

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into

- seminars
- proceedings
- dissertations
- thematics

current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: <http://www.gi.de/service/publikationen/lni/>

ISSN 1617-5468

ISBN 978-3-88579-644-2

This volume contains papers from the conference “Projektmanagement und Vorgehensmodelle 2015” (PVM 2015) held in Elmshorn October 22 to 23, 2015. During the last decade, agile approaches disseminated in software development processes as well as in project management. Today, traditional and agile approaches are more and more combined within hybrid project management concepts that are applicable not only in software development & implementation projects. The conference papers discuss conceptual approaches on emerging standards and social aspects as well as practical experience in different application fields. New approaches, concepts and tools are discussed at the future track papers.



M. Engstler, M. Fazal-Baqaie, E. Hanser, M. Mikusz, A. Volland (Hrsg.)  
Projektmanagement und Vorgehensmodelle 2015

250



# GI-Edition

## Lecture Notes in Informatics

**Martin Engstler, Masud Fazal-Baqaie,  
Eckhart Hanser, Martin Mikusz,  
Alexander Volland (Hrsg.)**

## Projektmanagement und Vorgehensmodelle 2015

**Hybride Projektstrukturen erfolgreich  
umsetzen**

**Gemeinsame Tagung der Fachgruppen  
Projektmanagement (WI-PM) und Vor-  
gehensmodelle (WI-VM) im Fachgebiet  
Wirtschaftsinformatik der Gesellschaft  
für Informatik e.V., Elmshorn 2015**

## Proceedings





Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser,  
Martin Mikusz, Alexander Volland (Hrsg.)

## **Projektmanagement und Vorgehensmodelle 2015**

**Hybride Projektstrukturen erfolgreich umsetzen**

**Gemeinsame Tagung der Fachgruppen  
Projektmanagement (WI-PM) und  
Vorgehensmodelle (WI-VM)  
im Fachgebiet Wirtschaftsinformatik  
der Gesellschaft für Informatik e.V.**

**22. und 23. Oktober 2015  
in Elmshorn**

Gesellschaft für Informatik e.V. (GI)

## **Lecture Notes in Informatics (LNI) - Proceedings**

Series of the Gesellschaft für Informatik (GI)

Volume P-250

ISBN 978-3-88579-644-2

ISSN 1617-5468

### **Volume Editors**

Prof. Dr. Martin Engstler

Hochschule der Medien Stuttgart (engstler@hdm-stuttgart.de)

Masud Fazal-Baqaie

Universität Paderborn, s-lab – Software Quality Lab (masudf@uni-paderborn.de)

Prof. Dr. Eckhart Hanser

Duale Hochschule Baden-Württemberg Lörrach (hanser@dhbw-loerrach.de)

Dr. Martin Mikusz

Universität Stuttgart, GSaME (mikusz@wius.bwi.uni-stuttgart.de)

FOM Hochschule für Oekonomie und Management Stuttgart

Alexander Volland

Union IT-Services GmbH (alexander.Volland@union-investment.de)

### **Series Editorial Board**

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria

(Chairman, mayr@ifit.uni-klu.ac.at)

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Sigrid Schubert, Universität Siegen, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

### **Dissertations**

Steffen Hölldobler, Technische Universität Dresden, Germany

### **Seminars**

Reinhard Wilhelm, Universität des Saarlandes, Germany

### **Thematics**

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2015

**printed by** Köllen Druck+Verlag GmbH, Bonn

# Geleitwort

Agile Methoden sind in den letzten Jahren zunehmend in der Unternehmenspraxis angekommen. In Deutschland hat insbesondere Scrum eine große Verbreitung erlangt und erreicht nach Jahren der Diskussion und des Einsatzes an Hochschulen und innovativen, kleineren Firmen der Softwareentwicklungsbranche mittlerweile auch etablierte, große Unternehmen.

Die NORDAKADEMIE bildet als "Hochschule der Wirtschaft" im Bachelorbereich ausschließlich in dualen Studiengängen aus, die Theorieblöcke an der Hochschule mit Praxiserfahrungen bei der Ausbildung in Kooperationsunternehmen verbinden. Wir machen in den letzten Jahren verstärkt die Erfahrung, dass immer mehr Studierende in ihren Praxisphasen im Unternehmen agile Ideen kennenlernen.

Gerade in großen Unternehmen in klassischen Branchen wie der Banken- und Versicherungsindustrie ist dabei in der Regel eine Einbettung in nichtagile Kontexte mit plangeordneten Strukturen nötig. Daher sind hybride Ansätze in der Praxis so wichtig. Konzepte und Methoden zur nahtlosen Integration von agilen und nichtagilen Projekten in einem Unternehmen und von agilen und nichtagilen Techniken in einem Projekt sind entscheidend für die Akzeptanz neuer Ansätze im Management und bei weniger IT-affinen Mitarbeitern. Mit zunehmender Reife und breiterem Einsatz von agilen Methoden wird auch deutlich, dass sie nicht für alle Anwendungskontexte die beste Lösung darstellen und plangetriebene Methoden sehr wohl ihre Berechtigung haben können.

Im Konferenzprogramm der PVM2015 finden sich viele spannende Beiträge zu diesen und benachbarten Fragestellungen. Ich wünsche allen Lesern viel Freude und anregende Ideen beim Studieren dieses Tagungsbandes.

Joachim Sauer

NORDAKADEMIE Elmshorn



# Vorwort

Liebe Leserinnen und Leser,

agile Vorgehensmodelle haben sich in der Softwareentwicklung und in weiteren Anwendungsfeldern des Projektmanagements längst etabliert und werden erfolgreich neben klassischen Vorgehensmodellen eingesetzt. Die Grundsatzfrage lautet heute nicht mehr „klassisch oder agil“, vielmehr wird die Frage einer sinnvollen Integration klassischer und agiler Ansätze in den Projekten diskutiert und auch bereits in Praxis realisiert. Mit hybriden Ansätzen wollen die Unternehmen die Vorteile klassischer und agiler Projektmethoden gleichermaßen nutzen. Hierdurch soll ein flexiblerer Umgang mit noch unscharfen bzw. sich ändernden Anforderungen ermöglicht und gleichzeitig die Gesamteffizienz in den planbaren Projektphasen sichergestellt werden. Durch die hybriden Vorgehensmodelle wird das vorgehensmodellbezogene Gestaltungsspektrum in Projekten somit erweitert. Dennoch werden auch weiterhin rein klassische bzw. agile Vorgehensmodelle in den Projekten zum Einsatz kommen. Bei der Entscheidung und konkreten Ausgestaltung des Vorgehensmodells müssen projektbezogene Anforderungen ebenso wie die jeweiligen Einsatzbedingungen (z. B. die Qualifikationen, Erfahrungen sowie Einstellungen der Projektbeteiligten, einsetzbare Planungs- und Steuerungsinstrumente etc.) berücksichtigt werden.

Die GI-Fachgruppen Vorgehensmodelle (WI-VM) und Projektmanagement (WI-PM) stellen die Erfolgsfaktoren hybrider Projekte in den Mittelpunkt ihrer zweiten gemeinsamen Fachtagung Projektmanagement und Vorgehensmodelle PVM 2015. Ziel der Veranstaltung ist es, Erkenntnisse aus der Wissenschaft und Erfahrungen aus der Praxis einem Fachpublikum vorzustellen und Raum für die fachübergreifende Diskussion und den Erfahrungsaustausch zu geben. Die Fachtagung wird in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V. durchgeführt.

Die Fachtagung beleuchtet folgende Schwerpunkte in hybriden Projektstrukturen:

- Bewertung geeigneter Kombinations- und Integrationsansätze der Vorgehensmodelle;
- Bestimmung der erforderlichen Rollenprofile und Weiterentwicklung von Kooperationsmechanismen in der Projektarbeit;
- Beschreibung der Führungsaufgaben und Bestimmung relevanter Steuerungsgrößen
- Ansätze der Kompetenzvermittlung für die Rollenprofile und Führungskonzepte;
- Unterstützung hybrider Projekte durch geeignete Tools.

Die Fachtagung eröffnet mit der eingeladenen Keynote von Guido Zockoll zum Thema „Komplex oder doch nur kompliziert? Was Projektleiter von Schachspielern lernen können“. In der Keynote am zweiten Veranstaltungstag analysieren Paul Hilmer & Karl Pollok unter dem Vortragstitel „Agil & Klassisch - Projekt & Linie: Die Integration

vermeintlicher Gegensätze“ die Rahmenbedingungen und mögliche Integrationsansätze hybrider Projektstrukturen in der Praxis.

Das Hauptprogramm der Tagung umfasst acht ausgewählte Beiträge aus Praxis und Wissenschaft, die einen Review-Prozess durchlaufen haben. Wir möchten uns an dieser Stelle ausdrücklich bei den Mitgliedern des Programmkomitees bedanken, die durch ihre Gutachten der eingereichten Beiträge (Annahmequote 33%) erst einen objektiven Bewertungsprozess möglich machten.

Unter dem Leitthema der Tagung „Hybride Projektstrukturen erfolgreich umsetzen“ analysieren die Experten aus Wissenschaft und Wirtschaft in ihren Fachbeiträgen Integrationskonzepte und Methoden hybrider Vorgehensmodelle im Gesamtkontext des Projektmanagements. Hierzu gehören managementorientierte Aspekte wie die konzeptionelle Einordnung hybrider Vorgehensmodelle, die Beschreibung von Kriterien zur Auswahl eines Vorgehensmodells (z. B. mit Hilfe von Kennzahlen), die Erfolgsfaktoren bei der Umsetzung hybrider Projekte in der Praxis sowie die Auswahl und der Einsatz von spezifischen Methoden im Projektalltag oder in spezifischen Projektsituationen (z. B. öffentlicher Vergabeprozess, Projekte in anderen Kulturkreisen). Ergänzend dazu liefern die sieben ausgewählten „Future Track-Vorträge“ weitere Impulse in Form von (teilweise provokanten) Thesen und Vorstellung innovativer Konzepte, Methoden und Tools (mit Erfahrungstransfer), die mit dem Auditorium direkt oder in ergänzenden Open Spaces diskutiert und vertieft werden.

Ein großer Dank gilt den Hauptsponsoren *COPITOS GmbH* und *KOMTEAMER GmbH* sowie dem Medienpartner *Projekt Magazin*, die die Finanzierung der Tagung und des Tagungsbands erheblich erleichterten.

Unser Dank gilt auch der *NORDAKADEMIE*, die uns die Tagungsräumlichkeiten zur Verfügung stellt und eine inspirierenden Rahmen für die Vorträge und den interaktiven Diskurs bietet. Ein besonderer Dank gebührt hierbei Frau Elena Hermann für die engagierte Organisation der vielen Aufgaben vor Ort.

Wir hoffen, dass der vorliegende Tagungsband für Sie neue Erkenntnisse, authentische Erfahrungen und Anregungen enthält. Wir würden uns freuen, die eine oder andere Fragestellung auch in der GI-Fachgruppenarbeit zu vertiefen. Informationen zu Workshops, Terminen und Kontakten finden Sie auf den Internetseiten der Fachgruppe *Projektmanagement WI-PM* (<http://fg-wi-pm.gi.de/>) und der Fachgruppe *Vorgehensmodelle WI-VM* (<http://www.vorgehensmodelle.de/>).

Wir wünschen Ihnen allen eine anregende, erkenntnisreiche und unterhaltsame Veranstaltung in Elmshorn mit vielen spannenden Diskussionen, die unsere Überlegungen zum Thema Projektmanagement und Vorgehensmodelle sicherlich voranbringen werden.

