu erkennen und zu differenzieren, welche Sachverhalte vielleicht nur kompliziert sind und mit klassischen Methoden und Prozessen angegangen werden können. Mit entsprechenden Leitfragen wird bspw. überprüft, ob eine Unternehmensarchitektur mit den Komponenten Prozesse, Organisation und Anwendungen sowie deren Verknüpfungen beschrieben ist oder ob typische IT-Risiken hinsichtlich Schadenshöhe und Eintrittswahrscheinlichkeiten analysiert und priorisiert sind.

Darauf aufbauende Leitfragen zur Komplexitätsreduktion überprüfen bspw., ob die tatsächliche Nutzung der in der Unternehmensarchitektur gelisteten Systeme regelmäßig erhoben und analysiert wird, um nicht genutzte Funktionen zu identifizieren, oder ob es klare Governance-Richtlinien und Risk Audits der IT in den Fachbereichen gibt.

Angesichts der zunehmenden Dynamik im IT-Umfeld und der damit verbundenen Begrenzung der klassischen Methoden zur Komplexitätsreduktion sind die Ansätze zur Komplexitätsbeherrschung als künftig bedeutsamer einzustufen. Typische Fragestellungen lauten hier, ob in die Entwicklung der Unternehmensarchitektur Vertreter des Business und IT-Mitarbeiter eingebunden werden oder ob die stärkere Selbstverantwortung des Business für IT-Risiken bekannt und unternehmensweit akzeptiert ist.

Insgesamt erstrecken sich die Fragen über die Führungsprozesse *Strategie/Projektportfoliomanagement*, *IT-Controlling*, *IT-Risikomanagement*, die Leistungsprozesse *Anforderungsmanagement*, *Produkt- und Projektmanagement* sowie *IT-Betrieb*, und schließlich die Supportprozesse *Personal*, *Infrastruktur* und *Lieferantenmanagement*. Beispielhaft ist hier dieser Fragenkatalog für den Prozess „Anforderungsmanagement“ dargestellt:

Leitfragen je Prozess		Kategorie	Bewertung der Umsetzung						
Anforderungsmanagement		Wert	nicht bewerten	Umsetzungsgrad					
				gar-nicht	1	2	3	4	perfekt 5
1	Die Kunden der IT sind klar definiert und deren Erwartungen an die IT sind bekannt.	heute schon relevant	3				x		
2	Die Themenfelder der anstehenden Veränderungen bei den IT-Kunden sind der IT bekannt	heute schon relevant	0	x			x		
3	Die IT nimmt eine beratende Rolle für ihre Kunden ein und ist bei der Identifikation von Handlungsfeldern systematisch eingebunden	heute schon relevant	3				x		
4	Es gibt einen klar geregelten Prozess zur Aufnahme und Bewertung der Anforderungen der IT-Kunden. Key-User und Anwender werden einbezogen	heute schon relevant	3				x		
5	Die Bedeutung einer Anforderung wird vom Fachbereich vorgegeben und relativ zu anderen Anforderungen bewertet	heute schon relevant	5						x
6	Die Phasen von der Entstehung einer Anforderung bis zur Entscheidung über die Art der Erfüllung sind beschrieben und mit Gate-Entscheidungen gekoppelt	heute schon relevant	5						x
7	Die IT ist in der Lage mit unklaren Anforderungsbeschreibungen zu arbeiten. Durch stetige Kommunikation und iteratives Vorgehen werden Anforderungen konkretisiert.	künftig bedeutsamer	1		x				
8	Veränderungen der Anforderungen werden als alltäglich akzeptiert und in Projekten und im Betrieb neutral abgearbeitet	künftig bedeutsamer	2			x			
9	Business und IT stimmen neue Anforderungen mit allen Stakeholdern ab – der Blick auf das Ganze wird gewährleistet	künftig bedeutsamer	1		x				
10	Die wahren Anforderungen hinter dem Wunsch nach Einsatz der IT-Trends werden bewusst gemacht	künftig bedeutsamer	1		x				

Abbildung 9: Fragebogen Self-Assessment zum Anforderungsmanagement

Insgesamt umfasst das Assessment 9 Dimensionen mit jeweils 10-12 Fragen, so dass sich ein Set von ca. 100 Fragen ergibt. Dies ist einerseits hinreichend ausführlich, um eine Standortbestimmung durchzuführen, andererseits aber noch so einfach, um es in maximal zwei Workshopterminen abzarbeiten.

Im Ergebnis werden die Prozesse deutlich, in denen noch Handlungsbedarf besteht und die somit die Handlungsfelder in der oben erläuterten Roadmap bilden. Konkrete Maßnahmen ergeben sich dann aus den Empfehlungen zu den Leitfragen.

5. FAZIT

Die Anwendung des vorgestellten Methodensets ermöglicht es, die Komplexität der IT als eigenes Konstrukt zu messen und Veränderungen zu erkennen. Wenn die Analyse zeigt, dass die Komplexität trotz aller Anstrengungen zur Reduktion und Vermeidung durch Standardisierung signifikant steigt, müssen die Managementmethoden kritisch hinterfragt werden, da es nicht zielführend ist, die bei komplizierten Systemen erfolgreichen Lösungsansätze bei einer hohen Komplexität unverändert weiter zu nutzen.

Das Self-Assessment ermöglicht es, Defizite bei den Managementmethoden zu erkennen, und weist auf Ansätze des Lean IT-Management hin, die sich u. a. im Bereich des agilen Projektmanagements etabliert haben. Die dort gewonnenen Erkenntnisse zur Lösung von Problemstellungen in komplexen Systemen können auf andere Bereiche der IT-Organisation übertragen werden. Dies zu untersuchen ist Gegenstand weiterer Vorhaben.

6. LITERATUR

- Adam, D. (1998): Produktionsmanagement, 9. überarbeitete Auflage, Nachdruck 2001, Wiesbaden, Gabler.
- Adam, D.; Johannwille, U. (1998): Die Komplexitätsfalle, in: Adam, D. (Hrsg.), Komplexitätsmanagement, Schriften zur Unternehmensführung, Band 61, Wiesbaden, S. 5 – 28.
- Aumendo Polacek, Klostermann & Partner, Unternehmensberatung (Hrsg.) (2011): Komplexität in der IT, Studie, Berlin.
- Dorr, E.; Holland, S. (2014): The world-class performance advantage: How leading IT organizations outperform their peers, in: The Hackett Group (Hrsg.), Executive Insight Management Issue May 2014.
- Fraunhofer IPA (Hrsg.) (2015): Studie Komplexitätsbewirtschaftung 2014, Stuttgart.
- Grebe, M.; Danke, E. (2013): Simplify IT. Six Ways to Reduce Complexity, The Boston Consulting Group, Studie.
- Hanschke, I. (2011): Beherrschen der IT-Komplexität mithilfe von EAM, in: Wirtschaftsinformatik & Management, 3. Jg.; Heft 4, S. 66 – 71.
- InfoQ (2015): Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, zugegriffen über: <http://www.infoq.com/articles/standish-chaos-2015n> am 29.01.2016.
- Malik, F. (2004): Systemisches Management, Evolution, Selbstorganisation. Grundprobleme, Funktionsmechanismen und Lösungsansätze für komplexe Systeme, 4. unveränderte Auflage, Bern, Stuttgart, Wien, Haupt Verlag.
- Malik, F. (2006): Strategie des Managements komplexer Systeme. Ein Beitrag zur Management-Kybernetik evolutionärer Systeme, 9., unveränderte Auflage, Bern, Stuttgart, Wien, Haupt Verlag.
- Maturity GmbH (Hrsg.) (2015): IT-Komplexität 2015, Studie, München.
- Meyer, C. M. (2007): Integration des Komplexitätsmanagements in den strategischen Führungsprozess der Logistik, Berlin, Stuttgart, Wien, Haupt Verlag.
- Müller, A.; Schröder, H.; von Thienen, L. (2011): Lean IT-Management: Was die IT aus Produktionssystemen lernen kann, Wiesbaden, Gabler.
- Müller, A.; Schröder, H. (2015): IT mit Zukunft – Vorgehensmodell zur Ableitung von Handlungsprioritäten zur Einführung von IT-Innovationen, in: Plate, G. (Hrsg.): Forschung für die Wirtschaft 2014, Göttingen, Cuvillier Verlag, S. 159 – 170.
- Pfläging, N. (2015): Organisation für Komplexität. Wie Arbeit wieder lebendig wird – und Höchstleistung entsteht, 2. Auflage, München, Redline Verlag.
- Piller, F. T. (2000): Mass Customization, Wiesbaden, Deutscher Universitätsverlag.

- Probst, G. J. B.; Gomez, P. (1991): Vernetztes Denken. Ganzheitliches Führen in der Praxis, 2., erweiterte Auflage, Wiesbaden, Gabler.
- Schoeller, N. (2009): Internationales Komplexitätsmanagement am Beispiel der Automobilindustrie, Diss., Aachen.
- Schuh, G. (2005): Produktkomplexität managen: Strategien - Methoden – Tools, 2. überarbeitete Auflage, München, Wien, Hanser.
- Tiemeyer, E. (2013): Enterprise Architecture Management (EAM) – IT Architekturen erfolgreich planen und steuern, in: Tiemeyer, E. (Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis, 5. Auflage, München, Carl Hanser Verlag.
- Ulrich, H.; Probst, G.J.B. (1991): Anleitung zum ganzheitlichen Denken und Handeln – Ein Brevier für Führungskräfte, 3., erweiterte Auflage, Bern, Verlag Paul Haupt.
- Vester, F. (2002): Die Kunst vernetzt zu denken, aktualisierte und erweiterte Taschenbuchausgabe Mai 2002, München, Deutscher Taschenbuch Verlag.
- Wiemeyer, M. (2005): Management Komplexer Projekte - oder: PMI – das Handbuch des Misslingens, Whitepaper, Eppstein, zugegriffen über: <http://ratwechsel.de/html/texte.html> am 29.01.2016.
- Wildemann, H. (2013): Komplexitätsmanagement in Vertrieb, Beschaffung, Produkt, Entwicklung und Produktion 14. Aufl. München, TCW Transfer-Centrum.

KRITISCHER VERGLEICH VON METHODEN ZUR BEWERTUNG VON SOFTWAREARCHITEKTUREN



Stefanie Jasser, M. Sc.
Universität Hamburg, Fachbereich Informatik

Prof. Dr. Joachim Sauer
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: In der professionellen Softwareentwicklung nimmt die Softwarearchitektur eine wichtige Rolle ein. Sie ist das wesentliche Ergebnisartefakt des Softwareentwurfs und dient u. a. als Blaupause für die spätere Entwicklung. Zur Bewertung der Qualität einer Softwarearchitektur und ihrer Eignung für einen bestimmten Anwendungsbereich existieren verschiedene Verfahren. Diese lassen sich in szenariobasiert, erfahrungsbasiert und metrikbasiert kategorisieren. Die wichtigsten Verfahren werden in diesem Artikel auf Basis eines Literaturreviews ausgewählt, vorgestellt und systematisch miteinander verglichen. Der Schwerpunkt liegt auf szenariobasierten Bewertungsverfahren.

Keywords: Softwarearchitektur, Bewertung, Review, ATAM, ALMA

1. BEDEUTUNG VON SOFTWAREARCHITEKTUR

In der professionellen Softwareentwicklung spielt heutzutage die Softwarearchitektur (kurz: Architektur) eine besondere Rolle (Reussner & Hasselbring 2008). Es gibt verschiedene Definitionen von Softwarearchitektur. Häufig wird die folgende Definition von Bass et al. verwendet:

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both (Bass et al. 2013: 4).

Eine Softwarearchitektur besteht demnach hauptsächlich aus Softwareelementen, die auch als Subsysteme oder Komponenten bezeichnet werden, und den Beziehungen zwischen ihnen. Der innere Aufbau der Komponenten ist für die Architektur unwichtig. Man unterscheidet die Sollarchitektur, als vom Softwarearchitekten intendierte und vorgegebene Architektur, von der Istarchitektur, die sich tatsächlich im implementierten System wiederfindet.

Die Bedeutung von Softwarearchitektur für die Praxis der Softwareerstellung hat Kari Smolander in Interviews mit Architekten, Designern und Managern herausgearbeitet und vier Metaphern für Softwarearchitektur identifiziert (Smolander 2002): Softwarearchitektur kann 1. als *Blaupause* einer Anwendung gesehen werden. Sie spezifiziert ein System so weit, dass auf ihrer Grundlage die Anwendung implementiert werden kann. Des Weiteren ist die

Stefanie Jasser, M. Sc. arbeitet als wissenschaftliche Mitarbeiterin am Fachbereich Informatik der Universität Hamburg in den Arbeitsgruppen "Softwareentwicklungs- und -konstruktionsmethoden" und "Sicherheit in verteilten Systemen". In ihrer Dissertation beschäftigt sie sich mit der Frage, wie sichere Softwaresysteme entwickelt werden können. An der NORDAKADEMIE hat sie ihr Bachelor- und Masterstudium in Wirtschaftsinformatik absolviert und ist als nebenamtliche Dozentin im Einsatz.
E-Mail: s.jasser@nordakademie.de

Prof. Dr. Joachim Sauer ist Sprecher des Fachbereichs Informatik und Studiengangsleiter des Bachelorstudiengangs Angewandte Informatik. Nach dem Studium zum Diplom-Informatiker an der Universität Hamburg promovierte er im Arbeitsbereich Softwaretechnik. Er sammelte Erfahrungen in der Lehre an der HAW Hamburg und war in der Wirtschaft als Softwareentwickler und –architekt sowie als IT-Berater und Projektleiter tätig. Seine Forschungsschwerpunkte sind die Softwaretechnik und –architektur.
E-Mail: joachim.sauer@nordakademie.de

Softwarearchitektur 2. als *Literatur* geeignet, ein System für eine spätere Anpassung oder Weiterentwicklung zu dokumentieren. Das Verständnis für das System wird erhöht und Komponenten können dadurch leichter wiederverwendet werden. Damit unterschiedliche Akteure über eine Anwendung sachkundig sprechen können, wird über die Softwarearchitektur 3. eine *Sprache* definiert. Eine Architektur kann 4. als *Entscheidung* betrachtet werden, da die Architektur als Folge von Entwurfsentscheidungen entstanden ist und erforderliche Ressourcen für ein System festlegt.

Eine passende und gut ausgearbeitete Softwarearchitektur erleichtert damit die Implementierung von Anwendungen wesentlich. Eine ungeeignete kann zu schlecht aufgebauten Anwendungen führen, die sich schwer weiterentwickeln lassen.

Aber auch eine anfangs passende Architektur kann ihre positiven Eigenschaften verlieren, da sie durch mangelnde Pflege und inkompatible Änderungen an Anwendungen im Laufe der (Weiter-)Entwicklung veraltet. Gründe dafür können beispielsweise fehlendes Wissen über die Sollarchitektur bei den Entwicklern oder Zeitdruck sein. Man spricht dann von Architekturerosion (Perry & Wolf 1992). Die Weiterentwicklung einer Softwarearchitektur (dies bezeichnet man als Architekturevolution (Reussner & Hasselbring 2008)) muss gezielt gesteuert und überprüft werden, um Architekturerosion und ein in der Folge schlecht wartbares System zu vermeiden.

2. BEWERTUNG VON SOFTWAREARCHITEKTUREN

Die Problematik einer schlecht passenden oder erodierten Architektur kann auf verschiedenen Wegen angegangen werden, beispielsweise durch eine konstruktive Qualitätssicherung. Diese soll mit verschiedenen Maßnahmen während des Entwicklungsprozesses die Qualität einer Softwarearchitektur bewahren. Dabei findet vor allem ein Abgleich von Ist- und Sollarchitektur statt, der Mängel in der Umsetzung und unpassende Vorgaben der Sollarchitektur aufdecken soll.

In diesem Artikel steht der Ansatz im Mittelpunkt, die Softwarearchitektur eines Systems über mehr oder weniger formale Methoden zu bewerten. Die Bewertung bildet die Grundlage, um daraus Entscheidungen über Korrekturmaßnahmen an der Architektur und Strategien für ihre Weiterentwicklung abzuleiten. Damit lassen sich potenzielle Risiken aufdecken und angehen sowie die Umsetzung von Anforderungen an die Qualität überprüfen. Die wichtigsten Verfahren werden im Folgenden vorgestellt und anschließend kritisch miteinander verglichen.

Prinzipiell lassen sich Bewertungsverfahren in qualitative und quantitative Verfahren unterteilen (Eeles et al. 2014). Zur den qualitativen Verfahren zählen szenariobasierte und erfahrungsbasierte Bewertungsverfahren. Metrikbasierte Verfahren sind zunächst quantitativ, werden aber oft um qualitative Analysen ergänzt. Einige Verfahren lassen sich schon früh im Entwicklungsprozess einsetzen. So kann die Eignung einer Architektur festgestellt werden, bevor Aufwand in ihre Umsetzung geflossen ist. Mit allen Verfahren lässt sich die Softwarearchitektur eines fertig implementierten Systems bewerten.

3. ERFAHRUNGSBASIERTE BEWERTUNGSVERFAHREN

Wie Quelltext kann die Architektur eines Softwaresystems basierend auf der Erfahrung und dem Wissen von Reviewern bewertet werden. Die Güte dieser erfahrungsbasierten Architekturbewertung hängt primär von den Erfahrungen, Kenntnissen und Fähigkeiten dieser Reviewer ab. Der Einsatz dieser Verfahren ist daher insbesondere bei begrenztem Risiko interessant, d. h. für Systemteile oder Reviewziele, die nicht geschäftskritisch sind.

Der Aufwand für die Durchführung erfahrungsbasierter Bewertungsverfahren reicht von leichtgewichtigen, informellen bis zu aufwendigen, formalen Prozessen. Erstere erlauben auch die Bewertung informell oder unvollständig dokumentierter Architekturen (Buchgeher &

Weinreich 2013: 164 f.). Da die Durchführung erfahrungsbasierter Bewertungsverfahren oft weniger Ressourcen benötigt als die Durchführung szenariobasierter Bewertungsverfahren, werden sie häufiger eingesetzt (Eeles et al. 2014: 12; Koziol 2011: 10; Babar & Gorton 2009: 27).

Während des Reviews sammeln der oder die Reviewer potenzielle Probleme. Im Anschluss wird geprüft, ob es sich um tatsächliche Probleme handelt. Für tatsächliche Probleme entscheiden der oder die Reviewer, ob sie behoben oder akzeptiert werden.

Um Architekturreviews erfolgreich einzusetzen, ist ihre Akzeptanz durch das Entwicklungsteam notwendig. Dazu ist es wesentlich, keine einzelnen Personen zu bewerten. Anderenfalls vermeiden Entwickler das Review oder führen „Probe-Reviews“ durch (Masak 2010: 44). Der Gesamtaufwand steigt.

4. METRIKBASIERTE BEWERTUNGSVERFAHREN

Metrikbasierte Bewertungsverfahren sind quantitative Methoden, die für den höchstmöglichen Nutzen eine manuelle Interpretation und Bewertung erfordern. Die Metriken selbst liefern stets nur Hinweise auf den Grad der Erfüllung eines Qualitätsmerkmals. Eine automatisierte Annäherung an die Bewertung dieser Metriken ist durch die Nutzung von Erfahrungswerten aus der Analyse vieler Projekte möglich. Dabei ist es notwendig, geeignete Metriken für jedes Qualitätsmerkmal auszuwählen, das betrachtet werden soll. Dies kann bei einer großen Anzahl von Metriken zu hohem Aufwand und Kosten führen. Daher werden solche quantitativen Verfahren zur Architekturbewertung meist nur für die Bewertung und Überprüfung der wichtigsten Qualitätsanforderungen eingesetzt (Eeles et al. 2014: 10).

Ein prototypisches, metrikbasiertes Verfahren ist das Architecture Quality Assessment (kurz: AQA) von Hilliard et al. 1997. Trotz seines prototypischen Charakters wurde seine grundsätzliche Anwendbarkeit im Rahmen einiger Projekte gezeigt (Eicker et al. 2007: 8). AQA definiert eigene Qualitätsmerkmale, denen insgesamt 200 Maße zugeordnet sind. Die Mehrzahl dieser Maße lässt sich auch auf das Qualitätsmerkmal Wartbarkeit nach ISO/IEC-Norm 25010 abbilden. Sie bewerten die Anforderungserfüllung von „ideal“ bis „nicht zutreffend“ (Thiel 2005). Der Durchführungsaufwand für AQA ist vergleichsweise hoch. Der primäre Grund sind die vielen Maße, die zu ermitteln und in die Bewertung einzubeziehen sind (Eicker et al. 2007: 11; Hilliard et al. 1997: 4 f.).

Auch das Software Architecture Evaluation Model (SAEM) basiert primär auf Metriken der Softwaretechnik und formuliert keine Qualitätsmerkmal-Szenarios. SAEM bewertet die Systemqualität, indem es interne und externe Qualitätsmerkmale des ISO/IEC-Standards 9126⁵ betrachtet. Zu den berücksichtigten Metriken gehören auch Indikatoren zu Aspekten der Wartbarkeit wie Modularität, Komplexität, Kopplung oder Kohäsion. Die Basis der Bewertung bilden die durch den Standard definierten Metriken, die meist auf Ebene des Quelltextes definiert sind. Die Ausprägungen der Metriken werden durch Experten interpretiert. Eine – ressourcenintensive – Einbindung weiterer Stakeholder ist nicht vorgesehen (Dobrica & Niemela 2002: 647). Zum Reifegrad des SAEM und seiner Validierung werden keine Angaben gemacht (Eicker et al. 2007: 8; Dobrica & Niemela 2002: 648).

⁵ Im Jahr 2011 abgelöst durch die ISO/IEC-Norm 25010

5. SZENARIOBASIERTE BEWERTUNGSVERFAHREN

In szenariobasierten Bewertungsverfahren erlauben es Szenarios, überprüfbare Qualitätsanforderungen an ein Softwaresystem zu formulieren (Malich 2008: 74; Lassing 2002: 30). Diese Idee zur szenariobasierten Analyse publiziert Parnas (1972) als erster, verwendet jedoch den Begriff „Szenario“ noch nicht (Bengtsson et al. 2004: 132). Diesen führen Bass et al. ein, als sie „Qualitätsmerkmal-Szenarios“ beschreiben (Bass et al. 2013: 67).

Die Szenarios zu Qualitätsanforderungen an das Softwaresystem bilden die Basis der Analyse. Einige Verfahren beziehen Stakeholder außerhalb des Entwicklungsteams ein, um diese Szenarios zu erheben und zu priorisieren. Die Reviewer nutzen die Szenarios, um die Erfüllung der Anforderungen in einem Walkthrough zu prüfen. Diese Prüfung dauert üblicherweise ein bis zwei Tage (Buchgeher & Weinreich 2013: 165).

Die Nutzung von Szenarios wird in der Software Architecture Analysis Method (SAAM) eingeführt. Sie ist die erste einer Reihe von Methoden, die am Software Engineering Institute der Carnegie Mellon University zur Evaluation von Softwarearchitekturen entwickelt wurden. Viele szenariobasierte Bewertungsverfahren basieren direkt oder indirekt auf der SAAM und spezialisieren oder ergänzen die Methode.

Der folgende Abschnitt 5.1 beschreibt das grundlegende Konzept der szenariobasierten Architekturbewertung. Die Abschnitte 5.2 und 5.3 beschreiben zwei Bewertungsverfahren näher, die aktiv in verschiedenen Kontexten eingesetzt werden (Koziolek 2011: 6).

5.1 Herangehensweise im szenariobasierten Ansatz

Vor Beginn des eigentlichen Bewertungsprozesses werden das eingesetzte Verfahren, die Qualitätsanforderungen an das Softwaresystem sowie dessen Sollarchitektur vorgestellt, um einen einheitlichen Wissensstand aller Reviewer sicherzustellen.

Im ersten Schritt zur Architekturbewertung formuliert und priorisiert eine Gruppe von Stakeholdern Qualitätsmerkmal-Szenarios. Diese Szenarios beschreiben nach Kazman et al. (2000a: 6) die Interaktion von Stakeholdern mit einem Softwaresystem. Anders als bei Use Cases inkludiert dies nicht nur Interaktionen zur Laufzeit, sondern beispielsweise auch die Anpassung des Systems. Szenarios operationalisieren die Anforderungen an ein oder mehrere Qualitätsmerkmale eines Softwaresystems. In einem Walkthrough analysieren und bewerten die Reviewer, wie flexibel die Architektur hinsichtlich der Szenariorealisation ist. Die Ergebnisse werden präsentiert. Abhängig vom Ziel der Architekturbewertung können weitere Schritte wie die Auswahl einer Architekturalternative durchgeführt werden. Eine übliche Dauer derartiger Bewertungsworkshops ist ein bis zwei Tage (Buchgeher & Weinreich 2013: 165).

5.2 Architecture Tradeoff Analysis Method (ATAM)

Die Architecture Tradeoff Analysis Method (kurz: ATAM) ermöglicht die gleichzeitige Bewertung einer Sollarchitektur hinsichtlich mehrerer Qualitätsmerkmale. Nicht bewertet werden die Anforderungen oder die Implementierung eines Softwaresystems. Zwischen den Qualitätsmerkmalen erfolgt eine Abwägung in Abhängigkeit der Geschäftsziele, indem ATAM die Auswirkung von Entwurfsentscheidungen auf die Qualitätsmerkmale untersucht und so Wechselwirkungen zwischen ihnen aufzeigt. (Silva 2014: 19; Grunske 2008: 282; Eicker et al. 2007: 3; Ionita et al. 2002: 5 f.). Die zu berücksichtigenden Qualitätsmerkmale leiten sich aus den Zielen der Stakeholder ab (Babar & Gorton 2004).

ATAM beschreibt vier Phasen, die jeweils in mehrere Schritte unterteilt sein können (Clements et al. 2002: 44 ff.):

- Phase 1: Wie in szenariobasierten Verfahren üblich, sehen die ersten drei Schritte der ATAM die Vorstellung des Verfahrens, der Einflussfaktoren und Geschäftsziele sowie der Softwarearchitektur vor.
- Phase 2: In den folgenden Schritten vier bis sechs identifizieren und analysieren die Reviewer Merkmale und Ansätze der Sollarchitektur, die die Erfüllung der Qualitätsanforderungen beeinflussen können. Diese Qualitätsanforderungen werden mithilfe eines Quality Attribute Utility Tree zu konkreten, messbaren Szenarios verfeinert.
- Phase 3: In Schritt sieben wird ein Brainstorming mit den Stakeholdern des Softwaresystems durchgeführt, um weitere Szenarios zu erheben. Die Stakeholder nehmen auch eine Priorisierung der Szenarios vor. Die im vierten Schritt identifizierten Merkmale und Ansätze der Architektur werden hinsichtlich der Erfüllung der ergänzten Szenarios analysiert (Schritt acht).
- Phase 4: Abschließend präsentiert das Review-Team die Ergebnisse der Architekturbewertung den Stakeholdern:
- Tradeoffs, d. h. Entscheidungen, die eine Abwägung konkurrierender Qualitätsmerkmale erfordern.
 - Risikobereiche der Sollarchitektur, die für Qualitätsmängel wie Architekturerosion besonders anfällig sind.
 - Sensitivitätspunkte der Sollarchitektur, die für die Erfüllung bestimmter Qualitätsanforderungen entscheidend sind.

Weitere Effekte können eine verbesserte Dokumentation der Architektur sowie der Qualitätsanforderungen sein, die durch die Szenarios operationalisiert werden. Der Erfüllung dieser Qualitätsanforderungen werden durch die Analyse Architekturmerkmale und -entscheidungen zugeordnet.

ATAM wird üblicherweise frühzeitig eingesetzt, um die geplante (Soll-)Architektur hinsichtlich ihrer Eignung zu untersuchen, die Qualitätsanforderungen an das Softwaresystem zu erfüllen. Dennoch ist der Einsatz von ATAM auch für eine Analyse bereits implementierter Systeme möglich (Babar & Gorton 2004: 1534; Clements et al. 2002: 43; Dobrica & Niemela 2002: 649). Clements et al. (2002: 39 ff.) nennen eine Mindestanzahl von vier Reviewern zur Durchführung des Verfahrens. Oft sind jedoch mehr Personen beteiligt. Hinzu kommt der Vor- und Nachbereitungsaufwand, sodass die Anwendung von ATAM einen vergleichsweise hohen Ressourcenbedarf verursacht (Eloranta et al. 2014: 159). In der Literatur finden sich Angaben von bis zu 70 Personentagen schon für die Anwendungen des Verfahrens in mittelgroßen Projekten (Clements et al. 2002: 39; Kazman et al. 2000b: 7). Dieser wird durch den kombinierten Einsatz von Analyse- und Befragungstechniken verursacht (Eicker et al. 2007: 10; Babar & Gorton 2004: 1534), die der Erhebung und Priorisierung der Szenarios dienen.

ATAM wurde wie SAAM an der Carnegie Mellon Universität entwickelt (Kazman et al. 2000b; Kazman et al. 1998). SAAM bildet einen integralen Bestandteil der ATAM. (Kazman et al. 2000a: 5) ATAM gilt als ausgereift und wurde in vielen Projekten validiert (Koziolek 2011: 7; Zalewski 2007: 97).

5.3 Architecture-Level Modifiability Analysis (ALMA)

Die Architecture-Level Modifiability Analysis (kurz: ALMA) gilt als ausgereifte und gut validierte Spezialisierung der SAAM. Ihre Anwendung ist durch verschiedene Fallstudien detailliert dokumentiert. Sie nutzt die Sollarchitektur zur Bewertung der Modifizierbarkeit. Es ist nicht möglich, weitere, kontextabhängig relevante Qualitätsmerkmale in die Bewertung einzubeziehen. Dies kann bei konkurrierenden Qualitätsmerkmalen zu Verschlechterungen der Gesamtqualität führen, wenn auf Basis der Analyseergebnisse Verbesserungsmaßnahmen eingeleitet werden. ALMA wird wie ATAM typischerweise vor Beginn der Implementierung angewendet (Bengtsson et al. 2000: 4), kann aber auch in Reengineering-Projekten für Altsysteme eingesetzt werden (Babar & Gorton 2004: 1533).