Stuttgart, Paderborn, Lörrach und Frankfurt am Main im Oktober 2015

Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser,  
Martin Mikusz und Alexander Volland

## **Programmkomitee**

### **Vorsitz**

Prof. Dr. Martin Engstler, Hochschule der Medien Stuttgart

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

### **Mitglieder**

Dr. Volker Arendt, Bergische Universität Wuppertal

Dr. Martin Bertram, Commerzbank AG

Prof. Dr. Gerhard Chroust, J. Kepler Universität Linz

Masud Fazal-Baqaie, Universität Paderborn

Dr. Thomas Greb, Thomas Greb Consulting

Dr. Alexander Grimme, Union IT-Services GmbH

Elena Hermann, NORDAKADEMIE

Prof. Dr. Georg Herzwurm, Universität Stuttgart

Dr. Stefan Hilmer, Acando GmbH

Gerrit Kerber, aragon interactive GmbH

Prof. Dr. Ralf Kneuper, Beratung für Softwarequalitätsmanagement u. Prozessverbesserung

Dr. Marco Kuhrmann, University of Southern Denmark

Alexander Volland, Union IT-Services GmbH, Frankfurt am Main

Dr. Oliver Linssen, FOM Hochschule für Oekonomie und Management

Dr. Martin Mikusz, Universität Stuttgart

Günther Müller-Luschnat, iteratec GmbH

Prof. Dr. Andreas Oberweis, Universität Karlsruhe

Prof. Dr. Wolfram Pietsch, Fachhochschule Aachen

Prof. Dr. Joachim Sauer, NORDAKADEMIE

Prof. em. Dr.-Ing. Thorsten Spitta, Universität Bielefeld

Dr. Christa Weßel, Dozentin und Autorin

Prof. Dr. Doris Weßels, Fachhochschule Kiel

Prof. Dr. Frank Zimmermann, NORDAKADEMIE

## **Organisationskomitee**

Prof. Dr. Martin Engstler, Hochschule der Medien Stuttgart

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

Alexander Volland, Union IT-Services GmbH, Frankfurt am Main

## Veranstaltungspartner und Sponsor

NORDAKADEMIE, Elmshorn



Die NORDAKADEMIE gehört mit gut 1700 Studierenden zu den größten privaten Hochschulen mit Präsenzstudiengängen in Deutschland. Auf ihrem Campus in Elmshorn bietet sie vier duale Bachelorstudiengänge an.

Die NORDAKADEMIE Graduate School im Hamburger Dockland bietet Teilnehmern mit einem ersten Hochschulabschluss Weiterbildung auf Hochschulniveau. Zum Studienprogramm gehören aktuell sechs berufsbegleitende Masterstudiengänge, fünf Zertifikatskurse und eine Vielzahl an Weiterbildungsmodulen.

Zum April 2016 wird das Studienangebot um drei Masterstudiengänge erweitert und zum Herbst 2016 geht ein berufsbegleitendes Promotionsprogramm an den Start.

# Der beste Nachwuchs kommt aus den eigenen Reihen

Die NORDAKADEMIE ist die Hochschule der Wirtschaft. Sie wird als gemeinnützige AG von namhaften Unternehmen und dem Arbeitgeberverband NORDMETALL getragen. Schon über 700 Partnerunternehmen haben gemeinsam mit der NORDAKADEMIE ihren akademischen Nachwuchs qualifiziert.



Gemeinsam ausbilden an der  
NORDAKADEMIE Hochschule der Wirtschaft:

Gemeinsam fortbilden an der  
NORDAKADEMIE Graduate School:

#### Duale Bachelorstudiengänge

- Betriebswirtschaftslehre (B.Sc.)
- Wirtschaftsingenieurwesen (B.Sc.)
- Wirtschaftsinformatik (B.Sc.)
- Angewandte Informatik (B.Sc.)

#### Berufsbegleitende Masterstudiengänge

- Master of Business Administration (MBA)
- General Management (M.A.)
- Marketing and Sales Management (M.A.)
- Financial Management and Accounting (M.Sc.)
- Wirtschaftsinformatik/IT-Management (M.Sc.)
- Wirtschaftsingenieurwesen (M.Sc.)
- Wirtschaftspsychologie (M.Sc.) – neu ab 1.4.2016
- Logistik (M.Sc.) – neu ab 1.4.2016
- Wirtschaftsrecht (LL.M.) – neu ab 1.4.2016

#### Weiterbildung mit Hochschulzertifikat

- Management-Grundlagenmodule
- Management-Vertiefungsmodule
- Zertifikatskurse

#### Neu ab Herbst 2016:

#### Berufsbegleitendes Promotionsprogramm



Köllner Chaussee 11 · 25337 Elmshorn  
info@nordakademie.de · www.nordakademie.de



Van-der-Smissen-Straße 9 · 22767 Hamburg  
master-office@nordakademie.de · www.nordakademie-gs.de

# Hauptsponsor

KOMTEAMER GmbH, Frankfurt am Main



- Sourcing Management
- Projekt Management
- IT Prozess Beratung
- IT Service Management

## DIE SICHERHEIT DER BESTEN LÖSUNG

Als „KOMpetenzTEAMbuilder“ begleiten wir unsere Kunden von der Planung bis zur Umsetzung - in den Schwerpunkten Sourcing, Projekt Management, Prozess Beratung und Service Management.

[www.komteamer.com](http://www.komteamer.com)

## **Hauptsponsor**

COPiTOS GmbH, Berg / Ravensburg

# COPiTOS

COPiTOS versteht sich als zuverlässiger Partner in sämtlichen Disziplinen des Projektmanagements. Durch unsere jahrelange Projekterfahrung sind wir sowohl in klassischen als auch agilen Vorgehensweisen methodensicher. Auch nach erfolgreichem Projektabschluss stehen wir gerne mit geeignetem Fachpersonal an Ihrer Seite.

Ein weiterer Schwerpunkt liegt in der individuellen Entwicklung von Software auf Basis von Microsoft Technologien.

**Ausführliche Informationen zu unserem Leistungsspektrum finden Sie unter [www.copitos.de](http://www.copitos.de).**

## Medienpartner

Projekt Magazin, Taufkirchen

The logo for ProjektMagazin is displayed on a dark grey rectangular background. The word "Projekt" is written in a bold, white, sans-serif font, while "Magazin" is in a smaller, regular weight of the same font. The two words are joined together.

**ProjektMagazin**

Das Fachportal für Projektmanagement

**Projekt Magazin** ist das Online-Fachportal zum Thema Projektmanagement für Projektleiter, Projektmitarbeiter und Berater. Unter [www.projektmagazin.de](http://www.projektmagazin.de) finden Sie praxisnahe Artikel und konkrete Unterstützung für Ihre Projekt-Aufgaben. Im Projekt Magazin schreiben Experten für Experten: In über 1800 Fachartikeln, Praxisberichten und Software-Besprechungen können Sie sich über aktuelle Trends und Entwicklungen im Projektmanagement informieren. Das Projekt Magazin erscheint mittwochs alle zwei Wochen.

# Inhaltsverzeichnis

## Teil I – Hauptprogramm

<b>Helge F. R. Nuhn, Bernhard Frühlinger, Daniel Klumpp</b> <i>Agile Management-Praktiken in Saudi-Arabien – Methodenwahl und Toolunterstützung</i> .....	19
<b>Ralf Kneuper</b> <i>Klassische und agile Vorgehensmodelle – Ein historischer Überblick</i> .....	29
<b>Anna Aldushyna, Martin Engstler</b> <i>Erfolgsfaktoren bei der Umsetzung hybrider Projekte – Ergebnisse einer Befragung und praktische Empfehlungen zur Umsetzung</i> .....	39
<b>Christian Hennen, Axel Kalenborn, Sascha Stadlbauer, Ingo Timm</b> <i>Systematisierung der Auswahl von Vorgehensmodellen durch Kennzahlen</i> .....	55
<b>Thomas Süptitz, Felix Ruppert, Torsten Eymann</b> <i>IT-Systementwicklungsprojekte der öffentlichen Hand: Der Einfluss des Vergaberechts auf die Verwendung agiler Methoden</i> .....	67
<b>Dimitri Petrik</b> <i>Hybride Vorgehensmodelle in der Versionserstellung – ein Praxisbeitrag</i> .....	75
<b>Wolfram Müller</b> <i>Hybrid ist Pflicht – mit Ultimate/Reliable Scrum und Critical Chain zu einer hochskalierbaren agilen Projektorganisation</i> .....	93
<b>Alexander Krieg</b> <i>Bierdeckelskizzen – Scrum ist leicht aber nicht einfach</i> .....	107

## Teil II – Eingeladene Beiträge der Session „Future Track“

<b>Philipp Diebold, Steffen Küpper, Thomas Zehler</b> <i>Nachhaltige Agile Transition: Symbiose von technischer und kultureller Agilität</i> .....	121
<b>Torsten J. Koerting</b> <i>Erkenntnisse und Impulse eines Projektmanagers</i> .....	127
<b>Christina Klüver, Jürgen Klüver</b> <i>Self-Enforcing Networks als Tools zur Auswahl eines geeigneten (ggf. hybriden) Vorgehensmodells in IT-Projekten</i> .....	139

<b>Sixten Schockert, Georg Herzwurm</b> <i>Das Business setzt die Prioritäten?! .....</i>	151
<b>Patrick Daut</b> <i>agile@scale: Do more with LeSS or be SAFe? Ansätze zur Skalierung – ein Überblick.....</i>	159
<b>Martin Jud, Jörg Hofstetter</b> <i>Softwareentwicklung im Maschinenbau – ein kooperativer Ansatz .....</i>	169
<b>Klaus Schopka</b> <i>Controlling von hybriden Projekten – Herausforderungen und Chancen .....</i>	183

Teil I

Hauptprogramm



# Agile Management-Praktiken in Saudi-Arabien – Methodenwahl und Toolunterstützung

Helge F. R. Nuhn<sup>1</sup>, Bernhard Frühlinger<sup>2</sup> und Daniel Klumpp<sup>3</sup>

**Abstract:** Klassische und agile Methoden können in einem Projekt durchaus kombiniert verwendet werden. Ein effizienter Methodenwechsel bringt auch in laufenden Projekten Struktur in die Selbststeuerung eines Teams. Bemerkenswert ist die Tendenz von Teams, eigene Rollen, Methoden und Arbeitsprozesse zu definieren. Die Basis für eine erfolgreiche Zusammenarbeit innerhalb der Gesamtprojektorganisation bilden Tools und Plattformen zum Austausch von Ergebnissen sowie zur Kommunikation und Planung. Der Einsatz von modernen Kommunikationsinstrumenten ist ebenso relevant für den Projekterfolg wie die Auswahl geeigneter Methoden und Prozesse und wird daher stets vom Projektteam diskutiert und bewertet. Die Projektleitung muss dabei in den Methoden-Wechsel-Prozessen moderierend einwirken und dem Team den notwendigen Freiraum für die eigene Arbeitsoptimierung verschaffen.