ALMA unterstützt drei mögliche Ziele (Bengtsson et al. 2000: 2, 7; Bengtsson 2002: 57 f.; Lassing et al. 2002: 48 f.; Babar & Gorton 2004: 1533):

1. Prognose der Wartungskosten: Die Aufwände zur Realisierung von Änderungsszenarios werden geschätzt. Diese Schätzung bildet die Basis für eine Prognose der Wartungskosten.
2. Identifikation von Architekturrisiken: Wie durch ATAM werden Risikobereiche der Architektur identifiziert. In diesen Bereichen ist das Softwaresystem nicht flexibel, d. h. es kann nur mit hohem Aufwand angepasst werden.
3. Vergleich alternativer Architekturen: Es wird analysiert, inwiefern verschiedene Architekturkandidaten bzw. alternative Architekturentscheidungen die Realisierung der Änderungsszenarios unterstützen bzw. behindern.

Die Auswahl der Änderungsszenarios basiert auf Stakeholder-Interviews. Anhand der verschiedenen Stakeholder-Perspektiven werden relevante Szenarios identifiziert. Relevant sind Szenarios, die mit relativ hoher Wahrscheinlichkeit eintreten und neuartige Änderungen beschreiben. Lassing et al. (2002: 50) wenden jedoch ein, dass die Einbeziehung der Stakeholder einen hohen Ressourcenbedarf verursacht. Die Auswirkung der betrachteten Architektur bzw. Architekturentscheidung wird hinsichtlich ihrer Unterstützung bzw. Behinderung der Änderungsszenarios diskutiert und bewertet. Insbesondere wird der sogenannte ripple effect analysiert, der die Auswirkungen einer Moduländerung auf andere Module beschreibt (Grunske 2008: 285). Die Bewertung kann durch die Bestimmung einer absoluten, aus dem Änderungsaufwand abgeleiteten Zahl oder qualitativ anhand einer Ordinalskala geschehen. Nach Bengtsson (2002: 61) und Lassing (2002: 109) ist eine qualitative Bewertung auch in beschreibender Form möglich. Beschreibungen ermöglichen zwar genauere Angaben, sind jedoch schlechter vergleichbar als Werte der Ordinalskala.

Bengtsson et al. (2004: 133) beschreiben fünf Schritte von ALMA:

1. Zielsetzung: Auswahl des Analyseziels
2. Beschreibung der Softwarearchitektur: Geeignete Darstellung der für die Analyse relevanten Aspekte der Sollarchitektur
3. Szenarioerhebung: Nutzung von Stakeholderinterviews zur Identifikation und Priorisierung von Änderungsszenarios
4. Evaluation: Analyse des Effekts der Änderungsszenarios auf die betrachtete Sollarchitektur
5. Interpretation der Ergebnisse: Schlussfolgerungen aus den Analyseergebnissen ziehen, Ergebnisse bewerten

5.4 Vergleich von ATAM und ALMA

Nach Koziol (2011: 6) sind ATAM und ALMA die einzigen szenariobasierten Bewertungsverfahren für Softwarearchitekturen, die noch aktiv eingesetzt werden und zu denen industrielle Fallstudien verfügbar sind. Insbesondere auf ATAM wird in der Literatur häufig verwiesen, beispielsweise in Knodel & Naab (2014: 115); Zhu et al. (2014: 153, 159); Silva (2014: 19); Zalewski (2007: 92); Babar & Gorton (2004); Babar et al. (2004: 11); Ionita et al. (2002: 1). Tabelle 1 stellt die vorgestellten Verfahren zur Architekturbewertung vergleichend gegenüber:

KRITERIEN	ATAM	ALMA
SPEZIFIKATION VON ÄNDERUNGSSZENARIOS	Template für Qualitätsmerkmal-Szenarios	Top-Down- und Bottom-Up-Methode zur Szenario-Formulierung; kein Template
BEWERTUNG UND AUSWEITUNGSANALYSE	Erfahrungsbasiert (manuell)	Erfahrungsbasiert (manuell)
BERÜCKSICHTIGUNG WEITERER QUALITÄTSMERKMALE	Flexible Auswahl und Abwägung von Qualitätsmerkmalen	Keine Unterstützung
ARCHITEKTUR-BESCHREIBUNG	Prozess-, Datenfluss-, Nutzungs-, physische und Modulsicht	Beliebig
UNTERSTÜTZUNG DER BEWERTUNG VON ALTSYSTEMEN	Bewertung möglich, keine explizite Unterstützung	Bewertung möglich, keine explizite Unterstützung
VALIDIERUNG	Über 20 industrielle Fallstudien	7 industrielle Fallstudien
AUFWAND	30 bis 70 Personentage	Aufwendig, allerdings meist leichtgewichtiger als ATAM

Tabelle 1: Vergleich der Verfahren ATAM und ALMA (in Anlehnung an Koziolk 2011: 8)

6. VERGLEICH DER VERFAHREN

KRITERIUM	METRIKBASIIERT	ERFAHRUNGSBASIIERT	SCENARIOBASIIERT
AUFWAND	Eher gering	Gering bis hoch	Hoch
QUALIFIKATION DER PRÜFER	Mittel	Hoch	Hoch
OBJEKTIVIERBARKEIT DER ERGEBNISSE	Eher hoch	Gering, abhängig von Größe der Expertengruppe	Mittel, abhängig von Güte des Szenarios und Größe der Expertengruppe
IMPLEMENTIERUNG VORAUSGESETZT	Ja	Nein	Nein
VERFÜGBARKEIT VON WERKZEUGEN	Mehrere verfügbar, zur Berechnung der Metriken	Spezifische Werkzeugunterstützung schwierig	Einige verfügbar, zur Formulierung von Szenarien
BETEILIGTE STAKEHOLDER	Reviewer	Reviewer/Experten	Reviewer, Nutzer, Auftraggeber, ggf. weitere Personen

Tabelle 2: Einschätzung der Bewertungsverfahren

Die Beschreibungen der Verfahren in den letzten Abschnitten zeigen, dass es keine Methode gibt, die gegenüber den beiden anderen zu bevorzugen ist. Stattdessen besitzen alle Verfahren individuelle Stärken und Schwächen. Die Auswahl des einzusetzenden Verfahrens für eine konkrete Anwendung sollte daher wohl überlegt sein.

In Tabelle 2 findet sich ein Vergleich der unterschiedlichen Arten von Bewertungsverfahren anhand wichtiger Kriterien, die von den Autoren aus der Analyse der einzelnen Bewertungsverfahren abgeleitet wurden.

Der zeitliche und finanzielle Aufwand für die Durchführung von Architekturbewertungen kann sehr unterschiedlich sein. Generell lässt sich aber festhalten, dass der Aufwand bei metrikbasierten Ansätzen geringer als bei erfahrungsbasierten und szenariobasierten ist. Das Erstellen und Prüfen von Szenarios ist in der Regel mit einem hohen Aufwand verbunden. Erfahrungsbasierte Verfahren können mit geringem Aufwand als informelle Reviews durchgeführt werden. Sie können aber auch einen hohen Aufwand verursachen, wenn mehrere Diskussionsrunden mit Experten durchgeführt werden.

Metrikbasierte Verfahren stellen an die Qualifikation der Prüfer geringere Anforderungen als die anderen Verfahren. Die Ergebnisse bei erfahrungsbasierten Verfahren sind in hohem Maße vom Wissen und der Erfahrung der Experten abhängig. Auch das Erstellen geeigneter Szenarios erfordert eine hohe Qualifikation.

Die Objektivierbarkeit der Ergebnisse szenariobasierter Verfahren ist wesentlich von der Güte des Szenarios abhängig. Genauso wie bei erfahrungsbasierten Ansätzen hilft eine größere Expertengruppe, zu objektiveren Ergebnissen zu kommen. Bei metrikbasierten Ansätzen kann es subjektive Einflüsse bei der Interpretation der berechneten Metriken geben; die Metriken als solche sind objektiv vergleichbar.

Ein großer Vorteil von erfahrungsbasierten und szenariobasierten Verfahren liegt darin, dass sie einsetzbar sind, wenn nur die Sollarchitektur, aber noch keine Implementierung vorliegt. Dadurch sind auch was-wäre-wenn-Untersuchungen recht leicht möglich. Bei metrikbasierten Verfahren können nur konkrete Implementierungen geprüft werden, sodass sie erst in späteren Phasen des Entwicklungsprozesses zur Anwendung kommen können. Bei allen Verfahren ist eine Werkzeugunterstützung zur Erleichterung der Arbeit sinnvoll. Für erfahrungsbasierte Verfahren ist eine spezifische Werkzeugunterstützung schwierig, bei metrikbasierten Voraussetzung für einen praktischen Einsatz der Verfahren. Für szenariobasierte Verfahren sind zur Unterstützung der einzelnen Methoden Werkzeuge verfügbar. Dort sind neben den eigentlichen Reviewern und Experten, zu denen insbesondere Softwarearchitekten zählen, je nach Methode auch weitere Stakeholder beteiligt, die Szenarios erheben und Priorisierungen vornehmen.

Nach Möglichkeit sollte nicht ein Verfahren allein eingesetzt, sondern eine geeignete Kombination der Verfahren gewählt werden. Für bereits implementierte Systeme bietet es sich insbesondere an, ein qualitatives Verfahren mit einer quantitativen Analyse mit Metriken zu verbinden. Der unterstützende Einsatz quantitativer Analysen ermöglicht eine Fokussierung der qualitativen Bewertung auf verdächtige bzw. risikobehaftete Bereiche der Softwarearchitektur und steigert die Objektivität der Ergebnisse.

Der Bewertung einer Softwarearchitektur sollte selbstverständlich die Behebung der gefundenen Schwachstellen folgen, die unterschiedlich umfangreich ausfallen kann. Sinnvoll ist in vielen Fällen nicht nur eine erneute Überprüfung, sondern auch die Etablierung eines regelmäßigen Bewertungsprozesses. Bei der architekturzentrierten oder architekturbasierten Softwareentwicklung (Scott 2002; Bass & Kazman 1999), bei der die Softwarearchitektur im Mittelpunkt aller Aktivitäten steht, ist dies ein integraler Bestandteil des Prozesses.

7. ZUSAMMENFASSUNG UND AUSBLICK

Die Softwarearchitektur ist ein wesentliches Artefakt der Softwareentwicklung, die eine Anwendung beschreibt und ihre Entwicklung, Wartung und Weiterentwicklung begleitet. Bewertungsverfahren für Softwarearchitektur helfen dabei, die Architektur für die Erfüllung der Qualitätsanforderungen an das System zu optimieren. So kann z. B. Softwareerosion reduziert werden. Bewertungsverfahren lassen sich in erfahrungsbasiert, szenariobasiert und metrikbasiert unterscheiden, die jeweils verschiedene Vor- und Nachteile aufweisen.

In der unternehmerischen Praxis ist der Aufwand der Verfahren meist das wesentliche Auswahl-Kriterium. Es überwiegt die Anwendung erfahrungsbasierter Verfahren, die oft informell durchgeführt werden können. Es bleibt festzuhalten, dass Architekturbewertungen in der Praxis nicht so häufig eingesetzt werden, wie es sinnvoll wäre (Knodel & Naab 2014: 115; Babar & Gorton 2009: 28, 31).

Ein Grund ist darin zu sehen, dass sich der Nutzen einer Architekturbewertung oft nicht sofort und direkt und damit für Auftraggeber kalkulierbar zeigt, sondern eher mittel- und langfristige Auswirkungen hat. Der Nutzen einer Bewertung ist stark abhängig von der Qualifikation und dem Wissen der Reviewer, insbesondere bei erfahrungsbasierten Verfahren. Daher ist die Kosten-/Nutzen-Relation schwer berechenbar.

Die zur Softwareentwicklung eingesetzte Vorgehensmethodik beeinflusst den Aufwand. Bei einem herkömmlichen, plangetriebenen Vorgehen, bei dem die Softwarearchitektur größtenteils schon in der Entwurfsphase definiert wird, sind nach einer initialen Bewertung nachfolgende Einsätze der Verfahren deutlich weniger aufwendig. Bei agilen Vorgehensmodellen, bei denen weniger im Vorfeld geplant wird, verschieben sich die Aufwände. Die initiale Bewertung einer anfangs nur grob ausgearbeiteten Softwarearchitektur sollte weniger Aufwand erfordern. Dieser fällt daher bei den Reevaluierungen nach Architekturänderungen in den Entwicklungsiterationen an. Auch die verwendete Technik kann den Aufwand verändern. Es bleibt abzuwarten, wie moderne Architekturstile, wie z. B. Microservices, den Nutzen und den Aufwand von Architekturbewertungen verändern.

Es ist zu hoffen, dass die Sensibilität für die Güte einer Softwarearchitektur in der Praxis der Softwareentwicklung zunimmt und damit auch Architekturbewertungen den Stellenwert erhalten, den sie verdienen. Die Softwareentwickler sollten sich für die von ihnen umgesetzte Softwarearchitektur verantwortlich fühlen und sich aktiv um deren Qualität und Erhalt kümmern.

8. LITERATURVERZEICHNIS

- Babar, M. A.; Gorton, I. (2004): Comparison of Scenario-Based Software Architecture Evaluation Methods, in: 11th Asia-Pacific Software Engineering Conference, S. 600 – 607.
- Babar, M. A.; Gorton, I. (2009): Software Architecture Review. The State of Practice, in: Computer, 42. Jg., Heft 7, S. 26 – 32.
- Babar, M. A.; Zhu, L.; Jeffery, R. (2004): A Framework for Classifying and Comparing Software Architecture Evaluation Methods, in: Proceedings of the 2004 Australian Software Engineering Conference, S. 309 – 318.
- Bass, L.; Clements, P.; Kazman, R. (2013): Software Architecture in Practice, 3. überarbeitete, Upper Saddle River, N.J, Addison-Wesley/Pearson.
- Bass, L.; Kazman, R. (1999): Architecture-Based Development. CMU/SEI-99-TR-007, Pittsburgh, Carnegie Mellon University.
- Bengtsson, P. (2002): Architecture-Level Modifiability Analysis, Dissertation, Schweden.
- Bengtsson, P.; Lassing, N.; Bosch, J.; van Vliet, H. (2000): Analyzing Software Architectures for Modifiability. HK/R-RES-00/11-SE, Ronneby, Högskolan of Karlskrona/Ronneby.
- Bengtsson, P.; Lassing, N.; Bosch, J.; van Vliet, H. (2004): Architecture-level Modifiability Analysis (ALMA), in: Journal of Systems and Software, 69. Jg., Heft 1-2, S. 129 – 147.
- Buchgeher, G.; Weinreich, R. (2013): Continuous Software Architecture Analysis, in: Babar, M. A. (Hrsg.): Agile Software Architecture: Aligning Agile Processes and Software Architectures. Burlington, Elsevier Science; Morgan Kaufmann, S. 161 – 188.
- Clements, P.; Kazman, R.; Klein, M. H. (2002): Evaluating Software Architectures. Methods and Case Studies, Boston, Addison-Wesley.
- Dobrica, L.; Niemela, E. (2002): A Survey on Software Architecture Analysis Methods, in: IEEE Transactions on Software Engineering, 28. Jg., Heft 7, S. 638 – 653.
- Eeles, P.; Bahsoon, R.; Mistrík, I.; Roshandel, R.; Stal, M. (2014): Relating System Quality and Software Architecture. Foundations and Approaches, in: Mistrík, I.; Bahsoon, R.; Eeles, P.; Roshandel, R.; Stal, M. (Hrsg.): Relating System Quality and Software Architecture. Waltham, MA, Morgan Kaufmann, S. 1 – 20.
- Eicker, S.; Hegmanns, C.; Malich, S. (2007): Auswahl von Bewertungsmethoden für Softwarearchitekturen, ICB Research-Report Nr. 14, Essen, Universität Duisburg-Essen.
- Eloranta, V.-P.; van Hesch, U.; Avgeriou, P.; Harrison, N.; Koskimies, K. (2014): Lightweight Evaluation of Software Architecture Decisions, in: Mistrík, I.; Bahsoon, R.; Eeles, P.; Roshandel, R.; Stal, M. (Hrsg.): Relating System Quality and Software Architecture, Waltham, MA, Morgan Kaufmann, S. 157 – 180.
- Grunske, L. (2008): Bewertungstechniken. Eine allgemeine Übersicht, in: Reussner, R.; Hasselbring, W. (Hrsg.): Handbuch der Software-Architektur, Heidelberg, dpunkt.-Verlag, S. 269 – 286.
- Hilliard, R. F.; Kurland, M. J.; Litvintchouk, S. D. (1997): MITRE's Architecture Quality Assessment, in: Proceedings of Software Engineering & Economics Conference, S. 1 – 9.
- Ionita, M. T.; Hammer, D. K.; Obbink, H. (2002): Scenario-Based Software Architecture Evaluation Methods. An Overview, zugegriffen über: <http://www.win.tue.nl/oas/architecting/aimes/papers/Scenario-Based%20SWA%20Evaluation%20Methods.pdf> am 31.01.2015.
- ISO/IEC-Standard. 9126 (2001): Software Engineering - Product Quality, letzter Zugriff am 31.01.2016.
- ISO/IEC-Standard. 25010 (2011): Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuARE) - System and Software Quality Models, letzter Zugriff am 31.01.2016.
- Kazman, R.; Carrière, S. J.; Woods, S. G. (2000a): Toward a Discipline of Scenario-based Architectural Engineering, in: Annals of Software Engineering, 9. Jg., Heft 1/4, S. 5 – 33.

- Kazman, R.; Klein, M. H.; Barbacci, M. R.; Longstaff, T., et al. (1998): The Architecture Tradeoff Analysis Method, CMU/SEI-98-TR-008, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University.
- Kazman, R.; Klein, M. H.; Clements, P. (2000b): ATAM: Method for Architecture Evaluation, CMU/SEI-2000-TR-004, Pittsburgh, Software Engineering Institute, Carnegie Mellon University.
- Knodel, J.; Naab, M. (2014): Software Architecture Evaluation in Practice. Retrospective on More Than 50 Architecture Evaluations in Industry, in: 2014 IEEE/IFIP Conference on Software Architecture (WICSA), IEEE Computer Society, S. 115 – 124.
- Koziolek, H. (2011): Sustainability Evaluation of Software Architectures. A Systematic Review, in: Crnkovic, I.; Stafford, J. A.; Petriu, D.; Happe, J.; Inverardi, P. (Hrsg.): Proceedings of the 7th International Conference on the Quality of Software Architectures (QoSA'11) and 2nd International Symposium on Architecting Critical Systems (ISARCS'11), New York, NY, USA, ACM, S. 3 – 12.
- Lassing, N. (2002): Architecture-Level Modifiability Analysis, Dissertation, Amsterdam.
- Lassing, N.; Bengtsson, P.; van Vliet, H.; Bosch, J. (2002): Experiences with ALMA. Architecture-Level Modifiability Analysis, in: Journal of Systems and Software, 61. Jg., Heft 1, S. 47 – 57.
- Malich, S. (2008): Qualität von Softwaresystemen. Ein pattern-basiertes Wissensmodell zur Unterstützung des Entwurfs und der Bewertung von Softwarearchitekturen, Wiesbaden, Betriebswirtschaftlicher Verlag Dr. Th. Gabler / GWV Fachverlage GmbH, Wiesbaden.
- Masak, D. (2010): Der Architekturreview. Vorgehensweise, Konzepte und Praktiken, Berlin, Springer.
- Parnas, D. L. (1972): On the Criteria To Be Used in Decomposing Systems into Modules, in: Communications of the ACM, 15. Jg., Heft 12, S. 1053 – 1058.
- Perry, D. E.; Wolf, A. L. (1992): Foundations for the Study of Software Architecture, in: ACM SIGSOFT Software Engineering Notes, 17. Jg., Heft 4, S. 40 – 52.
- Reussner, R.; Hasselbring, W. (Hrsg.) (2008): Handbuch der Software-Architektur, 2., überarbeitete und erweiterte Auflage, Heidelberg, dpunkt.Verlag.
- Scott, K. (2002): The Unified Process Explained, Boston, Addison-Wesley.
- Silva, L. R. de (2014): Towards Controlling Software Architecture Erosion Through Runtime Conformance Monitoring, Dissertation, St. Andrews.
- Smolander, K. (2002): Four Metaphors of Architecture in Software Organizations. Finding Out the Meaning of Architecture in Practice, in: 2002 International Symposium on Empirical Software Engineering. Los Alamitos, Calif. [u.a.], IEEE Computer Society, S. 211 – 221.
- Thiel, S. (2005): A Framework to Improve the Architecture Quality of Software Intensive Systems.
- Zalewski, A. (2007): Beyond ATAM. Architecture Analysis in the Development of Large Scale Software Systems, in: Oquendo, F. (Hrsg.): Proceedings of the 1st European Conference on Software Architecture. Berlin, Heidelberg, Springer, S. 92 – 105.
- Zhu, H.; Zhang, Q.; Zhang, Y. (2014): HASARD. A Model-Based Method for Quality Analysis of Software Architecture, in: Mistrik, I.; Bahsoon, R.; Eeles, P.; Roshandel, R.; Stal, M. (Hrsg.): Relating System Quality and Software Architecture. Waltham, MA, Morgan Kaufmann, S. 123 – 156.

EINSATZSZENARIEN VON ADOBE CONNECT AN DER NORDAKADEMIE



Prof. Dr. Cristina Trujillo
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Simon Hachenberg, M. A.
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Faria Essenwanger
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: Adobe Connect ist ein flexibel einsetzbares Kollaborationswerkzeug. Der Schwerpunkt liegt im direkten gegenseitigen Online-Austausch. An der NORDAKADEMIE wird dieses Werkzeug für verschiedene Szenarien eingesetzt. Dieser Artikel stellt mehrere dieser Szenarien vor. Um die Nutzung auch sehen zu können, sind zu drei Szenarien die Aufzeichnungen integriert. Diese können in der E-Paper-Ausgabe aufgerufen werden.

Keywords: Virtuelles Klassenzimmer, E-Learning-Didaktik, Webinare

1. EINFÜHRUNG

Die NORDAKADEMIE arbeitet bereits seit mehreren Jahren in Lehrveranstaltungen mit Adobe Connect und trägt mit diesen Erfahrungen zum wissenschaftlichen Diskurs bei. Aufbauend auf dem Erfahrungsbericht von Trujillo und Essenwanger (2013: 445 ff.; siehe Abbildung 1), in denen die Grundlagen von sowie der Einstieg in Adobe Connect und Einsatzszenarien im Fachgebiet Spanisch beschrieben werden, stellt dieser Artikel weitere Einsatzszenarien von Adobe Connect an der NORDAKADEMIE vor und erläutert sie.

Adobe Connect ist ein Kollaborationswerkzeug, mit dem virtuelle Veranstaltungen eingerichtet und durchgeführt werden können. Die gängige Bezeichnung von Angeboten dieser Art lautet „virtuelles Klassenzimmer“ (vgl. Häfele & Maier-Häfele 2012: 33; Jankowski et al. 2012: 17). Der Zugang kann über die Lernplattform Moodle erfolgen. Hierbei wird die Verbindung zwischen Moodle und Adobe Connect über ein Plug-in hergestellt, welches vom Deutschen Forschungsnetz (DFN) entwickelt wurde. Die eingerichteten Räume werden über ein zentrales Konto verwaltet. Zudem können sich Mitarbeiter der NORDAKADEMIE ein eigenes Konto beim DFN anlegen lassen und virtuelle Klassenzimmer eigenständig erstellen und administrieren.

Prof. Dr. Cristina Trujillo ist seit 2010 Professorin für Wirtschaftsspanisch und Leiterin des Fachgebiets Spanisch an der NORDAKADEMIE. Sie studierte Hispanistik an der Universidad Complutense Madrid und promovierte an der Universität Hamburg. Ihre Forschungsaktivitäten liegen im Bereich des Blended Learnings. Als Doktorin der Philosophie ist sie zudem Lehrerfortbilderin in der Phonetik und der Affektivität am Instituto Cervantes.
E-Mail: cristina.trujillo@nordakademie.de

Simon Hachenberg, M. A. ist seit Juni 2015 wissenschaftlicher Mitarbeiter und E-Learning-Beauftragter an der NORDAKADEMIE sowie Mitglied der Forschungsgruppe E-Learning. Vor seiner Anstellung an der NORDAKADEMIE entwickelte er an der Fachhochschule Flensburg im Rahmen des eBusiness-Lotsen Schleswig-Holstein Blended-Learning-Kurse zu logistischen Themen.
E-Mail: simon.hachenberg@nordakademie.de

Faria Essenwanger ist nach einigen Jahren Tätigkeit als Rechtsanwältin in Peru seit 2006 an der NORDAKADEMIE als Dozentin für Spanisch und interkulturelles Management tätig. Ihr Forschungsschwerpunkt liegt im Bereich des Blended Learning.
E-Mail: faria.essenwanger@nordakademie.de

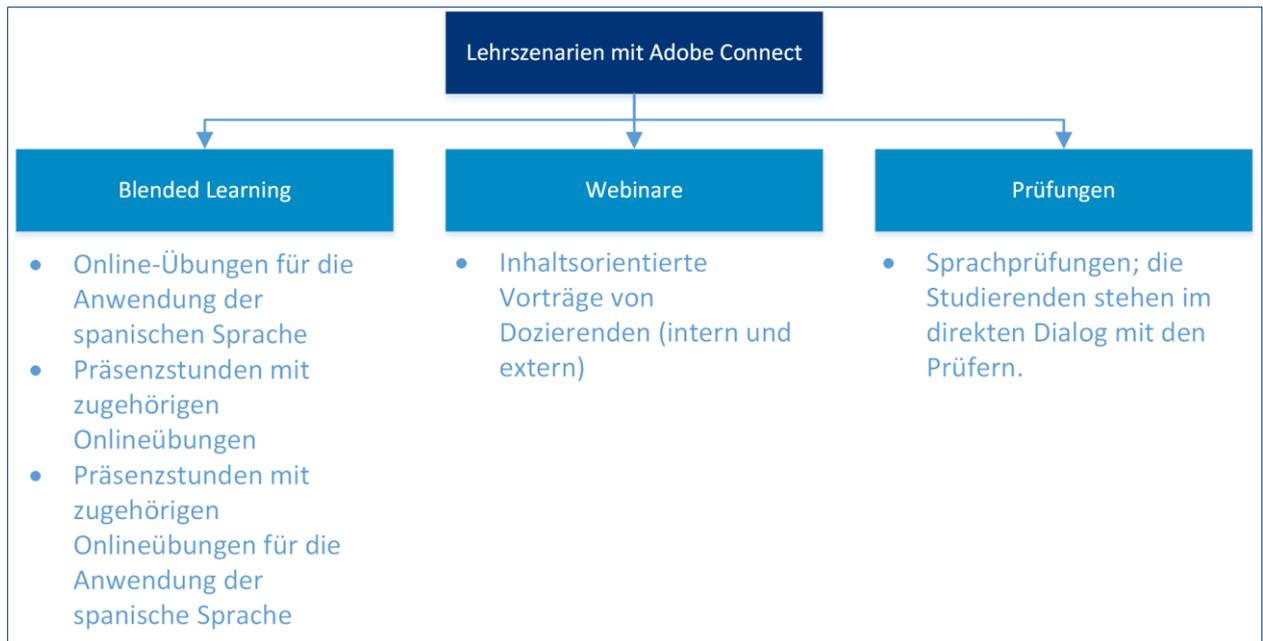


Abbildung 1: Lehrszenerarien mit Adobe Connect des Fachgebiets Spanisch

Der Austausch erfolgt in Adobe Connect über schriftliche, mündliche oder bildliche synchrone Kommunikationsformen (siehe Abbildung 2). Alle Kommunikationsformen sind kombinierbar, sodass Mischformen der Kommunikation entstehen. Beispielsweise ergibt die Kombination von Kamera und Ton einen ganzheitlichen Eindruck des Sprechenden.

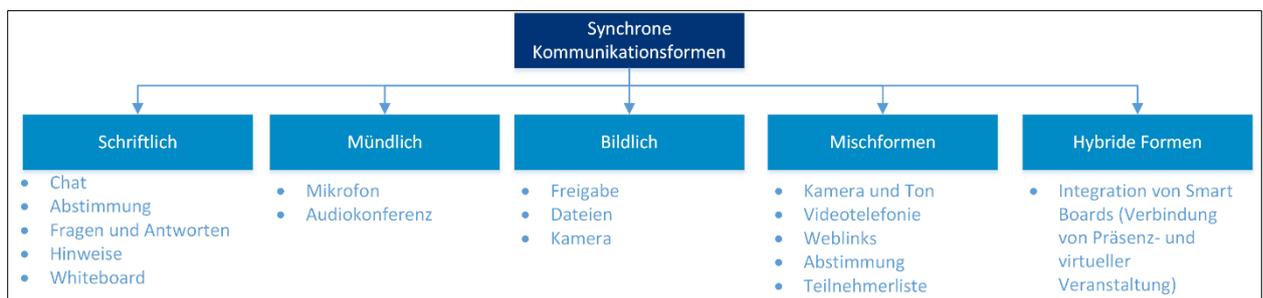


Abbildung 2: Kommunikationsformen in Adobe Connect

Rolle	Rechte synchroner Kommunikationsformen	Integrationsniveau
Teilnehmer <i>Primär Empfänger von Informationen</i>	Schriftlich Chat, Abstimmung, Fragen und Antworten, Hinweise Mündlich <ul style="list-style-type: none"> • Mikrofon (sofern freigegeben) • Audiokonferenz (sofern eingerichtet) 	Gering bis mittel
Moderator <i>Primär Sender von Informationen</i>	Wie Teilnehmer Abweichung: Mikrofonnutzung ist standardmäßig möglich + Bildlich <ul style="list-style-type: none"> • Kamera • Freigabe 	Mittel bis hoch

Veranstalter <i>Administrator des Raumes</i>	Wie Moderator Kann zusätzlich Layouts und Werkzeuge (Pods) festlegen/verändern	Hoch
---	---	------

Tabelle 1: Zusammenhang von Rollen, Kommunikationsmöglichkeiten und Integrationsniveau

Durch die vielseitigen Funktionen ermöglicht Adobe Connect auch hybride Lösungen, sodass Werkzeuge des Präsenzunterrichts in Adobe Connect eingebunden werden können. So ist es möglich, das SMART Board, das im Grunde eine elektronische Wandtafel darstellt, entweder als Projektor oder interaktiv als Schreibtafel einzubinden. Teilnehmer können in Abhängigkeit vom didaktischen Konzept unterschiedlich in ein Adobe Connect-Meeting integriert werden. Je nach Integrationsgrad benötigen die anwesenden Personen entsprechende Rechte. Reicht es didaktisch aus, dass die Teilnehmer Informationen nur konsumieren, ist die einfache Rolle als Teilnehmer ausreichend. Sollen Teilnehmer eigene Beiträge einbringen (schriftlich, mündlich, bildlich), sollten Moderationsrechte vergeben werden (siehe Tabelle 1).