Im Rahmen der Durchführung eines Projekts im Gesundheitsministerium von Saudi-Arabien entstand aufgrund hohen Zeitdrucks und der Anforderung zur raschen Lieferung von Teilergebnissen die Notwendigkeit, die ursprünglich gewählte Projektmethode zu flexibilisieren und mit Elementen des agilen Projektmanagements anzureichern. Wesentliche Erfolgsfaktoren lagen in der Schaffung eines einheitlichen Qualitätsverständnisses, einer gemeinsam verabschiedeten Definition-of-Done sowie in der Ermöglichung von Spezialisierung und Selbstorganisation innerhalb des Teams. Die Erfahrungen aus diesem Projekt zeigen, dass Ansätze aus agilen Paradigmen in Kombination mit klassischen Vorgehensweisen auch in konzeptionellen Projekten (ohne IT-Umsetzung) kraftvolle Hebel für stärkere Ergebnisorientierung, kontinuierliche Verbesserung und höhere Teameffizienz sind.

**Keywords:** Agile Methoden, Scrum, Reporting, Performance Management, Lean Ansatz, Projekt-Tools

## 1 Einleitung

Saudi-Arabien wurde im Jahr 2014 vom gefürchteten Corona Virus heimgesucht. Im Zuge des Krisenmanagements des Gesundheitsministeriums wurden deutliche Schwachstellen im Healthcare-Bereich aufgedeckt. Der damalige Gesundheitsminister musste im Zuge der Krise seinen Posten räumen. Das Ministeramt wurde interimsmäßig vom amtierenden Arbeitsminister übernommen.

---

<sup>1</sup> Horváth & Partners, CIO & Project Advisory, Mainzer Landstraße 41, 60329 Frankfurt a.M.,  
hnuhn@horvath-partners.com

<sup>2</sup> Horváth & Partners, CIO & Project Advisory, Ganghoferstraße 39, 80339 München,  
bfruehlinger@horvath-partners.com

<sup>3</sup> Horváth & Partners, CIO & Project Advisory, Königstraße 5, 70173 Stuttgart,  
dklumpp@horvath-partners.com

Wie für jeden Politiker sind die ersten 100 Tage im Amt eine Zeit der Bewährung, der Veränderung und der Bewegung. Der Interims-Gesundheitsminister setzte ein Projekt mit einer Vielzahl von Beratern auf, um als fachfremde Person einen schnellen Überblick über das Gesundheitsministerium und das Gesundheitswesen im gesamten Königreich zu erlangen.

Die Autoren waren an diesem Projekt beteiligt. Sie und fünf weitere Kollegen waren für den Performance Management Workstream zuständig. Der Workstream hatte unter anderem zum Ziel, die Leistungserbringung des Ministeriums mit Hilfe von Kennzahlen transparent zu machen. Durch die Dringlichkeit der Lage waren unkonventionelle Methoden in der Ergebniserarbeitung erforderlich, um die hohen Erwartungen seiner Exzellenz, des amtierenden Gesundheitsministers, zu erfüllen. Gemeinschaftlich entschloss sich das Projektteam, agile Methoden und Praktiken anzuwenden, um die eigene Lieferfähigkeit früh und kontinuierlich unter Beweis stellen zu können.

Dieser Artikel berichtet über die Erfahrungen, die die Berater während des Projekts gesammelt haben. Bemerkenswert ist, dass hierbei unterschiedliche agile Methoden verwendet wurden, dass keine IT-Implementierung im Fokus agiler Methoden stand und dass das vergleichsweise junge, internationale Team in modernen Kommunikationstechnologien wertvolle Unterstützung der agilen Prozesse fand.

## **2 Vorgegebenes Projektumfeld**

Das Projekt und insbesondere der Performance Management Workstream waren in ein sehr spezielles Umfeld eingebettet. Dies trifft sowohl auf das Makro-Umfeld (Saudi-Arabien und die islamische Kultur) als auch auf das Mikro-Umfeld (Organisationsstruktur und -kultur im saudi-arabischen Gesundheitsministerium sowie die Projektteamstruktur) zu.

Für Deutsche ist die Projektarbeit im Nahen Osten sehr stark von interkulturellen Differenzen in Bezug auf Identität, Werte, Grundannahmen, Kompetenzen und Verhaltensmuster geprägt. Die Berücksichtigung dieser Elemente ist dabei jedoch ein wesentlicher Erfolgsfaktor. Für das beschriebene Vorhaben ergab sich daher die Notwendigkeit, übliche Projektansätze kritisch zu hinterfragen und an die kulturellen Gegebenheiten und Anforderungen von Saudi-Arabien anzupassen.

Weitere Unsicherheit im Projektumfeld rührte aus der bereits vom ausgeschiedenen Minister begonnenen organisatorischen Transformation des Gesundheitsministeriums. In diese Situation musste sich der neue Gesundheitsminister, ein Neueinsteiger im Gesundheitswesen, einfügen. Im Gegensatz dazu waren die meisten Positionen der zweiten Managementebene - der Director Generals - im Ministerium von ehemals praktizierenden Ärzten bereits seit längerer Zeit besetzt. Diese wiederum wiesen wenig Erfahrungen mit und Begeisterung für Performance-Messungen auf Basis von gegebenen und messbaren Zielen auf. In diesem Spannungsfeld ergab sich für das Projekt ein kontinuierli-

cher Balanceakt zwischen der Erfüllung des Projektauftrags von seiner Exzellenz sowie dem Aufbau und der Erhaltung von guten Beziehungen zu den Director Generals.

Die Kernteamstruktur stellte einen weiteren Komplexitätsfaktor im Projekt dar. Die Rolle des Projektponsors wurde vom “Assistant Deputy Minister of Planning & Training” übernommen. Das Projektteam wurde auf Kundenseite durch die “Decision Support Unit” (DSU) gestellt. Die DSU bestand aus drei Frauen und zwei Männern, die medizinisches Grundwissen, Kenntnisse des Gesundheitswesens und analytische Fähigkeiten besitzen. Das Vorhandensein eines internen Teams war von nicht zu unterschätzender Bedeutung: In einer Kultur wie der Saudi-Arabischen, in welcher Hierarchie, Macht, Vertrauen und Seniorität die wesentlichen Anreize für Zusammenarbeit darstellen, ist es beinahe unmöglich, Meetings oder nutzbare Informationen ohne die Unterstützung eines in der Organisation bekannten und respektierten Teams zu bekommen.

Unter Berücksichtigung des oben beschriebenen Projektumfelds werden im folgenden Abschnitt der ursprüngliche Projektansatz, dessen Unzulänglichkeiten und die wesentlichen Gründe für den Wechsel zu einer maßgeschneiderten, agilen Projektmethodik skizziert.

### **3 Ursprünglicher Projektansatz und Gründe für den Einsatz agiler Methoden**

Das prominenteste und visibelste Lieferergebnis des Performance Management Workstreams war ein monatlicher, manuell erstellter Management-Bericht, welcher die wesentlichen Kennzahlen zu ausgewählten Fokusthemen beinhaltete. Das Projekt beinhaltete keine IT-Implementierung. Nichtsdestotrotz war die Erwartungshaltung seiner Exzellenz an den Report beträchtlich, da das Reporting als wesentliches Instrument zur Schaffung von Transparenz und zur Entscheidungsfindung in den “First 100 Days” genutzt werden sollte. Neben der Projektstätigkeit (Berichtskonzeption, Automatisierung, Belegung der Implementierbarkeit in BI-Systemen) sollte das Projektteam als interimistische Reporting-Abteilung auch die operativen Aufgaben der Datensammlung, Berichterstellung, Kommentierung und des Maßnahmentrackings übernehmen.

Der ursprüngliche Projektansatz sah eine klassische Projektmethodik vor. Zuerst sollten im Rahmen einer Bestandsaufnahme alle bestehenden Berichte und Kennzahlen gesammelt, konsolidiert und auf Relevanz geprüft werden. Anschließend würde ein vollumfängliches Ziel-Berichts-Framework inklusive Best-Practice-Kennzahlen konzipiert und mit den relevanten Stakeholder abgestimmt werden. Dies entspricht im Wesentlichen einer sehr verkürzten Fachkonzeptionsphase, die dann mittels Maßnahmen zur Datenerhebung und Datenqualitätsverbesserung umgesetzt werden sollte. Diese „klassische“ Vorgehensweise bedingte, dass Projektergebnisse erst nach einer gewissen Projektlaufzeit - zur Erreichung definierter Meilensteine - geliefert würden. Die Anforderung seiner Exzellenz an das Projekt war jedoch, zum frühestmöglichen Zeitpunkt nutzbare Ergebnisse für die Entscheidungsfindung zu erhalten. In den Regelabstimmungen mit dem

Projektsponsor wurde dem Team daher bald klar, dass die gewählte Vorgehensweise diesen Anspruch kurzfristig nicht erfüllen kann. Aufgrund steigender Unzufriedenheit seitens des Kunden und der Gefahr eines Scheiterns des Projekts sah sich das Projektteam gezwungen, den gewählten Projektansatz in Richtung der Erfüllung folgender drei Kernanforderungen anzupassen:

- Rasch nutzbare Ergebnisse für Entscheidungen (nicht zwingend 100%-Ergebnis)
- Kontinuierliche Verbesserung und dynamische Erweiterung des Lieferergebnisses basierend auf inhaltlich-funktionalen Anforderungen
- Zuverlässige, regelmäßige Bereitstellung des Lieferergebnisses

Die Analyse dieser Kernanforderungen zeigt auf, dass sie typischen Anforderungen an agile Softwareentwicklungsprojekte stark ähneln. Diese Erkenntnis führte zu der Entscheidung, die bekannten Methoden der agilen Softwareentwicklung für ein IT-unabhängiges Projekt, in diesem Fall das visuelle Design und die inhaltliche Ausgestaltung von Top-Management-Reports, zu adaptieren [Sc01].

#### **4 Aufbau eines agilen Projekt-Set-Up mit Schnittstellen zu vorhandenen klassischen Methoden**

Das Projekt-Set-Up gliederte sich in zwei Teilstrukturen. Während das „externe Set-Up“ in Bezug auf Kommunikation mit den Anforderern (seine Exzellenz, Programmleitung, Project Office) Elemente eines klassischen Projekt-Set-Ups mit regelmäßiger Statusberichterstattung und Anforderungsmanagement hatte, entwickelte sich das “interne Set-Up” zu einer agilen Projektorganisation [Ko14]. Folgende Abbildung stellt die Gesamtprojektstruktur schematisch dar.

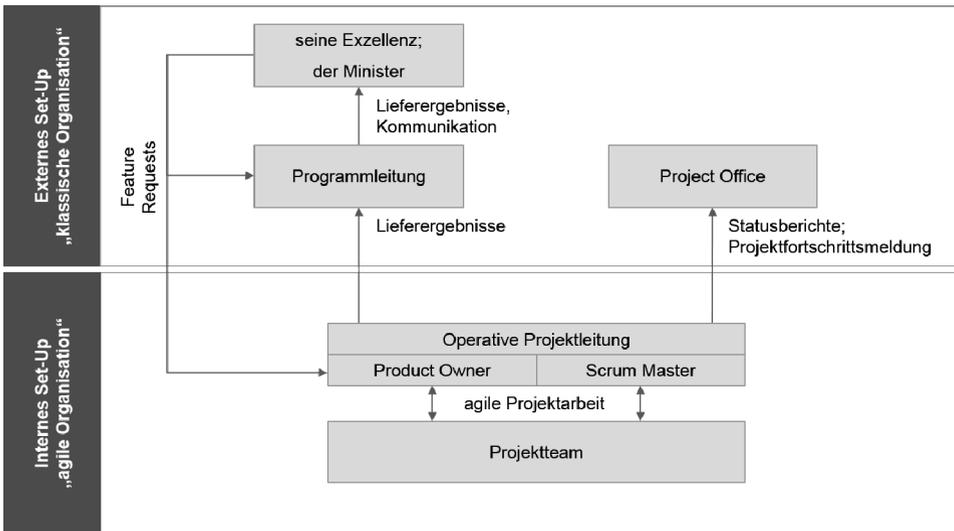


Abb. 1: Gesamtprojektstruktur

Innerhalb kurzer Zeit wurden innerhalb des Projektteams Rollen und Verantwortlichkeiten definiert und zugewiesen, wie sie sich in der Scrum-Methodologie wiederfinden: Product Owner, Scrum Master und Teammitglieder. In einem ersten Meeting wurden sämtliche Beteiligte für die Relevanz eines gemeinsamen Qualitätsverständnisses sensibilisiert. Hinsichtlich der Projektsituation war es wichtig, eine gemeinsame Definition-of-Done zu finden. Sofern bei Fertigstellung eines Sprints die Möglichkeit gegeben war, das Produkt - also den Management-Bericht - seiner Exzellenz und seiner Führungsriege ohne weitere Vorbehalte zukommen zu lassen, war das Produkt in der angestrebten Qualität erstellt. Diese vom finalen Lieferergebnis her gedachte Qualitätsanforderung führte zu einem einheitlichen Qualitätsverständnis innerhalb des Teams. Diese Denkweise war eine wichtige Leitplanke bei der kontinuierlichen Verbesserung der teaminternen Vorgänge. Zusätzlich waren sowohl Definition-of-Done als auch vorgelagerte, unterschiedlich ausdefinierte niedrigere Qualitätsniveaus ein wertvolles Mittel zur Abstimmung der agilen Vorgehensweise mit der tradierten Projektmanagement-Sichtweise, welche insbesondere durch das Program Management Office eingefordert wurde. Diese Mehrstufigkeit einer Definition-of-Done erwies sich damit als äußerst praktische Verbindung zwischen der agilen und der nicht-agilen Projektmanagement-Welt.

Im Projektverlauf konnte beobachtet werden, wie sich eine emergente Struktur des Projektteams bildete und zunehmend weitere Rollen durch das Team geschaffen wurden. Primärer Treiber für diese Emergenz war die Identifikation von Fähigkeiten unter den Teammitgliedern. So wurde dem Team - gefördert durch eine gute Kommunikationskultur - wesentlich schneller deutlich, welche Ressource sich für welche Aufgaben am besten eignete, als dies bei einer herkömmlichen Projektrealisierung der Fall gewesen wäre. Einzelne Teammitglieder entwickelten sich folglich schnell zu Experten für spezifische

Fragestellungen. Sowohl Product Owner als auch Scrum Master mussten diese Spezialisierungen im Auge behalten, um die Leistungsfähigkeit des Teams sowie den benötigten Aufwand für anstehende Aufgaben stets realistisch einschätzen zu können [SS01].

Das Team strukturierte sich und die eigene Arbeitsweise selbst. Intensiv wurde nach Möglichkeiten gesucht, Standardisierbares auch selbstgemachten Standards zu unterwerfen. Es trat dabei eine Bestrebung zur Industrialisierung diverser Aufgaben auf. Während Lean Startup und Design Thinking zu früherer Zeit Ansätze bei der Konzeption unseres Produkterstellungsprozesses gewesen wären, und so wie Scrum der passende Ansatz zur Erarbeitung neuer Inhalte des Reports war, so war ein Ansatz, der Anleihen bei Lean und Kanban suchte, der richtige, um die monatlich wiederkehrende Aktualisierung von bereits etablierten Reportinhalten effizient zu gestalten.

Die Projektorganisation wanderte merklich durch die entsprechenden Phasen einer agilen Einführung, die viele Organisationen in Teilen oder im Ganzen bereits durchgemacht haben. Aus der Steuerungsperspektive war es wichtig, stets genügend Raum (und auch Kapazität) zur Selbstentfaltung für das Team zu lassen, wenn der Wunsch nach einer Veränderung der Arbeitsmethoden artikuliert wurde. Dies bedeutete im Einzelfall, auch den Verzicht auf die Lieferung von Features, um im Gegenzug durch das Team eine effizientere zukünftige Lieferung ähnlicher Features garantiert zu bekommen. Dieses Phänomen lässt sich mit „technical debt“ in der agilen Software-Entwicklung vergleichen.

Regelmeetings mit hoher Akzeptanz innerhalb des Projektteams waren insbesondere Daily-Stand-Up-Meetings und Sprint-Retrospektiven. Hier wurden die wichtigsten Effizienzsteigerungen erreicht und eine zielgerichtete Kommunikation gefördert. Die Stand-Up-Meetings stießen auf große Resonanz. Nach wenigen Sprints gesellten sich die Mitglieder des DSU-Teams zu den täglichen Kurzmeetings hinzu. Fasziniert von der offenen Arbeitsweise begannen sie, am agilen Prozess maßgeblich teilzunehmen. Diese untradierte Arbeitsweise führte jedoch mitunter zu kulturell bedingten Herausforderungen, die die Projektleitung aktiv steuern musste. Zusätzlich ist anzumerken, dass durch den Zuwachs an Personen die oft zitierten Grenzen einer nicht-skalierten agilen Herangehensweise schnell erreicht werden [KS14]. Stand-Up-Meetings mit mehr als acht Personen konnten nicht mehr die gewohnte Schlagkraft entwickeln, sodass fortan bereits im Vorfeld Teams ausgegründet wurden, die sich selbst organisierten und von denen ein Vertreter im Stand-Up-Meeting Aussagen traf und Fragen stellte.

Von hoher Bedeutung für die erfolgreiche Projektdurchführung war die visionsgetriebene Zielsetzung, welche gleichzeitig ein allgemeingültiges Qualitätsverständnis erforderlich machte. Qualität und Vorgehensweisen wurden innerhalb des Teams intensiv diskutiert, teils mit und teils ohne Product Owner und Scrum Master. Eine wichtige Leitlinie für diese Art von Diskussionen war, dass die Dauer der Diskussion im angemessenen Verhältnis zur erwartbaren Qualitätssteigerung stehen musste. Dies zuzulassen ist eine schwerwiegende Herausforderung für einen konventionellen Projektleiter, welcher es möglicherweise gewohnt ist, entsprechende Vorgaben zu machen. Im agilen Prozess ist

es von entscheidender Bedeutung, dass der Projektleiter seine Rolle neu interpretiert und beginnt, als "Schmierschicht" zwischen Anforderern und dem liefernden Team zu fungieren [A111][AM05].

#### 4.1 Eingesetzte Projekt-Tools zur Unterstützung der agilen Methoden

Im Zuge des Projektverlaufs wurden verschiedene Tools angewandt. Die Auswahl der Tools richtete sich stets nach den durch das Team wahrgenommenen Notwendigkeiten und war regelmäßig Gegenstand von Daily-Stand-Up-Meetings oder Sprint-Retrospektiven. Die folgende Abbildung zeigt eine Übersicht der im Projekt verwendeten Tools.

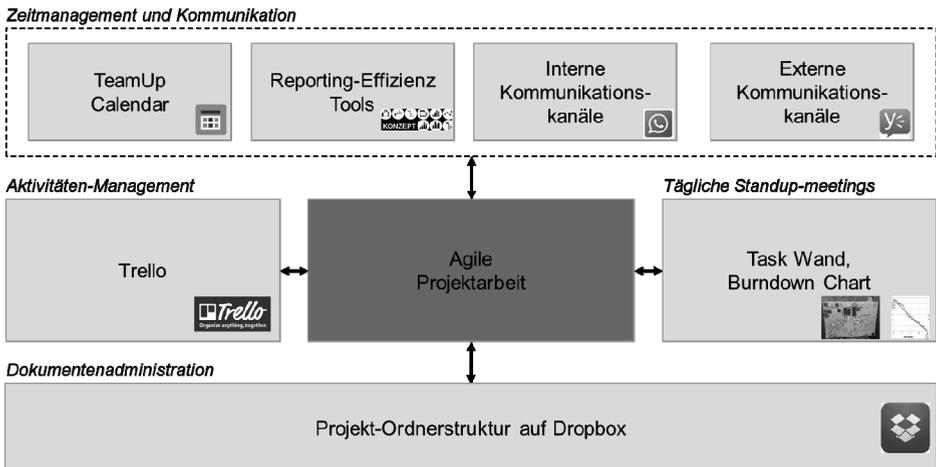


Abb. 2: Übersicht über den Tool Einsatz im Rahmen des Projekts

#### Scrum Board

Zunächst war, gemäß den Standards von Scrum, ein Scrum Board das zentrale Werkzeug bei der Arbeitsorganisation. Auch dieses erfuhr im Verlauf der Projektzeit stete Erweiterungen. Beispielsweise wurden zusätzliche Elemente eingefügt: Ein Velocity-Zähler, der die Summe der abgearbeiteten Story-Points im vergangenen Sprint darstellte; eine Review-Uhrzeit, damit das Team sich auf einen definierten Zeitpunkt zur Konsolidierung verständigen konnte; ein spezieller Bereich für blockierte Stories und letztlich ein zusätzlicher Bereich für Stories, die „after sprint“, also nach dem eigentlichen Review-Zeitpunkt, erledigt werden sollten. Speziell Letzteres war den speziellen Rahmenbedingungen von Beratungsprojekten geschuldet, bei welchen Informationen typischerweise über den Verlauf eines konventionellen Geschäftstages gesammelt und erst im Nachgang verarbeitet werden.

## **TeamUp**

Bereits nach der ersten offiziellen Lieferung des Reports an den Kunden analysierte das Team wiederkehrende Aktivitäten. Eine zeitliche Ausplanung dieser Aktivitäten erfolgte im Online-Tool Teamup. Das Tool ist ein Online-Kalender ohne nennenswerte Hürden für Anmeldung und Administration. Auf diese Art konnten auch externe Teammitglieder ohne administrativen Aufwand jederzeit einen aktuellen Überblick über die zeitliche Planung der Aktivitäten erlangen.

## **Trello**

Wichtig für die Kollaboration wurde im späteren Projektverlauf eine Online-Kollaboration, die sowohl zeitlich als auch inhaltlich komplexere Daten aufnehmen und wiedergeben konnte. Das Team fand in Trello eine hervorragend geeignete Plattform, um Aktivitäten selbständig zu definieren, zu clustern und zeitlich einzuordnen. Color-Codings und Themenblöcke erlaubten einen schnellen Überblick über anstehende Probleme. Die Option, Trello anstelle des Scrum Boards für die tägliche Bearbeitung des Backlogs zu verwenden, wurde vom Team jedoch übereinstimmend abgelehnt, da die Erfahrungen der physischen Präsenz eines aktuellen Projektplans bereits sehr positiv war. Jedoch ersetzte Trello ein Kanban-Board, das zunächst für die Darstellung der anstehenden, wiederkehrenden Aufgaben selbstständig durch das Team eingerichtet wurde. Zur Sicherstellung des Datenschutzes wurden eigens definierte Kürzel zur Aktivitätenbeschreibung verwendet und auf die Nennung von Personen verzichtet.