Neben dem Integrationsniveau gilt es Interaktionsmöglichkeiten in Abhängigkeit von der Teilnehmerzahl und der technischen Ausstattung zu berücksichtigen (siehe Tabelle 2). Grundsätzlich benötigen alle Teilnehmer ein Endgerät (Laptop, stationärer Computer, mobile Endgeräte), eine Internetverbindung, einen Webbrowser und den Adobe Flash Player 11.2+. Weitere technische Ausstattung wird in Abhängigkeit von der Interaktionsmöglichkeit benötigt (vgl. Seifert & Kerschbaumer 2012: 36 ff.).

Teilnehmerzahl	Interaktionsmöglichkeiten	Technische Ausstattung
Bis 10 Teilnehmer (Erfahrungsgemäß liegt die optimale Teilnehmerzahl bei 6 Personen)	<ul style="list-style-type: none"> • Verbales Feedback • Mündliche Diskussionen • Chat • Gemeinsame Whiteboards • Application Sharing 	<ul style="list-style-type: none"> • Endgeräte <ul style="list-style-type: none"> ○ Stationär: PC, Laptop ○ Mobil: Tablet, Smartphone • Mikrofon/Headset • Tastatur • Lautsprecher/ Kopfhörer
11 – 25 Teilnehmer	<ul style="list-style-type: none"> • Mündliche Wortmeldungen • Chat • Blitzlichttechnik • Abstimmung 	<ul style="list-style-type: none"> • Endgeräte <ul style="list-style-type: none"> ○ Stationär: PC, Laptop ○ Mobil: Tablet, Smartphone • Mikrofon/Headset • Tastatur • Lautsprecher/ Kopfhörer
Ab 26 Teilnehmer	<ul style="list-style-type: none"> • Präsentation • Abstimmung/Umfragen • Fragen 	<ul style="list-style-type: none"> • Endgeräte <ul style="list-style-type: none"> ○ Stationär: PC, Laptop ○ Mobil: Tablet, Smartphone • Tastatur • Lautsprecher/Kopfhörer

Tabelle 2: Interaktionsmöglichkeiten in Abhängigkeit von der Gruppengröße

Dozierende – und Studierende mit Moderatorenrechten – benötigen zusätzlich eine Webkamera für die Bildübertragung, ein Headset, damit die Sprachübertragung und Tonausgabe ohne Rückkopplungen erfolgen können, und das Adobe Connect Add-in, welches erforderlich ist, um den Bildschirm und/oder Dateien auf dem Computer freigeben zu können. (Die Aufforderung und Installationsroutine startet in diesem Fall automatisch).

Ein wesentlicher Vorteil der Arbeit mit Adobe Connect liegt darin, dass die Anwendung browserbasiert funktioniert. Die Nutzer müssen somit keine zusätzlichen Programme installieren (vgl. Jankowski et al. 2012: 13). Neben den technischen Möglichkeiten stellt die kostenlose Nutzung für Hochschulen einen Anreiz/Vorteil dar. Sie wird über das Deutsche Forschungsnetz (DFN) ermöglicht, in dem die NORDAKADEMIE Mitglied ist. Bei technischen Fragen steht zudem ein kostenloser Service des Kompetenzzentrums für Videokonferenzdienste des DFN zur Verfügung, der an der TU Dresden ansässig ist (vgl. Technische Universität Dresden 2015).

2. WIE WIRD ADOBE CONNECT AN DER NORDAKADEMIE EINGESETZT?

2.1 Webinare

Ein Webinar ist ein webbasiertes Seminar, in dem die interaktive Präsentation der Inhalte über das Internet durchgeführt wird. Es handelt sich hier um eine lerninhaltsbezogene Veranstaltung seitens eines online zugeschalteten Dozierenden. Im Fachgebiet Spanisch werden Webinare in drei verschiedenen hybriden Szenarien durchgeführt:

Hybrides Szenario 1 (Beispiel online aufrufbar)

Der Vortrag und die PowerPoint-Präsentation des externen Dozierenden werden mittels Adobe Connect live übertragen, während sich sämtliche Studierende in einem Seminarraum vor Ort an der NORDAKADEMIE befinden. Mit einer Konferenzwebcam (Logitech BCC950 ConferenceCam) ist es möglich, die Studierenden live in den Meetingraum zu übertragen, sodass die Dozierenden das Geschehen unter den Studierenden verfolgen können. Ein weiterer Dozierender im Saal übernimmt die Moderation. Neben Ton- und Bildübertragung wird der Freigabepod für die PowerPoint-Präsentation eingesetzt. Während der aktiven Interaktionsphase mit dem Referenten und unter den Teilnehmern wird das Layout Diskussion ausgewählt, in dem der Kamera- und Tonpod den größten Platzanteil ausmacht. Die Veranstaltung kann nun verschiedene Abläufe haben:

- Onlineseminar: Eine aktive und kreative Interaktion zwischen Referenten und Studierenden sowie unter den Studierenden selbst wird durch unterschiedliche Gruppendynamiken gefördert. An dieser Stelle ist die Aufgabe des Moderators vor Ort besonders gefordert.
 - Onlinevorlesung oder sog. Expertengespräch: Der externe Dozierende referiert und die Studierenden hören dem Vortrag zu. Die Kommunikation ist tendenziell einseitig. In der Fragerunde können die Studierenden ihre Fragen direkt an die Vortragenden stellen oder über den Moderator, der sich im Saal befindet.
- Hybrides Szenario 2: (Beispiel online aufrufbar)
Die beteiligten Personen befinden sich in drei verschiedenen Umgebungen: ein externer Dozierender ist aus dem Ausland zugeschaltet, eine Gruppe von Studierenden ist vor Ort an der NORDAKADEMIE anwesend und eine dritte Gruppe von Studierenden ist online aus verschiedenen Orten zugeschaltet. Auch anwesend an der NORDAKADEMIE ist ein Dozierender, der die Moderation übernimmt. Den Onlinestudierenden, die direkt an den Referenten ihre Fragen stellen möchten, werden die Rechte für die Tonübertragung zugewiesen. Während des Vortrags wird empfohlen, den online zugeschalteten Studierenden keine Tonübertragungsrechte zuzuweisen, um mögliche Störungen zu vermeiden. Fragen, Kommentare oder Bemerkungen können während des Vortrags im Chat angebracht werden.
 - Hybrides Szenario 3 (Beispiel online aufrufbar)
Sämtliche beteiligte Personen (Referent, Moderator der Sitzung und Studierende) sind online zugeschaltet. Auch in diesem Szenario wird empfohlen, den Studierenden keine Tonübertragungsrechte zuzuweisen. Sie können weiterhin im Chat mit dem Moderator

und mit dem Vortragenden kommunizieren. Erst während der Fragerunde können einzelne Teilnehmer Rechte zur Tonübertragung zugewiesen bekommen.

In den Masterstudiengängen wird das Modul „Wissenschaftliches Arbeiten“ für beinahe alle Studiengänge angeboten. Die Präsenzphase von 2,5 Tagen findet in zwei Intervallen statt, wobei der halbe Tag etwa eineinhalb Jahre nach den ersten beiden Tagen stattfindet. Um Studierenden die Teilnahme ohne langen Anreiseweg zu ermöglichen, fand dieses Jahr ein Teleteaching-Angebot statt. Studierende konnten über Adobe Connect per Livestream an der Veranstaltung teilnehmen. Bild- und Ton wurden wie im hybriden Szenario 1 übertragen. Beim Einsatz von PowerPoint-Präsentationen wurden diese in Vollbild auf dem SMART Board angezeigt. Zeitgleich wurde der Chat über einen Moderator vor Ort betreut und Fragen der zugeschalteten Studierenden direkt an die Dozierenden weitergegeben. Verwendeten die Dozierenden keine Folien, wurde der Adobe Connect Raum auf dem SMART Board projiziert, sodass die Fragen direkt von den Dozierenden gelesen und beantwortet wurden. Zu beachten ist, dass Fragen aus dem Auditorium vor Ort durch die Dozierenden wiederholt werden, sodass die Onlineteilnehmer die Frage verstehen. Alternativ kann ein zweites Mikrofon eingesetzt werden. Fragen aus dem Chat sollten laut vorgelesen werden, damit die Vor-Ort-Teilnehmer die Fragen mitbekommen und die entsprechende Antwort nachvollziehen können.

Der didaktische Mehrwert der Webinare im Hochschulkontext besteht unter anderem darin, dass sie den Studierenden erlauben, einen schnellen und unkomplizierten Kontakt zu externen Dozierenden und Experten herzustellen, die sich außerhalb ihrer geographischen Reichweite befinden.

2.2 Virtuelle Gruppenarbeitsräume

Die Masterstudiengänge finden zu großen Teilen in einer Selbststudienphase statt, die über die E-Learning-Infrastruktur der NORDAKADEMIE realisiert wird. Gerade während dieser Phasen oder während der Projektarbeiten ist ein direkter Austausch in regelmäßigen Treffen sehr hilfreich, wozu bei berufs begleitenden Studiengängen oft die Zeit fehlt. Webkonferenzräume schaffen hier Abhilfe. Über einen Buchungsplan können die Studierenden selbstständig die Räume buchen. Das Integrationsniveau ist hierbei hoch, da sich die Studierenden gegenseitig über aktuelle Gruppenarbeiten bzw. Projekte austauschen. Die Studierenden erhalten deshalb Moderatorenrechte.

Da nicht jeder Studierende Erfahrung im Umgang mit Adobe Connect besitzt, wird die Bedienung hybrid erklärt. Eine schriftliche Anleitung führt in die Struktur und Bedienung ein. Um Funktionsbeschreibungen zu verdeutlichen, wird an verschiedenen Stellen auf Podcasts verwiesen. Dies sind etwa fünfminütige Videoanleitungen, in denen die Bedienung exemplarisch dargestellt wird. Dadurch soll sichergestellt werden, dass die Bedienung Schritt für Schritt nachvollzogen werden kann.

2.3 Videosprechstunden

In der Selbststudienphase der Masterstudiengänge bieten Dozierende im eigenen Ermessen Videosprechstunden über Adobe Connect an. In dieser Sprechstunde wird ein direkter Kontakt zwischen Dozierenden und Studierenden ermöglicht. Ein wesentliches Kernthema ist hierbei die Selbststudienanleitung, welche den organisatorischen Verlauf (zeitlich und inhaltlich) regelt. Für die technische Umsetzung benötigen die Dozierenden ein Headset und eine Webcam.

Studierende stellten ihre Fragen einheitlich über den Chat, in welchem der Dozent mündlich antwortet. Im zweiten Halbjahr des Jahres 2015 wurde der Versuch, Studierenden die Mikrofonnutzung zu ermöglichen, erneut unternommen. An der Sprechstunde nahmen 3 Studierende teil, wobei ein Studierender die Mikrofonfunktion sehr ausgiebig nutzte, sodass ein reger Dialog zwischen dem Studierenden und dem Dozenten entstand. Die anderen beiden Studierenden nutzten das Mikrofon nicht, da dieses nicht funktionierte bzw. kein Headset zur Verfügung stand.

3. VOR- UND NACHTEILE

Der Einsatz von virtuellen Klassenzimmern mit Adobe Connect beinhaltet Vor- und Nachteile (siehe Tabelle 3; vgl. Ehlers 2011: 86 f.; Grotlüschen 2004, für Beschreibungen der Lurking-Problematik S. 135 f.).

Bewertungskriterien	Vorteil	Nachteil
Zeitlicher Aspekt	<ul style="list-style-type: none"> • Direkte Teilnahme ohne Anfahrtszeiten 	<ul style="list-style-type: none"> • Eventuelle Anpassung im didaktischen Konzept
Betreuungstechnischer Aspekt	<ul style="list-style-type: none"> • Persönlicher Kontakt während Onlinestudiumsphasen • Möglichkeit kurzer regelmäßiger Treffen 	<ul style="list-style-type: none"> • Ungewohnte Art der Wissensvermittlung • Lurking-Problematik • Gefahr des Motivationsverlust (Dozent kann weggedrückt werden)
Technischer Aspekt	<ul style="list-style-type: none"> • Technische Ausstattung ist in der Regel vorhanden 	<ul style="list-style-type: none"> • Schlechte Internetverbindungen oder technische Störungen können die Veranstaltungen unterbrechen
Kostentechnischer Aspekt	<ul style="list-style-type: none"> • Kostenlose Softwarenutzung für Hochschulangehörige • Technische Grundausstattung ist in der Regel in den Endgeräten enthalten 	<ul style="list-style-type: none"> • Technische Ausrüstung für komplexe Anwendungsszenarien, wie Teleteaching, müssen angeschafft werden • Personalkosten für Schulungen und Betreuung während der Veranstaltungen

Tabelle 3: Vor- und Nachteile von virtuellen Klassenzimmern mit Adobe Connect

4. FAZIT UND AUSBLICK

In der vorliegenden Arbeit wurde ein Überblick über den aktuellen Einsatz von virtuellen Klassenzimmern an der NORDAKADEMIE gegeben. Dieser hat sich als eine unerlässliche Ergänzung und eine qualitative Bereicherung von an der Hochschule durchgeführten E-Learning-Szenarien bewährt. Der Einsatz von virtuellen Klassenzimmern fördert kollaborative sowie ortsflexible Lern- und Lehrprozesse. Er dient als Schnittstelle zwischen traditionellen Präsenzveranstaltungen und asynchronen Arbeitsphasen in onlinebasierten Lernplattformen wie Moodle. Im Vergleich zu der alleinigen Arbeit in Lernplattformen, kann das virtuelle Klassenzimmer viele Vorteile der face-to-face-Kommunikation durch die Sendung von Ton und Bild übernehmen.

Der Einsatz von virtuellen Klassenzimmern mit Adobe Connect beinhaltet nicht nur Vorteile, sondern auch mögliche Nachteile wie technische Probleme oder einen hohen Zeitaufwand bei der Anpassung des didaktischen Konzeptes. Letztlich geht es darum, die Vor- und Nachteile für die eigene Veranstaltung und die zugrunde liegende Didaktik abzuwägen. Versuche in Szenarien mit kontrollierbarem Integrations- und Interaktionsniveau können bei dieser Abwägung unterstützend wirken.

5. QUELLENANGABEN

- Ehlers, U. – D. (2011): Qualität im E-Learning aus Lernericht. 2., überarbeitete und aktualisierte Auflage, Heidelberg, VS Verlag.
- Grotlüschen, A. (2004): Verschwiegene Interessen und ungefragte Fragen beim E-Learning als Ausdruck fehlender Einbindung heterogener Lerninteressen, in: Bos, W.; Lankes, E.-M.; Plaßmeier, N.; Schwippert, K. (Hrsg.): Heterogenität – Eine Herausforderung an die empirische Bildungsforschung, Münster, S. 124 – 140.
- Häfele, H.; Maier-Häfele, K. (2012): 101 e-Learning Seminarverfahren – Methoden und Strategien für die Online- und Blended-Learning-Seminarpraxis, Bonn, managerSeminare Verlags GmbH.
- Jankowski, R.; Osthoff, M.; Zöller-Greer, P. (2012): Virtuelles Klassenzimmer und Teleteaching für die Praxis, Wächtersbach, Composita Verlag.
- Ebster, C.; Stalzer, L. (2003): Wissenschaftliches Arbeiten für Wirtschafts- und Sozialwissenschaftler, 2. Auflage, Wien, WUV.
- Seifert, J. W.; Kerschbaumer, B. (2012): 30 Minuten Online Moderation, 2., überarbeitete Auflage, Offenbach, GABAL.
- Technische Universität Dresden (2015): Kompetenzzentrum für Videokonferenzdienste, zugegriffen über: <http://vcc.zih.tu-dresden.de/index.php> am 12.12.2015.
- Trujillo, C.; Essenwanger, F. (2013): Erfahrungen mit der Software Adobe Connect im Fachgebiet Spanisch an der NORDAKADEMIE, in: Plate, Georg: Forschung für die Wirtschaft 2013, Göttingen, Cuvillier, S. 445 – 463.
- Links zu den verschiedenen Szenarien
- Hybrides Szenario 1: <https://webconf.vc.dfn.de/p9ossfzt6d>
- Hybrides Szenario 2: <https://webconf.vc.dfn.de/p210rm1am87/>
- Hybrides Szenario 3: <https://webconf.vc.dfn.de/p4myfhyor1a/>

EIN MASSIVE OPEN ONLINE COURSE ALS E-LEARNING-KOMPONENTE IM INVERTED CLASSROOM MODELL



Frank Zimmermann
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Uwe Neuhaus
NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn

Abstract: Glaubt man den Protagonisten des E-Learnings, so stößt die traditionelle Lehre an Hochschulen hinsichtlich Transparenz, Lernautonomie und bereitgestellter Lernmaterialien an Grenzen, die der Qualität und Aktualität der Lehre entgegenstehen (Handke, Schäfer 2012). Der vorliegende Artikel beschreibt einen Modellversuch, der im vierten Quartal 2014 im Rahmen eines Wahlpflichtfachs an der NORDAKADEMIE durchgeführt wurde, und dessen Ziel es war, die angesprochenen Defizite mittels eines Inverted Classroom Modells auf Basis eines MOOCs zu beseitigen.

Keywords: Hochschullehre, E-Learning, Inverted Classroom Model, MOOC

1. PROBLEMSTELLUNG

Ein zentraler Aspekt einer Fachhochschule ist ihr Angebot an praxisrelevanter aktueller Lehre. Studenten sollen – besonders an dualen Hochschulen – an Techniken herangeführt werden, die von unmittelbarer Relevanz für die berufliche Praxis sind. Insbesondere in Fachrichtungen mit kurzen Innovationszyklen wie etwa der Wirtschaftsinformatik stellt dies die Lehrenden vor besondere Herausforderungen. Themen wie Cloud-Computing, mobile Anwendungen oder Big Data erlangen innerhalb weniger Jahre große Bedeutung in Unternehmen, sind aber gleichzeitig technologisch sehr anspruchsvoll und müssen ständig auf dem neusten Stand gehalten werden.

Obwohl die Einbindung aktueller technischer Fragestellungen in das Studium für die Studierenden einen großen Mehrwert darstellt, ist ihre Verankerung im Pflichtbereich eines Studiengangs aus zwei Gründen nicht empfehlenswert: Erstens sind die Veränderungszyklen von Studienplänen aufgrund formaler Anforderungen zu lang, um rasch auf technische Änderungen reagieren zu können. Und zweitens verbreiten sich neue Technologien typischerweise nicht in allen Branchen gleich schnell, so dass nicht alle Studierenden unmittelbar Interesse an der neuen, herausfordernden Materie entwickeln. Flexibler und

Prof. Dr. Frank Zimmermann ist seit 1995 Professor an der Fachhochschule NORDAKADEMIE. Nach seiner Promotion im Fach Mathematik arbeitete er 8 Jahre in der IT Branche als Berater. Er lehrt in den Bereichen Mathematik, Programmierung und Softwareengineering. Sein Forschungsinteresse gilt modernen Methoden der Softwareentwicklung, speziell der modellgetriebenen Entwicklung.
E-Mail: frank.zimmermann@nordakademie.de

Dipl.-Inform. Uwe Neuhaus arbeitet seit April 2011 als wissenschaftlicher Mitarbeiter im Fachbereich Informatik an der NORDAKADEMIE, war zuvor aber bereits mehrere Jahre als Lehrbeauftragter für die Hochschule tätig. Nach seinem Informatikstudium an der Technischen Universität Braunschweig nahm er an einem einjährigen, interdisziplinären Ergänzungsstudium in St. Louis, USA teil. Anschließend arbeitete er für die FernUniversität Hagen im Lehrgebiet Datenbanken und Informationssysteme sowie als Technical Trainer und Manager Training für einen führenden Entwickler von Content Management Systemen. Seine Schwerpunkte an der NORDAKADEMIE sind die Bereiche Algorithmen, Data Science und Machine Learning.
E-Mail: uwe.neuhaus@nordakademie.de

besser geeignet ist es daher, entsprechende Lehrangebote interessierten Studierenden im Rahmen von Wahlpflichtveranstaltungen anzubieten. Nachteilig dabei ist allerdings, dass trotz des großen und kontinuierlichen Vorbereitungsaufwands solcher spezialisierter Veranstaltungen diese nur in geringem Umfang zum Lehrdeputat der Dozenten beitragen.

Insgesamt ergeben sich folgende Rahmenbedingungen für die Lehre innovativer technologischer Themen:

- Große Nachfrage nach Lehrangeboten zu aktuellen Technologien von Studierenden und Unternehmen.
- Hohe technische Komplexität setzt eine intensive Beschäftigung mit der Materie voraus und verursacht einen außergewöhnlich hohen Vorbereitungsaufwand durch die Lehrenden.
- Hoher Innovationsgrad erfordert kontinuierliche Aktualisierung der Materialien.
- Die Lehrveranstaltungen decken trotz ihres großen Aufwands nur wenig Lehrdeputat ab.

Eine Form, die sich in der Vergangenheit als geeignet erwies, den oben genannten Herausforderungen zu begegnen, sind projektorientierte Lehrveranstaltungen. Hierbei wird eine praxisrelevante Problemstellung von einer Gruppe von Studierenden gelöst. Die Lehrenden übernehmen dabei nicht mehr die Rolle des Fachexperten, sondern agieren als Auftraggeber und Coach. Sie vermitteln notwendiges Basiswissen und delegieren technische Details gegebenenfalls an externe Spezialisten. Der größte Anteil am Wissenserwerb der Studierenden muss problemorientiert durch sie selbst erarbeitet werden.

Diese Lehrform setzt jedoch eine gewisse Reife und einen ausreichenden Erfahrungshintergrund bei den Studierenden voraus. Die Autoren haben in den Jahren 2012 und 2013 das Wahlpflichtfach „Mobile Anwendungen“ als projektorientierte Lehrveranstaltung im fünften Semester des Bachelorstudiengangs Wirtschaftsinformatik durchgeführt und dabei folgende Erfahrungen gemacht:

- Nach der Reduzierung der Schulzeit für Abiturienten sowie der Umstellung vom Diplom auf den Bachelorabschluss und der damit einhergehenden Verkürzung und stärkeren Vorstrukturierung des Studiums war die Mehrzahl der Studierenden im fünften Semester noch nicht in der Lage, selbständig neue Inhalte zu erarbeiten. Bei früheren Studienjahren war diese Fähigkeit in ähnlich gelagerten Wahlpflichtfächern noch spürbar stärker vorhanden.
- Der Ausbildungsstand der Studierenden im fünften Fachsemester ist noch nicht ausreichend, um anspruchsvolle Projekte durchzuführen. Zum Beispiel sind die Studierenden noch nicht mit dem Umgang mit Frameworks vertraut, fehlen ihnen Kenntnisse in der Gestaltung von komplexen Softwaresystemen und Softwarearchitekturen, haben sie Probleme, Ideen abstrakt in Modellen und Szenarien zu formulieren.
- Projektarbeiten mit mehr als zehn Teilnehmern erwiesen sich aufgrund des steigenden Kommunikations- und Abstimmungsaufwands als schwer durchführbar. Die Teilnehmerzahl des Wahlpflichtfaches war jedoch bis zu dreimal so hoch.
- Die Benotung der Ergebnisse der Projektarbeiten erwies sich als schwierig, weil Referenzarbeiten, die als Vergleichsmaßstab dienen können, fehlten.

Die Autoren haben sich aufgrund der gesammelten Erfahrungen bei der Durchführung des Wahlpflichtfachs „Mobile Anwendungen“ im Jahr 2014 daher für eine neue Lehrform entschieden. In diesem Jahr wurde ein Massive Open Online Course im Inverted Classroom Model verwendet. Die folgenden Kapitel beschreiben, welche Vorteile von der neuen Lehrform erwartet wurden, auf welchen Prinzipien sie basierte und welche Erfahrungen damit gesammelt wurden.

2. THESEN

In diesem Kapitel sollen folgende Thesen für die Verwendung eines Inverted Classroom Models (ICM) auf Basis eines Massive Open Online Courses (MOOC) begründet werden.

- 1) Verbesserung der Lehrinhalte:
 - Die Nutzung eines geeigneten MOOCs ermöglicht die Realisierung einer inhaltlich aktuellen und technisch anspruchsvollen Lehrveranstaltung.
 - Präsenzanteile der Lehrveranstaltung gehen über ein Tutorialniveau hinaus und können sich auf fortgeschrittene Fragestellungen (z. B. auf Architekturfragen oder spezielle Anwendungsszenarien) konzentrieren.
 - Die verschiedenartigen Lehrmedien (Moderation durch Dozenten sowie MOOC-Videos und -Code-Beispiele) ermöglichen eine unterschiedliche Sichtweise auf den Lernstoff und machen diesen Unterschied für die Studierenden erfahrbar.
- 2) Verbesserung der Transparenz:
 - Die Lernziele können in der Präsenzveranstaltung differenzierter herausgearbeitet werden.
 - Die Benotung der Leistung als Erreichungsgrad der Lernziele wird dadurch vereinfacht und für die Bewerteten transparent gemacht.
- 3) Verbesserung der Lernautonomie:
 - Die Lerngeschwindigkeit innerhalb der MOOC-Lernmaterialien kann von den Lernenden individuell festgelegt werden. Außerdem lassen sich beispielsweise Videoabschnitte nach Bedarf wiederholen.
 - Der Ort zur Bearbeitung der MOOC-Materialien kann von den Studierenden beliebig gewählt werden.
 - Der Lernumfang kann durch umfangreiches Zusatzmaterial individuell erweitert werden.
- 4) Verbesserung der Verfügbarkeit von Lehrmaterialien:
 - Die Erklärvideos des MOOCs stehen auch während der Hausarbeit zur Verfügung. So können relevante Lernbestandteile bei Bedarf nachträglich noch einmal genauer betrachtet und analysiert werden.

3. INVERTED CLASSROOM MODEL

Das Inverted Classroom Model (ICM, auch *inverted teaching* oder *flipped classroom* genannt) bezeichnet eine Unterrichtsmethode des integrierten Lernens und wurde erstmals im Jahr 2000 von Lage beschrieben (Lage et al. 2000) und später unter anderem durch die Publikationen von Bergmann und Sams einer breiteren Öffentlichkeit bekannt gemacht (Bergmann & Sams 2012). Das ICM wird an Schulen genutzt, aber auch von verschiedenen Hochschulen in unterschiedlichen Fachbereichen (NMC 2013). In Deutschland findet seit 2012 an der Universität Marburg jährlich eine ICM-Fachtagung statt.⁶

Das zentrale Konzept des Inverted Classroom Models besteht im Austausch der Lern- und Übungsorte (Kück 2014; ICM Wiki 2015). In klassischen Vorlesungen werden die Inhalte von den Dozenten durch Lehrvortrag oder seminaristischen Unterricht vermittelt. Der initiale Lernvorgang findet also innerhalb der Vorlesung statt. Die aktivere Rolle übernehmen dabei die Lehrenden, die Rolle der Lernenden beschränkt sich überwiegend auf die Rezeption des Lernstoffes. Zum Einüben und Festigen des Gelernten werden daher von den Dozenten Aufgaben bereitgestellt, die von den Studierenden dann im Nachgang und außerhalb der Vorlesung gelöst werden sollen. Im ICM hingegen erfolgt die initiale Beschäftigung mit den Lerninhalten außerhalb und vor der Vorlesung. Die Lehrenden definieren dafür geeignete Lernmaterialien (Lernvideos, Lehrbuchabschnitte, E-Learning-Materialien usw.) und stellen diese bereit. Die Materialien werden von den Studierenden eigenständig und selbstbestimmt

⁶<https://invertedclassroom.wordpress.com/>

vor der Präsenzveranstaltung durchgearbeitet. Dabei sind sie weder an einen konkreten Ort, eine bestimmte Uhrzeit oder eine vorgegebene Bearbeitungsdauer gebunden. Die zeit- und örtlich fixierte Präsenzveranstaltung wird anschließend insbesondere für folgende Tätigkeiten genutzt:

- Gemeinsame und interaktive Zusammenfassung der erarbeiteten Lerninhalte
- Klärung von Fragen, die beim selbstbestimmten Lernen offen geblieben sind
- Hervorhebung und Einordnung wichtiger Aspekte
- Herausarbeiten von Vor- und Nachteilen der gelernten Inhalte
- Einüben des gelernten Stoffs anhand neuer Aufgaben und Fragestellungen
- Bewertung der Anwendbarkeit des Gelernten auf neue Szenarien

Die Lernenden übernehmen innerhalb der Präsenzphase eine wesentlich aktivere Rolle als in der klassischen Vorlesung; die Dozenten coachen und moderieren. Die Bedeutung der Präsenzphase wird so im ICM gestärkt, da die dort durchgeführten Aktivitäten Lernziele unterstützen, die sich auf höheren Ebenen der Lernzieltaxonomie (z. B. nach Bloom et. al. 1956) befinden als dies bei traditionellen Lehrveranstaltungen der Fall ist.