## **Digitale Kommunikationskanäle für das Projektmarketing (WhatsApp, Yammer)**

Auch von außerhalb des Projektteams kamen Einflüsse, die die Teamarbeit nennenswert beeinflussten. Das vergleichsweise junge Team scheute dabei nicht davor zurück, neue Kommunikationskanäle wie WhatsApp oder Yammer zu verwenden. Dies war für die Integration zwischen deutschen und saudi-arabischen Teammitgliedern ein besonderer Vorteil, wenn nicht gar eine Voraussetzung für Effizienz. In Saudi-Arabien sind diese neuen Kommunikationskanäle im Geschäftsalltag bereits jetzt wesentlich weiter verbreitet als in Deutschland. Dabei ließen sich Befürchtungen hinsichtlich des Schutzes sensibler Daten durch einfache Übereinkünfte zu Kommunikationsregeln zerstreuen. Beispielsweise war eine Grundregel für die interne Kommunikation über WhatsApp, das Medium lediglich zur Teamorganisation und -kommunikation und ohne Bezug zur inhaltlichen Projektarbeit zu benutzen. [AM05]

Entscheidend für den Einsatz von Tools ist die kontinuierliche Reflexion des Teams über die Notwendigkeit und den gestifteten Nutzen. Ein wahrlich agiles Team wird ohne Bedenken die Nutzung eines Tools einstellen, wenn es zu der Erkenntnis gelangt, dass dieses eher hindert als hilft.

## 5 Fazit

Agile Methoden eignen sich auch für nicht-IT-fokussierte Projekte. Wichtig ist, dass sichtbare Erfolge errungen werden können, wozu sich auch jedes digital-visuelle Produkt eignet.

Agile Methoden sind aber selten die Antwort auf alle Fragen, wenn sie im Projektmanagement zur Anwendung kommen. Die Projektleitung ist das verpflichtende Bindeglied zwischen der strategischen Planung eines Vorhabens, welche in der Regel nach klassischen Vorgehensweisen und Methoden stattfindet, und der operativen Agilität. Wenn agile Methoden konsequent zur Anwendung kommen, ist folglich das Team Entscheider über die Methodenauswahl in der eigenen Arbeit. Dies erklärt, wieso die Anwendung agiler Methoden in kleineren Arbeitspaketen aufrechterhalten wurde und gleichzeitig adaptierte Methoden Einzug in den Projektalltag fanden.

Letztlich sind es moderne Kommunikationsinstrumente, die genauso agil zur Verwendung kommen können wie Methoden und Prozesse. Wichtig in den Augen der Autoren ist die undogmatische Betrachtung jeder neuen Option sowie eine Struktur, in der agile Methoden nach Bedarf des Teams angewendet werden können.

## Literaturverzeichnis<sup>4</sup>

- [Al11] Alberts, D. S.: The Agility Advantage - A Survival Guide for Complex Enterprises and Endeavors. Information Age Transformation series, CCRP Publication Series, U.K. Department of Defense, 2011.
- [AM05] Atkinson, R.; Moffat, J.: The Agile Organization - from informal networks to complex effects and agility. Information Age Transformation series, CCRP Publication Series, U.K. Department of Defense, 2005.
- [Ko14] Kolb, C.: Der Agile Projektleiter - Bindeglied zwischen SCRUM und klassischem Umfeld. ProjektMagazin, 23/2014.
- [KS14] Knittel, O.; Steckinger, O.: Wer Scrum einführt, muss auch agil werden. ProjektMagazin, 19/2014.
- [Sc01] Scrum like it hot!. Der F&E Manager, 01/2014, S. 6-11.
- [SS01] Sutherland, J.; Schwaber, K.: The Scrum Guide™. <http://www.scrumguides.org>, Stand: 07.08.2015, 2013.

---

<sup>4</sup> Die in diesem Artikel erwähnten Unternehmens-, Produkt- oder Markenbezeichnungen können Marken oder eingetragene Markenzeichen der jeweiligen Eigentümer sein. Die Wiedergabe von Marken- und/oder Warenzeichen in diesem Artikel berechtigt nicht zu der Annahme, dass diese als frei von Rechten Dritter zu betrachten seien. Alle erwähnten Marken- und/oder Warenzeichen unterliegen uneingeschränkt den länderspezifischen Schutzbestimmungen und den Besitzrechten der jeweiligen eingetragenen Eigentümer.

- [YM01] Yu, L.; Mishra, A.: An Empirical Study of Lehman's Law on Software Quality Evolution in International Journal of Software and Informatics, 11/2013; 7(3):469-481.

# Klassische und agile Vorgehensmodelle – Ein historischer Überblick

Ralf Kneuper<sup>1</sup>

**Abstract:** Dieser Beitrag gibt einen Überblick über die Unterschiede und Gemeinsamkeiten von klassischen und agilen Vorgehensmodellen, um diese besser verstehen und nachvollziehen zu können. Insbesondere wird erläutert, woher die beiden Ansätze kommen und welche externen Einflüsse auf die Entwicklung eingewirkt haben. Bei beiden Ansätzen sind die zu Grunde liegenden Ideen mittlerweile fast 60 Jahre alt, auch wenn die systematische Auseinandersetzung mit klassischen Vorgehensmodellen erst etwa Mitte der achtziger Jahre begann, mit agilen Vorgehensmodellen etwa zehn Jahre später.

**Keywords:** klassische Vorgehensmodelle, agile Vorgehensmodelle, Phasenmodell

## 1. Einleitung

Softwareprozesse und ihre Repräsentationen in Vorgehensmodellen haben im Laufe der Zeit erhebliche Veränderungen erfahren. Dieser Beitrag gibt einen Überblick über diese Entwicklung, mit besonderem Augenmerk auf den Ansätzen der „klassischen“ oder Plan-getriebenen Entwicklung einerseits und der agilen Entwicklung andererseits.

Eine andere Sichtweise auf zumindest einen Teil dieser Entwicklung hat der Autor in [Kn13] beschrieben.

## 2. Die „Frühzeit“ der Vorgehensmodelle

Erste Überlegungen zum Vorgehen bei der Softwareentwicklung entstanden von Beginn an, wobei die Softwareentwicklung sich in dieser Frühzeit weitgehend auf die Programmierung beschränkte.

Beispielsweise hat schon Alan Turing in [Tu51, Abschnitt „Programming Principles“] folgendes aus heutiger Sicht sehr einfache Vorgehen beschrieben:

- Make a plan
- Break the problem down

---

<sup>1</sup> Beratung für Softwarequalitätsmanagement und Prozessverbesserung, Philipp-Röth-Weg 14, 64295 Darmstadt, ralf@kneuper.de

- Do the programming of the new subroutines
- Programme the main routine

Einige Jahre später, als die Entwicklungsaufgaben allmählich umfangreicher wurden, stellte Benington 1956 ein erstes Phasenmodell der Entwicklung vor, in dem bereits Begriffe wie „Spezifikation“, „Design“ und „Test“ auftauchen (siehe [Be83]).

Fast zur gleichen Zeit, um 1957, wurde aber auch schon mit einer iterativen Vorgehensweise gearbeitet, wie Gerald M. Weinberg in [LB03] berichtet.

Beide Arbeiten hatten aber zu dieser Zeit noch wenig Einfluss auf die weitere Entwicklung, wie sich schon an den erst wesentlich später erschienen Veröffentlichungen erkennen lässt. Trotzdem wurden diese Ideen in den Folgejahren auch von anderen Autoren allmählich weiter ausgebaut.

Das führte u.a. 1970 zum Vorgehensmodell von Royce, siehe [Ro70], das heute oft als „die“ Referenz für Wasserfallmodelle bezeichnet wird. Royce beschreibt in der Tat einen sequentiellen Ablauf der Entwicklungsphasen, geht dabei allerdings explizit auch auf die Notwendigkeit von Rücksprüngen zu vorherigen Phasen, gelegentlich auch weiter zurück, ein, d.h. es handelt sich nicht um ein streng sequentielles Vorgehen.

Auch das iterative Vorgehen wurde danach immer wieder in der Literatur beschrieben und in der Praxis eingesetzt, siehe [LB03]. Alleine aus Kostengründen konnte dies aber nicht im gleichen Umfang geschehen wie heute, da die Rechenzeit wie auch Speicherplatz teure Ressourcen waren.

Bis etwa Mitte der achtziger Jahre allerdings standen Softwareprozesse und Vorgehensmodelle noch nicht selbst im Fokus der Betrachtung, sondern es mussten erst einmal die darin enthaltenen Aufgaben, Techniken und Methoden entwickelt werden. Insbesondere die aus der Systemanalyse entstandenen strukturierten Methoden entstanden in dieser Zeit, beispielsweise die „Structured Analysis and Design Technique“ (SADT), „Structured Analysis / Structured Design“ (SA/SD) und „Structured Systems Analysis and Design Method“ (SSADM). Grundlegende Bausteine dieser strukturierten Methoden waren die Modellierung der Funktionalität, beispielsweise durch Funktionsbäume und hierarchische Prozessbeschreibungen mit Input, Verarbeitung und Output, oder durch Flussdiagramme; die Modellierung der Daten durch Entity-Relationship-Diagramme; und die Modellierung des Programmablaufs durch Kontroll- und Datenfluss-Diagramme oder Nassi-Shneidermann-Diagramme.

### **3. Mitte der achtziger Jahre: Beginnende systematische Betrachtung der Softwareprozesse**

Ab etwa Mitte der achtziger Jahre wurden, nicht nur in der Softwareentwicklung, die Prozesse zur Erstellung von Produkten zunehmend systematischer betrachtet und als Hebel zur Verbesserung der zu erstellenden Produkte gesehen.

Dies führte beispielsweise 1987 zur Herausgabe der ersten Version der ISO 9000er-Normenreihe (siehe [PG12]) und etwas später zu IT-spezifischen Ansätzen wie CMM [Pa93], V-Modell [Hu90] und ITIL.

Auslöser für die systematische Beschäftigung mit Prozessen in der Softwareentwicklung, neben dem Anstoß aus dem allgemeinen Qualitätsmanagement heraus, waren u. a. die zu dieser Zeit erneut laufenden Diskussionen über die „Softwarekrise“ (siehe z.B. [We92]), sowie ab etwa Anfang der neunziger Jahre die auch in Unternehmen aufkommende objektorientierte Entwicklung (über die Nutzung objektorientierter Programmiersprachen hinaus).

Gemeinsam führten diese verschiedenen Auslöser zu einer intensiven Beschäftigung mit Vorgehensmodellen und Softwareprozessen in den neunziger Jahren. Schon 1984 begann die Tagungsreihe International Software Process Workshop (ISPW, siehe [Po84]), 1991 dann auch die Tagungsreihe der European Workshops on Software Process Technology (EWSPT), wenn auch zuerst noch unter etwas anderem Namen (siehe [FCA91]). In Deutschland erschien 1992 mit [Ch92] eines der ersten Bücher zu diesem Thema, 1993 wurde die Fachgruppe „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“ der Gesellschaft für Informatik gegründet und organisierte 1994 ihren ersten Workshop (siehe [Kn13]). 1995 erschien erstmals die Zeitschrift „Software Process: Improvement and Practice“ bei Wiley.

Schwerpunkte der Arbeit in dieser Zeit waren die Modellierung und Anpassung (Tailoring) von Softwareprozessen sowie deren Unterstützung durch Werkzeuge, beispielsweise durch Process-Centered Software Engineering Environments (PCSEE, siehe z.B. [Gr02]), Werkzeuge für Computer-Aided Software Engineering (CASE) oder Integrated Project Support Environments (IPSE). An einem dieser Projekte, IPSE 2.5 (siehe [Wa90]), und dort speziell dem Teilprojekt mural (siehe [Jo91]) hat auch der Autor dieses Beitrags mitgearbeitet.

Die veröffentlichten Vorgehensmodelle in dieser Zeit kamen meist aus dem Umfeld großer, komplexer Systeme, oft bestehend aus Hard- und Software, häufig auch im militärischen Umfeld.

In der Forschung und Entwicklung von Vorgehensmodellen wurde meist selbstverständlich von eher umfangreichen (später auch als „schwergewichtig“ bezeichneten) Vorgehensmodellen ausgegangen, die die Vorgehensweise bei der Entwicklung relativ detailliert vorgeben und diese Vorgaben ggf. auch durch entsprechende Werkzeuge durchsetzen. Osterweils bekannter Artikel [Os87] beschreibt diese Sichtweise recht gut als den Ansatz, Softwareprozesse zu „programmieren“.

Auch in der Anfang der neunziger Jahre aufkommenden objektorientierten Entwicklung war diese Sichtweise anfangs sehr verbreitet. Zwar änderten sich die verwendeten Entwicklungsmethoden, aus strukturierter Analyse und Design wurden objektorientierte Analyse und Design und schließlich die UML, aber auch hier versuchte man, die Vorgehensweise relativ detailliert festzulegen, beispielsweise im (Rational) Unified Process (siehe [Kr98], [JBR99]).

Bei der Umsetzung dieser Vorgehensweise gab es allerdings, neben den technischen Schwierigkeiten, auch das Problem der Akzeptanz durch die Entwickler. Überspitzt gesagt: Die Entwickler wollten keine „Computer“ sein, die Prozess-Programme ausführen. Die Frage nach den sozialen Aspekten, der Akzeptanz und der passenden Vorgehensweise zur Einführung von Softwareprozessen in Entwicklungsorganisationen war daher ein weiteres wesentliches Thema der Forschung und der Diskussion.

#### **4. Ende der neunziger Jahre: Leichtgewichtige und agile Prozesse**

Die genannten Akzeptanz-Probleme wurden von vielen Vertretern der „klassischen“ Softwareprozesse als eine Frage der Motivation der Entwickler betrachtet. Mit der Zeit setzte sich aber immer mehr die Einsicht durch, dass diese Sichtweise zu einfach war und die Kritik an diesen Softwareprozessen durchaus auch ihre Berechtigung hatte. DeMarco und Lister beispielsweise argumentierten schon 1987 in [DL87] gegen die „Method madness“ und wiesen darauf hin, dass man für Softwareentwickler als Wissensarbeiter andere Management-Methoden benötigt als in der Produktion. Damit ist auch die Frage nach einer anderen Unternehmenskultur verbunden, die weit über die reine Softwareentwicklung hinausgeht, und die gemeinsame Ausrichtung an einem Ziel in den Mittelpunkt stellt, im Gegensatz zur zentral gesteuerten und Plan-getriebenen Arbeitsweise. Als Beispiel für diese in den achtziger Jahren laufende Diskussion sei hier der Bestseller von „Thriving on Chaos“ von Tom Peters [Pe87] genannt.

Wesentliche Anregungen kamen in diesem Zusammenhang auch aus dem gerade entstehenden Thema Wissensmanagement. Beispielsweise entwickelten Nonaka und Takeuchi ihr Modell des Wissensmanagements und führten dabei bereits 1986 in [TN86] Scrum als Arbeitsweise für wissensintensive Arbeiten ein. Ken Schwaber und Jeff Sutherland übernahmen diese Arbeitsweise für die Softwareentwicklung, passten sie an und veröffentlichten sie in [Sc97].

Darüber hinaus wurde immer stärker in Frage gestellt, ob eine Klärung der Anforderungen sowie eine Definition der zu verwendenden Architektur vorab wirklich sinnvoll und realistisch möglich sind, da beide im Laufe der Entwicklung immer wieder mehr oder weniger stark überarbeitet oder korrigiert werden müssen.

Ein weiterer Kritikpunkt war, dass die Möglichkeit zu Rücksprüngen zu früheren Phasen in einem Wasserfall-artigen Vorgehen nicht ausreicht und Iterationen und evolutionäres Vorgehen explizit im Vorgehensmodell berücksichtigt werden sollten. Dies führte beispielsweise zu Ansätzen wie Prototyping [F184], Boehms Spiralmodell [Bo86] und etwas später zum Rapid Application Development [Ma91].

Im Lauf der neunziger Jahre wurden diese Ideen weiter ausgebaut und setzten sich nun auch in der Breite durch. Dies führte neben dem produktiven Einsatz der genannten Methoden zu einer Vielzahl von neuen Methoden wie Extreme Programming (XP, siehe [Be99]), Dynamic Systems Development Method (DSDM), die Crystal-Methodenfamilie, Feature Driven Development (FDD) etc. In unterschiedlichen Ausprägungen und Schwerpunktsetzungen erweiterten diese den Ansatz der iterativen und inkrementellen Entwicklung und legten Wert auf die heute als „agile Methoden“ bezeichneten Prinzipien wie intensive Kommunikation im Projekt, häufige Iterationen, schnelle Rückmeldungen, wenig externe Festlegungen des Projektablaufs etc.

Eine ähnliche, wenn auch nicht ganz so ausgeprägte, Gegenbewegung gegen eine stark Plan- und Prozess-getriebene Arbeitsweise gab es auch in der Industrieproduktion, wo Anfang der neunziger Jahre das Konzept der „Lean Production“ aufkam, angestoßen in erster Linie durch [WJR91]. Viele der Grundideen der Lean Production wurden dann auch in den agilen Methoden in der Softwareentwicklung aufgegriffen.

Unterstützt wurden diese neuen Ansätze durch die ab Mitte der neunziger Jahre stark wachsende Bedeutung des Internets, genauer gesagt des WWWs, und die damit einhergehende wachsende Bedeutung kleiner, zeitkritischer Projekte mit unklaren und schnell wechselnden Anforderungen.

Anfangs wurden diese Vorgehensweisen meist als „leichtgewichtig“ bezeichnet, im Gegensatz zu den „schwergewichtigen“ Modellen, die die Entwicklungsprozesse relativ genau vorschreiben. Im Februar 2001 gab es dann das bekannte Treffen der Vertreter der verschiedenen leichtgewichtigen Prozesse, bei dem das Agile Manifesto entstand (siehe [Hi01], [Be01]). Damit wurde dann auch die Bezeichnung der „agilen“ Methoden eingeführt, der sich sofort durchgesetzt hat, da viele der Vertreter mit der Bezeichnung als „leichtgewichtige“ Methoden eher unglücklich waren.

Durch diese Einigung auf gemeinsame Werte und Prinzipien der agilen Methoden erreichten diese nochmals eine stark wachsende Bedeutung, auch wenn es in der Anfangszeit sehr viele Grundsatzdiskussionen und „Religionskriege“ über die Angemessenheit klassischer, Plan-getriebener Entwicklung einerseits und agiler Entwicklung andererseits gab.

Erst allmählich setzte sich die Einsicht in der Breite durch, dass beide Ansätze ihre Stärken und Schwächen haben, dadurch für unterschiedliche Typen von Projekten geeignet sind, und sich auch nicht notwendig widersprechen, sondern in vielen Fällen eine Kombination von Aspekten beider Ansätze sinnvoll ist. Zu dieser Sichtweise haben insbesondere Boehm und Turner mit ihrem Buch [BT04] beigetragen, in dem sie Kriterien für die Auswahl und Kombination herausarbeiten.

## 5. Softwareprozesse heute

Erfreulicherweise setzt sich die Sichtweise von agiler und Plan-getriebener Entwicklung als Enden eines Spektrums mit vielen Zwischentönen und Kombinationsmöglichkeiten immer mehr durch. Es gibt zwar immer noch gelegentlich Vertreter der einen oder anderen Seite, die nur „ihren“ Ansatz als praktikabel und nützlich gelten lassen, aber diese werden weniger. Immer mehr Projekte versuchen, von beiden Ansätzen das für ihre Aufgabenstellung am besten geeignete zu übernehmen (hybrides Vorgehen).

Das zeigt sich beispielsweise darin, dass auch bei großen, komplexen Projekten mit erheblichem Hardwareanteil, die ohne gründliche Planung kaum Aussicht auf Erfolg hätten, die Projektdauer bzw. der Zeitraum zwischen den Releases kürzer und nur noch in Ausnahmefällen nach Jahren gemessen wird.

Bis vor kurzem gab es allerdings kaum Untersuchungen, die sich kritisch-offen mit den verschiedenen agilen Ansätzen und deren Leistungsfähigkeit, aber auch ihren Einschränkungen, auseinandersetzen. Das hat sich erfreulicherweise mit dem 2014 erschienenen Buch [Me14] von Bertrand Meyer geändert.

Bei den in Unternehmen genutzten Vorgehensmodellen gibt es eine erhebliche Bandbreite, wobei zumindest in Deutschland Scrum sowie V-Modell, meist in organisations- oder projektspezifischer angepasster Form, überwiegen [KL15].

Bei den neuen Entwicklungen seien hier beispielhaft Kanban und DevOps genannt. Kanban ist eine weitere agile Vorgehensweise, die aus der industriellen Produktionssteuerung kommt, mittlerweile aber auch in der Softwareentwicklung an Bedeutung gewonnen hat. Hauptziel von Kanban ist es, die Anzahl der parallel bearbeiteten Aufgaben zu reduzieren und dadurch die Durchlaufzeit bei der Umsetzung von Aufgaben bzw. Anforderungen zu reduzieren.

DevOps adressiert dagegen die verbreiteten Schwierigkeiten bei der Kommunikation zwischen den für die Entwicklung und den Betrieb von Software verantwortlichen Gruppen, insbesondere bei der agilen Entwicklung, wo sehr häufig neue Versionen von Software ausgeliefert werden, die dann möglichst kurzfristig in Betrieb gehen sollen. Ziel von DevOps ist es daher, eine durchgängige und weitgehend automatisierte Lieferkette („Pipeline“) aufzubauen, die von der Klärung der Anforderungen bis zur Inbetriebnahme neuer Funktionalität reicht und dabei die Qualität der gelieferten Funktionalität sicherstellt.

## Literaturverzeichnis

- [Be83] Benington, Herbert D.: Production of Large Computer Programs. *IEEE Annals of the History of Computing* 5 (4): 350–361. 1983. Verfügbar unter <http://sunset.usc.edu/csse/TECHRPTS/1983/usccse83-501/usccse83-501.pdf>
- [Be99] Beck, Kent: *Extreme Programming Explained – Embrace Change*. Addison-Wesley, 1999.
- [Be01] Beck, Kent et al.: *Manifesto for Agile Software Development*. <http://agilemanifesto.org/>. 2001.
- [Bo86] Boehm Barry W.: A Spiral Model of Software Development and Enhancement, *ACM SIGSOFT Software Engineering Notes*, ACM, 11(4). S. 14-24, August 1986.
- [BT04] Boehm, Barry W.; Turner, Robert: *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA: Addison-Wesley, 2004.
- [Ch92] Chroust, Gerhard: *Modelle der Software-Entwicklung*. Oldenbourg Verlag, 1992.
- [DL87] DeMarco, Tom; Lister, Timothy: *Peopleware. Productive Projects and Teams*. Dorset House Publishing Co., 1987.
- [Fl84] Floyd, Christiane: A systematic look at prototyping. In (Budde et al., Hrsg.): *Approaches to Prototyping*; Proc. Namur. S. 1-18. Springer-Verlag, 1984.
- [FCA91] Fuggetta, A.; Conradi, R.; Ambriola, V. (Hrsg.): *First European Workshop on Software Process Modeling : CEFRIEL*, Milan (Italy), 30-31 May 1991. Associazione Italiana per l'Informatica ed il Calcolo Automatico, Working Group on Software Engineering, 1991. (cf. <http://trove.nla.gov.au/work/22457426>)
- [Gr02] Gruhn, Volker: Process-Centered Software Engineering Environments. A Brief History and Future Challenges. *Annals of Software Engineering* 14. S. 363–382, 2002.
- [Hi01] Highsmith, Jim: *History: The Agile Manifesto*. <http://agilemanifesto.org/history.html>. 2001.
- [Hu90] Hummel, Helmut: The Life Cycle Methodology for Software Production and the Related Experience. In (Ehrenberger, W., Hrsg.): *Approving Software Products*. Proc. of the IFIP WG 5.4 Working Conference. North-Holland, 1990.
- [JBR99] Jacobson, Ivar; Booch, Grady; Rumbaugh, James: *The Unified Software Development Process*. Addison-Wesley, 1999.