4. MASSIVE OPEN ONLINE COURSES

Sogenannte Massive Open Online Courses sind eine spezifische Art von Online-Kursen (siehe Hill 2012; Bremer 2013). Die Adjektive „massive“ und „open“ bringen zum Ausdruck, dass diese Kurse für eine sehr große Teilnehmerzahl ausgelegt sind (tausende oder sogar zehntausende Teilnehmer) und dass die Teilnahme grundsätzlich allen Internetnutzern offensteht, unabhängig von geografischer Herkunft, Alter oder bisherigem Bildungsstand. Angeboten werden MOOCs meist von Hochschulen, in letzter Zeit aber auch von größeren Unternehmen wie Google, Microsoft oder Facebook. Im ersten Fall orientieren sich die MOOC-Inhalte häufig am Inhalt regulärer Hochschulvorlesungen, im zweiten an speziell für den Arbeitsmarkt relevanten Qualifikationen. Zugang zu den MOOC-Materialien erhalten die Teilnehmer typischerweise über eine MOOC-Plattform⁷, auf der zahlreiche MOOCs verschiedener Anbieter gebündelt und mit einer einheitlichen Benutzerschnittstelle versehen werden. In der Regel ist die Teilnahme an einem MOOC kostenlos möglich, für verschiedene Zusatzleistungen (Zertifizierung, Vergabe von Kreditpunkten, individuelle Online-Betreuung, Präsenzprüfungen) werden jedoch häufig Gebühren erhoben.

Lehrvideos, Übungsaufgaben, Tests sowie diverse Zusatzmaterialien (Kurssyllabus, FAQ, Diskussionsforum, Lernfortschrittsanzeige usw.) sind typische Bestandteile eines MOOCs. Die zeitliche Struktur zur Bearbeitung der Lerneinheiten und Übungen ist häufig vorgegeben, da dies für den organisatorischen Ablauf Vorteile besitzt (beispielsweise für die Veröffentlichung von Testfragen oder bei Peer-Review-Aufgaben). Es gibt aber auch MOOCs, die den Lernenden eine völlig freie Zeiteinteilung ermöglichen. MOOCs, die sich in ihrer Struktur eher am traditionellen Vorlesungskonzept orientieren, werden xMOOCs genannt. Bildet ein MOOC hingegen eher ein Seminar oder einen Workshop mit den Mitteln des Internets ab, bezeichnet man ihn als cMOOC (Robes 2012). Kursteilnehmer, Lehrende und (häufig ehrenamtliche) Tutoren kommunizieren und vernetzen sich in der Regel mittels Foren, können aber zusätzlich auch Werkzeuge wie Wikis, Facebook-Gruppen oder Twitter nutzen. Bedingt durch die hohen Teilnehmerzahlen werden bei Tests und Prüfungen entweder Aufgabenformen verwendet, die automatisch geprüft werden können (z. B. Multiple-Choice, Lückentexte, Programm-Code-Fragmente) oder man greift auf ein Peer-Review-Verfahren zurück, bei dem jeder Teilnehmer mehrere anonymisierte Lösungen von anderen Teilnehmern auf Grundlage eines detaillierten Bewertungskataloges beurteilt.

⁷ Bekannte MOOC-Plattformen sind etwa Coursera (www.coursera.org), EdX (www.edx.org), Udacity (www.udacity.com) oder, im deutschsprachigen Raum, Iversity (www.iversity.org).

MOOCs sind eine vergleichsweise junge Entwicklung, bei der noch vielfältige didaktische und technische Umsetzungsformen erprobt werden. Zwei wichtige Aspekte beim Einsatz von MOOCs im Rahmen der Hochschullehre sind die zeitliche Verfügbarkeit der Materialien und ihre lizenzrechtlichen Nutzungsvoraussetzungen.

5. MOBILE ANWENDUNGEN

Das Modul „Mobile Anwendungen“ wird als Wahlpflichtmodul im 5. Semester angeboten. Es richtet sich überwiegend an Wirtschaftsinformatikstudenten, interessierte Studenten des Studiengangs Wirtschaftsingenieurwesen haben allerdings auch schon erfolgreich an dem Kurs teilgenommen. Die Studierenden haben zu diesem Zeitpunkt ihre Programmiergrundausbildung abgeschlossen und erste Erfahrungen in der Programmiersprache Java gesammelt.

5.1 Lernziele

Vermittelt werden sollen die Grundprinzipien der Entwicklung von mobilen Anwendungen in betriebswirtschaftlichen Anwendungsszenarien, wie sie bei den Kooperationsbetrieben der NORDAKADEMIE typischerweise auftreten. Dabei soll eine realistische Sicht auf die Komplexität der Entwicklung mobiler Anwendungen vermittelt werden, bei der deutlich wird, dass für den kommerziellen Einsatz geeignete Apps gewissen Standards genügen und nach speziellen Programmiermustern entworfen werden müssen.

Als Zielplattform wurde Android gewählt, da es sich dabei um die am weitesten verbreitete mobile Plattform handelt (Statista 2015) und vornehmlich in Java programmiert wird, was den Vorkenntnissen der Studierenden entgegenkommt. Im Kern der Veranstaltung stehen technische Fragestellungen, die sich aus den Besonderheiten der mobilen Infrastruktur ergeben und deren Komplexität ausmachen:

- Integration in eine offene Infrastruktur mobiler Apps, in der verschiedene Anwendungen miteinander interagieren können.
- Umgang mit stark beschränkten Ressourcen wie zum Beispiel der Batterieleistung, dem verfügbaren Speicher und der im Verhältnis zu Desktop oder Serversystemen eingeschränkten CPU-Leistung.
- Umgang mit zeitlich eingeschränkt verfügbaren Ressourcen wie zum Beispiel dem WLAN oder der Mobilfunkverbindung.
- Umgang mit heterogenen mobilen Geräten (unterschiedliche Bildschirmgrößen, Auflösungen, Screen-Layouts usw.)
- Anforderungen der Anwender an Interaktionsformen und deren Laufzeitverhalten (Responsiveness).

5.2 Inhalte

Die Veranstaltung soll die Studenten in die Lage versetzen, eine einfache Geschäftsanwendung selbst zu entwickeln sowie korrekt und angemessen umzusetzen. Die dafür erforderlichen Grundkonzepte der Android-Architektur werden dabei anhand einer Beispielanwendung vermittelt. Dies geschah im 2013 in Tutorialform. Als Beispielanwendung diente dabei die im Buch von Marko Gargenta (Gargenta 2011) beschriebene Anwendung „yamba“ (yet another micro blogging application). Die wichtigsten Lehrinhalte waren dabei:

1. Aufbau und Zusammenspiel der Android-Grundkomponenten (z. B. Activities, Services, ContentProvider und Broadcast Receiver)
2. Die Bestandteile der grafischen Benutzeroberfläche (Widgets)
3. Darstellung umfangreicher Informationsmengen (mittels Listen und Adapter)
4. Einstellungsdialoge (Preference Screens)
5. Das Android-Rechtssystem
6. Zugriff auf Daten in Netzwerken (mittels JSON)

7. Die Android-Datenbank (SQLite)
8. Benachrichtigungen (Notifications)

Im Kern sollten dieselben Inhalte auch in der neu gestalteten Wahlpflichtveranstaltung 2014 vermittelt werden.

5.3 Aufbau der Lehrveranstaltung

Zur Unterstützung des Wahlpflichtfachs „Mobile Anwendungen“ kam der MOOC „Developing Android Apps“ auf der MOOC-Plattform Udacity zum Einsatz (Google 2013), der auf der Google Developer-Seite als Einstiegskurs in die Android-Entwicklung empfohlen wurde (Google 2013b). Für diesen Kurs sprach auch, dass er vom Android-Hersteller Google selbst stammt, auf dem aktuellen Stand Entwicklung war, die definierten Lerninhalte abdeckte und zeitlich flexibel absolviert werden konnte. Auf anderen MOOC-Plattformen werden ebenfalls Android-Kurse angeboten, diese können häufig jedoch nur während bestimmter Zeitperioden bearbeitet werden, was die Integration in das Studiensemester stark erschwert. Der gewählte Kurs hat einen geschätzten Umfang von 51 – 64 Stunden (exklusive Abschlussprojekt). Der Aufwand für die vorgesehenen sechs Lektionen wurde von den Autoren des MOOCs wie folgt angegeben:

Lektion	Inhalt	Dauer
1	Erzeugen eines Android-Projekts mit einfacher Benutzerschnittstelle	5-8 Stunden
2	Datenaustausch mit einem Server mittels JSON	8-10 Stunden
3	Erweiterung der App um neue Ansichten, Interaktionen und eine Navigation (Activities, Intents)	8-10 Stunden
4	Speicherung der Serverdaten in einer lokalen Datenbank (ContentProvider, Loader)	15-20 Stunden
5	Entwicklung einer für mobile Endgeräte geeigneten Bedienoberfläche (rich and responsive layout)	8-10 Stunden
6	Nutzung von Hintergrundprozessen (Services/Notifications)	5-8 Stunden

Tabelle 1: Inhalt und Dauer der MOOC-Lektionen

Das Wahlpflichtmodul ist mit einem Workload von 180 Stunden geplant, 60 davon als Kontaktstunden. Die Durcharbeit der MOOC-Inhalte sollte gemäß dem ICM weitgehend außerhalb der Vorlesungszeit stattfinden, sodass in der Vorlesung Fragen geklärt, Inhalte diskutiert sowie Konzepte zusammengefasst und gefestigt werden konnten. Da mehr als die Hälfte der Nicht-Kontaktstunden für die Anfertigung der abschließenden Hausarbeit eingeplant wurden, war klar, dass in den verbleibenden Nicht-Kontaktstunden nicht der gesamte Kurs durchgearbeitet werden konnte. Zum einen wurden deshalb diejenigen Inhalte des MOOCs für optional erklärt, die sich gut aus dem Gesamtkonzept herausnehmen ließen. Dadurch reduzierte sich der Aufwand um ca. 10 Stunden. Diese Anteile wurden entsprechend nicht in der Prüfungsleistung abgefordert. Zum anderen konnte ein Teil der Kontaktstunden ebenfalls zur Bearbeitung der MOOC-Inhalte genutzt werden. Konkret wurden die zur Verfügung stehen sechs Vorlesungsstunden pro Woche in zwei Terminen à drei Stunden abgehalten. An einem der Termine wurden die Inhalte der vergangenen Woche noch einmal aufgearbeitet und neue Konzepte und Aufgaben motiviert und vorbereitet. Die Teilnahme an diesem Termin war verpflichtend. Der andere Termin stand zur Verfügung, um die zu bearbeitenden MOOC-Aufgaben mit Unterstützung der Dozenten zu erledigen. Um die Idee des selbstgesteuerten Lernens zu fördern, konnte dieser Teil auch zu Hause erledigt werden. Beide Varianten wurden angenommen. Ein großer Teil der Betreuung während der Vorlesungszeiten beschäftigte sich

allerdings weniger mit den Inhalten des MOOCs, sondern mit dem Aufbau und der Erhaltung der Entwicklungsumgebung. Die sehr kurzen Entwicklungszyklen der Android-API und Entwicklungswerkzeuge, verbunden mit automatischen Updates, erwiesen sich als ein Stolperstein, der zu viel zusätzlichem Aufwand führte.

5.4 Best Practices

Grundlage der Taktung der Veranstaltung waren wöchentliche Aufgabenbeschreibungen mit To-dos für die Studierenden. Sie wurden am Anfang jeder Woche in einen NORDAKADEMIE Moodle-Kurs eingestellt und enthielten neben einer Zusammenfassung der wichtigsten Lehrinhalte auch ungefähre Angaben, in welchem zeitlichen Rahmen die Aufgaben idealerweise zu erledigen waren.

Die Präsenzzeiten wurden zu unterschiedlichen Aktivitäten genutzt:

- 1) Wiederholung der gelernten Inhalte:
 - Gemeinsame, interaktive Entwicklung von Mindmaps zur Visualisierung der gelernten Inhalte.
Beispiel: Welche Konzepte wurden in der letzten Lerneinheit vermittelt?
 - Rekapitulation der wichtigsten Lernaspekte und Herausarbeiten der Stärken/Schwächen sowie der Grenzen der verwendeten Techniken.
Beispiel: Darstellung der Anwendungsszenarien der SQLite-Datenbank in mobilen Apps.
 - Übertragung des Gelernten auf neue Situationen.
Beispiel: Welche Lebenszyklusmethoden einer Activity werden ausgelöst, wenn man das mobile Endgerät dreht?
 - Bewertung von Szenarien und Einschätzung der Anwendbarkeit des Gelernten.
Beispiel: Welche Informationen sollten in einem Preferences-Dialog gespeichert werden?
- 2) Vorbereitung neuer Inhalte:
 - Motivation der Aufgaben der nächsten Lerneinheit.
Die wesentlichen Lerninhalte wurden vorbereitet und ihre Bedeutung für eine Entwicklung von mobilen Anwendungen hervorgehoben.

Eine der größten Herausforderungen bei dieser neuen Lehrveranstaltungsform war es, den Wissensstand der Studenten im Laufe des Semesters synchron zu halten.

6. VERGLEICH ZUR KLASSISCHEN VORLESUNG

6.1 Sicht der Dozenten

Die erste Herausforderung beim Einsatz eines MOOCs bestand darin, zunächst überhaupt einen geeigneten Kurs zu finden, der sich hinsichtlich Qualität, Anspruchsniveau und Umfang für die eigene Vorlesung eignet. Außerdem musste der Kurs während der Semesterzeiten verfügbar sein, was durchaus nicht bei allen MOOCs der Fall war, da ihre Inhalte häufig nur während bestimmter Zeitperioden angeboten wurden. Schließlich waren noch die Lizenzbedingungen des entsprechenden MOOC-Anbieters zu prüfen, die festlegen, ob bzw. wie die MOOC-Inhalte im Rahmen einer Vorlesung eingesetzt werden dürfen.

Die Dozenten sind sich einig, dass die Lerninhalte des Wahlpflichtmoduls auf Basis des MOOCs im ICM im Vergleich zur Durchführung als klassische Vorlesung im Vorjahr deutlich umfangreicher und besser auf das Lernziel abgestimmt sind. Zusätzlich zu den Themen in 2013 wurden 2014 folgende Themen behandelt:

- Fragmente und dynamische Bestandteile einer Activity
- Loader und nutzerdefinierte List Adapter
- Responsive Screen Layout für Tablet-Anwendungen

Diese zusätzlichen Bestandteile machen ca. 20 % bis 30 % des Gesamtlehrinhalts aus. Die geeignete Aufbereitung dieser Inhalte zur Vermittlung innerhalb einer klassischen Vorlesung wäre mit einem großen zeitlichen Aufwand verbunden gewesen.