- [Jo91] Jones, Cliff .B.; Jones, Kevin D.; Lindsay, Peter A.; Moore, Richard: mural. A Formal Development Support System. Springer-Verlag, 1991.
- [KL15] Kührmann, Marco; Linszen, Oliver: Vorgehensmodelle in Deutschland. Nutzung von 2006-2013 im Überblick. WI-MAW-Rundbrief, April 2015.
- [Kn13] Kneuper, Ralf: Zwanzig Jahre GI-Fachgruppe "Vorgehensmodelle für die betriebliche Anwendungsentwicklung". In Vorgehensmodelle 2013, Lecture Notes in Informatics, Vol. P-224, Gesellschaft für Informatik, Bonn. S. 17-30, 2013.
- [Kr98] Kruchten, Philippe: The Rational Unified Process: An Introduction. Addison-Wesley, 1998.
- [LB03] Larmann, Craig; Basili, Victor R.: Iterative and Incremental Development: A Brief History. IEEE Computer, Juni 2003.
- [Ma91] Martin, James: Rapid Application Development. Macmillan, 1991.
- [Me14] Meyer, Bertrand: Agile! The Good, the Hype and the Ugly. Springer, 2014.
- [Os87] Osterweil, Leon: Software Processes are Software Too. In Proceedings of the Ninth International Conference on Software Engineering, 1987.
- [Pa93] Paulk, Mark et al.: Capability Maturity ModelSM for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, 1993.
- [Pe87] Peters, Tom: Thriving on Chaos. Handbook for a Management Revolution. Alfred A. Knopf, Inc., 1987.
- [PG12] Petrick, Klaus; Graichen, Frank: 25 Jahre ISO 9001. Erfolgsweg einer Systemnorm. In QZ Qualität und Zuverlässigkeit, Heft 3, 2012, S. 3-5. Verfügbar unter <http://www.qz-online.de/qz-zeitschrift/archiv/artikel/erfolgsweg-einer-systemnorm-271979.html>.
- [Po84] Potts, Colin (Hrsg.): Proceedings of a Software Process Workshop, February 1984, Egham, UK. IEEE Computer Society, 1984.
- [Ro70] Royce, Winston W.: Managing the Development of Large Software Systems. In Proceedings, IEEE Wescon August 1970, S. 1-9. 1970.
- [Sc97] Schwaber, Ken: SCRUM Development Process. In OOPSLA '95 Workshop Proceedings 16 October 1995, Austin, Texas. Springer, 1997. Verfügbar unter <http://jeffsutherland.org/oopsla/schwaber.html>.
- [TN86] Takeuchi, Hirotaka; Nonaka, Ikujiro: The New New Product Development Game. Harvard Business Review, Januar 1986.
- [Tu51] Turing, Alan M.: Programmer's Handbook for Manchester Electronic Computer Mark II. Faksimile verfügbar unter [http://www.alanturing.net/turing\\_archive/archive/m/m01/M01-001.html](http://www.alanturing.net/turing_archive/archive/m/m01/M01-001.html). Transkript verfügbar unter <http://curation.cs.manchester.ac.uk/computer50/www.computer50.org/kgill/mark1/RobertTau/turing.html>. Undatiert, erstellt um 1951.
- [Wa90] Warboys, Brian C.: The IPSE 2.5 Project: Process Modelling as a Basis for a Support Environment. In (Madhavji, N.; Schäfer, W.; Weber, H., Hrsg.): Proceedings of the First International Conference on System Development Support Environments and

Factories, S. 59-74, Pitman, 1990. Beitrag verfügbar unter  
<http://apt.cs.manchester.ac.uk/ftp/pub/IPG/bw89.pdf>.

- [We92] Weber, Herbert: Die Software-Krise und ihre Macher. Springer-Verlag, 1992.
- [WJR91] Womack, James P.; Jones, Daniel T.; Roos, Daniel: The Machine That Changed the World : The Story of Lean Production, Harper Perennial, 1991.



# Erfolgsfaktoren bei der Umsetzung hybrider Projekte – Ergebnisse einer Befragung und praktische Empfehlungen zur Umsetzung

Anna Aldushyna<sup>1</sup>, Martin Engstler<sup>2</sup>

**Abstract:** In hybriden Projekten werden klassische und agile Vorgehensmodelle im Projektablauf kombiniert, um die Vorteile beider Ansätze nutzensteigernd zu verbinden. Der vorliegende Artikel beschreibt die Erfolgsfaktoren für die Umsetzung solcher hybrider Projekte. Grundlage der Studie bildet eine empirische Studie bei Projektmanagern sowohl hybrider als auch klassischer Projekte bei einem international agierenden Finanzdienstleister in der Automobilindustrie. Die Ergebnisse der Analyse zeigen, dass hybride Vorgehensmodelle im operativen Einsatz bereits heute erfolgreich umgesetzt werden können. In der Studie werden die hierfür zu schaffenden Rahmenbedingungen ermittelt und praktische Empfehlungen zur Etablierung hybrider Projekt im Unternehmen abgeleitet.

**Keywords:** Hybride Projekte, klassische, agile u. hybride Vorgehensmodelle, Erfolgsfaktoren.

## 1 Einleitung

Vorgehensmodelle reduzieren durch Regeln und vordefinierte Schritte die Komplexität bei der Projektplanung. Sie sorgen für eine effiziente und sichere Durchführung des Projekts und leisten darüber hinaus einen Beitrag zur Qualitätssicherung [RF08]. Ein richtig ausgewähltes Vorgehensmodell wird als erfolgskritischer Faktor für ein Projekt gesehen [Sc12]. Für die projektspezifische Auswahl eines Vorgehensmodells und der Formulierung der Regeln der Projektdurchführung müssen zunächst die situativen Rahmenbedingungen geklärt werden. Hinzu kommt, dass die Auswahl eines Vorgehensmodells auch modischen Trends unterworfen ist [Cr12].

In der Praxis kommt heute eine Vielzahl an Vorgehensmodellen zum Einsatz, auch setzen viele Unternehmen mehrere Modelle parallel ein bzw. nehmen individuelle Anpassungen oder Ergänzungen vor [KL14]. Vereinfachend wird in diesem Beitrag eine Untergliederung der Ansätze in klassische und agile Vorgehensmodelle vorgenommen, die in einem hybriden Ansatz kombiniert werden. Klassische Vorgehensmodelle werden als systematisch, phasenorientiert und gut berechenbar beschrieben, sie eignen sich für klar definierte Problemstellungen. Agile Vorgehensmodelle hingegen werden als hoch flexibel charakterisiert und liefern auch für noch nicht eindeutig definierte Problemstellungen eine transparente Vorgehensweise [u.a. SBK14, Ha13]. Der steigende Bedarf an Flexibi-

---

<sup>1</sup> Daimler AG, Strategisches Controlling & Systeme FM, Epplestrasse 225, 70546, Stuttgart, anna.aldushyna@daimler.com

<sup>2</sup> Hochschule der Medien, Fakultät IuK, Nobelstraße 10, 70569 Stuttgart, engstler@hdm-stuttgart.de

lität (z.B. hinsichtlich der Umsetzung von *unscharfen* Anforderungen) und gleichzeitig an Stabilität hat Mischformen verschiedener Vorgehensmodelle hervorgebracht [Ba12]. Die Hybridität erlaubt eine Kombination aus klassischen und agilen Mustern sowie die Möglichkeit, mehrere Vorgehensweisen innerhalb einer Projektstruktur situativ zu variieren, um den projektspezifischen Anforderungen bestmöglich gerecht zu werden. Gleichzeitig sind die hybride Vorgehensweise mit erhöhten Konfliktpotenzialen und Risiken verbunden. Die Umsetzung eines hybriden Projekts erfordert zudem eine Anpassung oder eine Erweiterung der zu kombinierenden Vorgehensmodelle [KK12]. Für die erfolgreiche Umsetzung eines hybriden Projekts sind daher zunächst die relevanten Einflussfaktoren auf den Erfolg des Projekts zu identifizieren [Da01]. Zu berücksichtigen ist, dass nicht alle Erfolgsfaktoren für individuelle Projekte von gleicher Bedeutung [An o.J.] und spezifische Konfigurationen in den konkreten Projekten umsetzbar sind. Im Mittelpunkt der vorliegenden Untersuchung steht die Frage, welche Erfolgsfaktoren bei der Umsetzung hybrider Projekte in der Praxis identifiziert und welche Empfehlungen für die Umsetzung hybrider Projekte hieraus abgeleitet werden können.

## 2 Vorgehensmodelle für hybride Projekte

In *hybriden Projekten* werden bekannte und erprobte Vorgehensweisen aus der strukturierten (klassischen) und der flexiblen (agilen) Welt miteinander kombiniert [SBK14]. Sie verbinden hierbei ihre Vorteile und verbessern damit das Projektmanagement [Re13]. Hybride Vorgehensmodelle zeichnen sich somit durch die Kopplung zweier (oder mehrerer) Prozeduren aus, zwischen denen eine hohe Gegensätzlichkeit besteht [Re12]. Die Vorgehensmodelle können je nach Rahmenbedingungen, Größe oder Komplexität des Projekts auch mit weiteren Werkzeugen ergänzt werden. Hybridkonzepte sind dabei nicht auf die Kombination klassischer und agiler Konzepte in Mixed-Methods-Ansätzen begrenzt. In der Softwareentwicklung existieren bereits mehrere hybride Kombinationen aus verschiedenen Vorgehensmodellen, insbesondere Weiterentwicklungen agiler Methoden z.B. *Leagile* als Neologismus für *lean & agile* oder *Scrumban* für *Scrum & Kanban* [Re10]. Die Umsetzung eines hybriden Vorgehensmodells erfordert zuallererst die Beschreibung der Kombinationsmuster, d.h. es ist zu entscheiden, wie und welche Vorgehensmodelle oder deren Werkzeuge sinnvoll zu kombinieren sind. Hierzu existieren keine Richtlinien oder ein *Hybrid-Manifest* für die Kombination und Umsetzung hybrider Vorgehensmodelle. Das ist einerseits ein Vorteil, da kein vollständiger Bruch mit klassischem Projektmanagement und Standardisierungen erforderlich ist. Andererseits wird es ohne genau beschriebene Richtlinien zu Unklarheiten bei der Gestaltung und Umsetzung der hybriden Vorgehensweise führen [Re10].