Die Lehrmaterialien des ausgewählten MOOCs waren, wie erhofft, sehr aktuell und didaktisch durchdacht. Im Vergleich zum buchbasierten Durchgang im Jahr 2013 fielen ferner weniger Ungenauigkeiten in der Programmierung und Programmierfehler auf. Damals mussten an den im Buch abgedruckten Codebeispielen noch etliche Korrekturen und Aktualisierungen vorgenommen werden. Dies war einerseits dem Umstand geschuldet, dass ein Buch notwendigerweise der sehr dynamischen Android-Entwicklung hinterherhinkt, andererseits aber auch der Tatsache, dass der Buchautor sich nicht immer an die von Google vorgegebenen Programmierstandards hielt.

Da die Vermittlung der Inhalte im Wesentlichen durch den MOOC erfolgte, konnten sich die Dozenten im ICM stärker auf die Unterstützung des Lernprozesses, die Beantwortung von Fragen und den Ausgleich von individuellen Wissensständen konzentrieren. Dies war gerade beim Wahlpflichtfach „Mobile Anwendungen“ besonders hilfreich, da hier aufgrund der umfangreichen und sich häufig ändernden technischen Voraussetzungen erfahrungsgemäß besonders viele individuelle Fragen entstehen. Die Rolle des Dozenten verlagerte sich so vom reinen Lehrenden zum Moderator und Lern-Coach. Durch die Integration des MOOCs stieg außerdem die didaktische Methodenvielfalt, da sich die Methoden des Online-Kurses mit denen in den Präsenzveranstaltungen ergänzten.

Die Lerninhalte wurden im MOOC (ähnlich wie im zuvor verwendeten Buch) anhand einer praxisnahen Beispielanwendung vermittelt, was didaktisch von Vorteil ist. Im MOOC war diese Anwendung allerdings relativ komplex. Dadurch waren die einzelnen Lektionen teilweise sehr umfangreich. Zwar standen Beispiellösungen für jeden Programmierschritt in einem Versionsverwaltungssystem zur Verfügung, dennoch hatte jeder Schritt ein Lernziel, sodass man die Lösung nicht einfach kopieren konnte, ohne ein Lerndefizit zu erzeugen. Einige Teile des Codes wurden im Laufe der Veranstaltung außerdem mehrfach neu geschrieben. Das ist zwar in der Praxis der Softwareentwicklung durchaus üblich, dennoch wäre eine zielstrebigere Vorgehensweise für die Studierenden mit weniger Aufwand verbunden gewesen. Die Komplexität der Beispielanwendung war durch den MOOC vorgegeben und ließ sich durch keine Aktivitäten der Dozenten nennenswert reduzieren.

Insbesondere die Lektion 4 stellte sich im Nachhinein als übergroße Aufgabe heraus. Die Studenten brauchten etwa drei Wochen, um diese Lektion durchzuarbeiten, einige beendeten die Lektion sogar nicht vollständig. Diese Lektion wurde wegen ihres Umfang und der relativ geringen Aufbereitung durch die MOOC-Anbieter auch im Forum des MOOCs kritisiert. Die Dozenten gaben, so weit möglich, Hilfestellungen, konnten den benötigten Aufwand aber kaum vermindern, da die Inhalte der Lektion für die Beispielanwendung zentral waren.

Als Prüfungsleistung wurde eine Aufgabenstellung ausgesucht, deren Struktur sich stark an der im MOOC erarbeiteten Beispielanwendung orientierte. Dadurch konnten die erlernten Techniken von den Studierenden vertieft und in einen neuen Anwendungskontext übertragen werden. Außerdem konnte sich die Bewertung der Hausarbeit so an der geeigneten Nutzung der präsentierten Techniken orientieren. Bei diesem Aspekt gab es bei den Hausarbeitslösungen allerdings eine unerwartet große Streuung, was zur Ausschöpfung des Notenspektrums führte.

6.2 Feedback der Studierenden

Am Ende der Veranstaltung wurden die Studierenden zu ihrer Einschätzung der neuen Lehrform befragt. Von 28 Teilnehmern füllten 15 eine speziell entworfene Moodle-Umfrage aus. Obwohl an einigen Aspekten Kritik geübt wurde, beurteilten die Studierenden die Möglichkeit, sich Lerninhalte selbstgesteuert anzueignen, überwiegend positiv. Die wichtigsten Ergebnisse der Umfrage sind:

- Die meisten Studenten hatten vorher keine oder wenige Vorerfahrungen mit MOOCs gesammelt, waren jedoch schnell in der Lage mit dem MOOC umzugehen.
- Neben den Lernvideos wurden vor allem die verlinkten Online-Artikel und die im Versionsverwaltungssystem hinterlegten Musterlösungen genutzt. Das Diskussionsforum des MOOC wurde nur in geringem Umfang genutzt (von etwa 1/4 der Studenten), vermutlich, weil Fragen auch im Rahmen der Vorlesung geklärt werden konnten.
- Die Möglichkeit, sich selbstgesteuert in eine komplexe Thematik einzuarbeiten, wurde größtenteils eher positiv bewertet. Nur ein Student empfand dies eher negativ.
- Dies deckt sich mit dem vorherrschenden Feedback, dass der verwendete Vorlesungstyp vor allen Dingen das eigenständige Lernen unterstützt. Auch hier war allerdings wieder ein Student der Meinung, dass praktisch alle Kompetenzen durch eine traditionelle Veranstaltung besser vermittelt werden können.
- Der Arbeitsaufwand wurde etwa 20 % höher eingeschätzt als bei einer klassischen Vorlesung. Dieser Mehraufwand entspricht ungefähr den zusätzlich vermittelten Inhalten gegenüber dem Wahlpflichtfach 2013. Hier deckt sich die Einschätzung der Dozenten mit der der Studierenden.
- Als Herausforderung wurde von den Studierenden das benötigte Maß an Selbstdisziplin gesehen, das für diese neuartige Lehrform Voraussetzung ist. Einige wünschten sich daher eine umfangreichere und kontinuierliche Lernfortschrittskontrolle durch die Dozenten.
- Als Vorteil wurde herausgestellt, dass der MOOC von Google (dem Entwickler von Android) selbst produziert wurde und die vermittelte Information daher aus erster Hand stammte.
- Die Stoffzusammenfassungen durch die Dozenten während der Präsenzzeiten wurden von etwa drei Viertel der Studierenden als hilfreich oder sehr hilfreich beurteilt. Einige wünschten sich zusätzlich kleine, traditionelle Vorlesungseinheiten oder ein Skript.
- Die gemeinsame Bearbeitung der MOOC-Inhalte im Rechnerraum wurde teilweise kritisch gesehen. Zwar bestand so die Möglichkeit, Fragen an die Dozenten zu stellen, allerdings erschwerte der durch die Gruppenarbeit erhöhte Lärmpegel das konzentrierte Studium der Lernvideos. Bei der Arbeit zu Hause wurde hingegen die Gelegenheit zur unmittelbaren Kommunikation mit den Dozenten vermisst.
- Das Niveau des Wahlpflichtfachs wurde größtenteils als überdurchschnittlich und weit überdurchschnittlich eingestuft. Auch dies entspricht dem Eindruck der Dozenten. Die Tatsache, dass die Kursinhalte in englischer Sprache angeboten wurden, spielte dabei nur für ein Drittel der Studierenden eine Rolle.
- Wenn beim Wahlpflichtfach beide Lehrveranstaltungsformen zur Auswahl ständen, würden über 60 % der Umfrageteilnehmer die MOOC/ICM-Variante einer Vorlesung in traditioneller Form vorziehen. Einige Teilnehmer würden eine in etwa gleichgewichtige Mischung von MOOC-Elementen und traditioneller Vorlesung präferieren. Einige wenige Studierende lehnen die neue Lehrveranstaltungsform kategorisch ab.

7. FAZIT

Insgesamt betrachten die Dozenten die Veranstaltung als Erfolg. Die ursprüngliche Hauptmotivation zur Nutzung des MOOCs im Wahlpflichtfach, die Steigerung der Aktualität und Qualität der angebotenen Lehrmaterialien (These 1), wurde erreicht. Die von den Studenten kritisierte zeitliche Belastung wird eher als Indiz für eine inhaltlich anspruchsvolle Veranstaltung gesehen. Die Schaffung einer größeren Transparenz durch die stärkere Herausarbeitung der Lernziele (These 2) gelang nur bedingt. Die Ergebnisse der Hausarbeit zeigten, dass bei den Studierenden in diesem Punkt weiterhin Erklärungsbedarf besteht. Eine größere Lernautonomie (These 3) sowie eine verbesserte Verfügbarkeit der Lehrmaterialien (These 4) konnten durch den Einsatz des MOOCs prinzipiell erzielt werden. Es zeigte sich aber auch, dass die größere Autonomie und die damit verbundene eigene Verantwortung für viele Studierende noch eine gewisse Herausforderung darstellt. Dies sollte sich jedoch ändern, wenn die Studierenden mit diesem Veranstaltungstyp mehr Erfahrung sammeln.

Die Umstellung des Lehrkonzepts auf das Inverted Classroom Modell verlief alles in allem vielversprechend, es wurde jedoch auch eine Reihe von Punkten deutlich, die bei zukünftigen Veranstaltungen verbessert werden sollten:

1. Sowohl für die Studierenden als auch für die Dozenten sind die neuen Rollen, die im ICM eingenommen werden sollen, ungewohnt. Um nicht in die klassischen Verhaltensmuster einer Präsenzvorlesung zurückzufallen, sollte die neue Lehr- und Lernstruktur zu Beginn deutlich kommuniziert, nachvollziehbar dokumentiert und später konsequent umgesetzt werden.
2. Die Verfolgung des Lernfortschritts der Studierenden muss konsequenter durchgeführt werden. Dies könnte zum Beispiel durch ein im Vorlesungsraum aufgestelltes Taskboard geschehen, auf dem alle Teilnehmer ihren aktuellen Arbeitsstand für alle transparent dokumentieren. Nur durch eine solch konsequente Verfolgung ist sichergestellt, dass die Präsenzphase allen Studierenden auch Nutzen bringt.
3. Damit die Studierenden sich stärker als selbständige Gestalter des Lernprozesses begreifen und weniger als reine Konsumenten der Inhalte, müssen in der Präsenzphase konsequent aktivierende Techniken (z. B. Entwicklung von MindMaps, selbstgeleitete Gruppendiskussionen, Bearbeitung von Fallbeispielen) eingesetzt werden.
4. Im konkreten Fall des Wahlpflichtfachs „Mobile Anwendungen“ ist an einigen Stellen aus didaktischer Perspektive eine Reduktion des Arbeitsaufwands wünschenswert. Dies kann etwa durch die Bereitstellung von Hilfen zur Lösung von besonders komplexen MOOC-Aufgaben geschehen.
5. Zur Verbesserung der Arbeitsbedingungen bei der Bearbeitung der MOOC-Inhalte im Rechnerraum sollte die Verwendung von Kopfhörern vorgeschrieben werden. So stören sich die Studierenden nicht gegenseitig und werden auch weniger von anderen gestört, die mit den Dozenten Fragen diskutieren.

Zusammenfassend lässt sich festhalten, dass die vollständige Integration eines MOOCs im Inverted Classroom Modell eine erfolgsversprechende, aber auch sehr weitgehende Veränderung des traditionellen Vorlesungskonzepts darstellt. Sie setzt voraus, dass ein Kurs gefunden wird, der sich inhaltlich, qualitativ und organisatorisch gut mit der eigenen Veranstaltung vereinen lässt. Aber auch weniger radikale Integrationsvarianten sind denkbar, etwa die Einbindung ausgewählter Lernvideos aus einem MOOC (diese stehen häufig zeitlich unbeschränkt auf YouTube zur Verfügung) in eine eher traditionelle Vorlesung oder die Nutzung des ICM auf Basis eines Lehrbuchs oder des eigenen Skripts.

8. QUELLEN

- Bergmann, J.; Sams, A. (2012): Flip Your Classroom: Reach Every Student in Every Class Every Day, Washington D. C, ISTE.
- Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals, Handbook I: Cognitive domain, New York, David McKay Company.
- Bremer, C. (2013): Massive Open Online Courses, in: Knaus, T.; Engel, O. (Hrsg.): fraMediale - digitale Medien in Bildungseinrichtungen [Band 3]. München, kopaed, S. 30 – 48.
- Gargenta, M. (2011): Einführung in die Android-Entwicklung, Köln, O'Reilly Verlag.
- Google (2013a): Developing Android Apps: Android Fundamentals, Udacity Inc., zugegriffen über: <https://www.udacity.com/course/developing-android-apps--ud853> am 16.01.2016.
- Google (2013b): Getting started. Android developer forum, Abschnitt „Training“, zugegriffen über: <http://developer.android.com/training/index.html> am 16.01.2016.
- Handke, J.; Schäfer, A. M. (2012): E-Learning, E-Teaching und E-Assessment in der Hochschule, München, De Gruyter Oldenbourg.
- Hill, P. (2012): Online Education Delivery Models: A Descriptive View, in: Educause Review, Heft 11/12-2012, S. 84 – 97.
- ICM Wiki (2015): Inverted Class Room Wiki, Berlin, Freie Universität Berlin, zugegriffen über: <http://wikis.fu-berlin.de/display/icm/Inverted+Classroom+Model> am 16.01.2016.
- Kück, A. (2014): Unterrichten mit dem Flipped Classroom-Konzept: Das Handbuch für individualisiertes und selbständiges Lernen mit neuen Medien, Mühlheim, Verlag an der Ruhr.
- Lage, M. J.; Platt, G. J.; Treglia, M. (2000): Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment, in: The Journal of Economic Education, Jg. 31, Heft 1, S. 30 – 43.
- NMC (2014): Flipped Classroom: Zeithorizont: ein Jahr oder weniger, in: NMC Horizon Report: 2014 Higher Education Edition, Austin, New Media Consortium, S. 48 – 50.
- Robes, J. (2012): Massive Open Online Courses: Das Potenzial des offenen und vernetzten Lernens, in: Hohenstein, A.; Wilbers, K. (Hrsg.): Handbuch E-Learning, Köln 2012 (Loseblattwerk, 42. Erg.-Lfg., Beitrag 7.21).
- Statista (2015): Vergleich der Marktanteile von Android und iOS am Absatz von Smartphones in Deutschland von Januar 2012 bis Oktober 2015, zugegriffen über: <http://de.statista.com/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-xabsatz-in-deutschland/> am 16.01.2016.

NORDAKADEMIE

Hochschule der Wirtschaft

Köllner Chaussee 11 · D-25337 Elmshorn

Tel.: +49 (0) 4121 4090-0 · Fax: +49 (0) 4121 4090-40

info@nordakademie.de · www.nordakademie.de