Bei der Kombination klassischer und agiler Vorgehensmodelle in hybriden Ansätzen können auch unterschiedliche Muster mit verschiedenen Kopplungsgraden und Ausprägungen gebildet werden. In einer Kombination können auch Teilprojekte innerhalb eines Projekts nach unterschiedlichen Vorgehensmodellen z.B. als eine *Kombination vom Wasserfallmodell (klassisch) und Scrum (agil)* agieren. Derartige Strukturierungsmuster

ermöglichen es, die Vorgehensmodelle nach Bedürfnissen und Spezifikation einzelner Teilprojekte effizient einzusetzen. Dieses hybride Vorgehensmodell wird für den Fall empfohlen, bei dem sich der Projektgegenstand in agil und klassisch abzuwickelnde Anteile trennen lässt und deren integrative Kombination einen Zusatznutzen verspricht [KK12]. Ein anderes Kombinationsmuster wäre z.B. *agile Anteile im klassischen Projekt*. Wenn schon zum Projektbeginn der Projektgegenstand und die Anforderungen ausführlich definiert werden können, wird empfohlen, ein klassisches Vorgehensmodell für die Projektplanung anzuwenden. Es kann jedoch in weiteren Phasen des Projekts zu nicht vorher definierten Anforderungen oder Änderungen kommen. Um auf solche Hindernisse flexibel reagieren zu können, werden die Grundregeln des klassischen Wasserfallmodells mit Scrum-Werkzeugen kombiniert und damit möglichst viel Nutzen aus beiden Vorgehensmodellen erzielt. Dabei wird in einem klassischen Projekt u.a. in der Entwicklungsphase eine agile Methodik verwendet [Hü14].

In Projekten werden die dargestellten Konfigurationen von Hybridansätzen eingesetzt und dabei entsprechende prozessuale und strukturelle Integrationsätze umgesetzt. Im Unternehmen sind beide oben angeführten Konfigurationen im Einsatz. Vorliegende Studien zum Einsatz von Vorgehensmodellen fokussieren auf die Verbreitung von Ansätzen [u.a. KL14] oder dokumentieren in Fallstudien deren Umsetzung [u.a. SBK14], die Erfolgsfaktoren der Umsetzung aus Sicht der Projektbeteiligten hingegen wurden bislang nicht adressiert. Ziel der Erhebung ist es, die Einsatz- und Erfolgsfaktoren hybrider Vorgehensmodelle in Unternehmen zu untersuchen und hieraus Empfehlungen für die Umsetzung hybrider Vorgehensmodelle abzuleiten.

### **3 Ermittlung der Erfolgsfaktoren hybrider Projekte**

Die Ermittlung der Erfolgsfaktoren hybrider Vorgehensmodelle erfolgte in einem mehrstufigen Vorgehen unter Einsatz verschiedener Befragungen und wurde im Zeitraum April bis Juni 2014 in einem international agierenden Finanzdienstleister in der Automobilindustrie (im Folgenden *Unternehmen*) durchgeführt. Der IT-Bezug gilt als wesentlicher Bestandteil für alle ausgewählten und analysierten Projekte des Unternehmens. Im Jahr 2014 beinhaltet das untersuchte Projektportfolio weltweit über 90 Projekte, davon etwa 10% mit agilem oder stark iterativem Ansatz. Im Vergleich zu den Vorjahren konnte zudem eine zunehmende Zahl an verschiedene Kombinationen aus agilen und klassischen Vorgehensmodellen beobachtet werden. Das Ziel der Befragung bestand in der Ermittlung der Erfolgskriterien für ein hybrides Projekt und Ausarbeitung einer Hilfestellung in Form von Empfehlungen für deren Umsetzung. Die zweistufige Befragung, bestehend aus einer schriftlichen Befragung und vertiefenden Interviews, sollte den derzeitigen Stand der Etablierung hybrider Vorgehensweise im Unternehmen ermitteln. Neben einem hybriden wurde auch ein klassisches Vorgehensmodell analysiert, um vergleichen zu können, wie sich diese beiden Vorgehensmodelle im Unternehmen zurzeit positionieren. Für diese Untersuchung war es nicht möglich sämtliche Einheiten aus der Grundgesamtheit der Projekte in die Befragung einzubeziehen. Deswegen

wurde die Befragung anhand einer gezielten Stichprobe (bewusste Auswahl) durchgeführt. Die Auswahl der Stichprobe der zu befragenden Projekte hat auf Basis der Unternehmensprojektdatenbank stattgefunden und stellt die Berücksichtigung falltypischer Merkmale sicher [SHE11]. Das gewählte Untersuchungsdesign und darin integrierte Methoden sind in Abb. 1 dargestellt.

**Untersuchungsdesign**

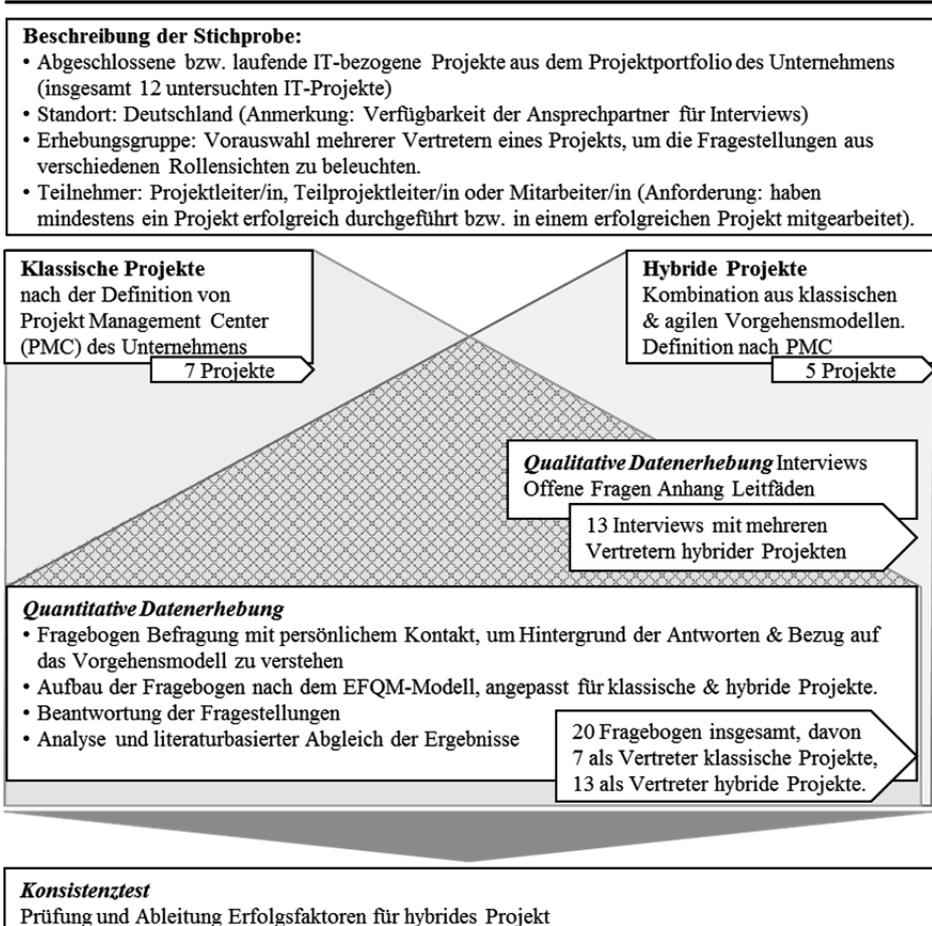


Abb. 1: Untersuchungsdesign

Auf Grund der kleinen Anzahl analysierter Projekte (12 gezielt ausgewählte Projekte) und ausgewählten Befragten (20 Teilnehmer an der schriftlichen Befragung, davon 13 Projektverantwortliche hybrider Projekte, mit denen zudem vertiefende Interviews durchgeführt wurden) kann die vorliegende Auswertung lediglich eine empirische Ten-

denz aufzeigen, die jedoch eine wertvolle Bereicherung zu den fallstudienorientierten Analysen in der Literatur [u.a. SBK14] darstellt. Rein agil durchgeführte Projekte wurden in der Analyse nicht untersucht, da diese Variante im untersuchten Projektfeld in zu geringer Zahl vorkam.

Als Basismodell für die Spezifizierung der Erfolgsfaktoren eines hybriden Projekts wurde das Project-Excellence-Modell verwendet [GP14], siehe Abb. 2. Die für diese Analyse definierten Untersuchungskriterien sind auf den Projektmanagementteil des Project-Excellence-Modells reduziert. Es wurden einerseits Anpassungen entsprechend spezifischer Anforderungen bzw. Begriffe im IT-Projektmanagement vorgenommen (z. B. Anforderungsmanagement [Hi09]) sowie aus der Literatur ableitbare Unterschiede in hybriden und klassischen Projekten ergänzt [Hü14, KK12]. Auf der Basis des Project-Excellence-Modells wurde ein Fragenkatalog für die Messung des Erfolgs eines hybriden Projekts erarbeitet.

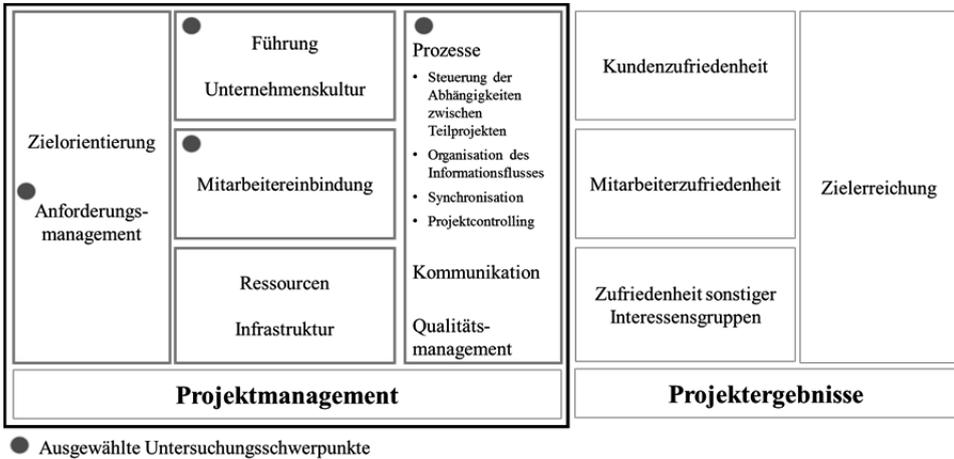


Abb. 2: Project Excellence auf Basis des EFQM-Modells, angepasst für hybride Projekte (Darstellung in Anlehnung an [GP14])

In der quantitativen Ergebnisdarstellung zum Projektmanagement sind die Fragestellungen in fünf Blöcke aufgeteilt (siehe Abb. 2), von denen vier ausgewählte Untersuchungsbereiche nachfolgend dargestellt werden (*Anforderungsmanagement, Führung, Unternehmenskultur, Mitarbeiterereinbindung, Prozesse*, der Bereich *Ressourcen / Infrastruktur* zeigte keine vorgehensmodellbezogenen Unterschiede und wurde daher in der Darstellung nicht vertieft). Die Bewertungsskala basiert auf einer Likert-Skala, für die folgende Aussagen formuliert wurden: 0 = *trifft für den Fall nicht zu*; 1 = *unter den Erwartungen*; 2 = *entspricht knapp den Erwartungen*; 3 = *den Erwartungen entsprechend*; 4 = *erkennbar über den Erwartungen*; 5 = *ausgezeichnet*.

### 3.1 Anforderungsmanagement

Die Vollständigkeit und Richtigkeit der Anforderungen ist ein wichtiger Erfolgsfaktor für Projekt (vgl. u.a. Chaos-Reports der Standish Group [Hi09]). Die Ergebnisse zeigen, dass die Projektbeteiligten die *Vollständigkeit und Konsistenz der Anforderungen* in hybriden Projekten deutlich besser bewerten als in klassischen Projekten (siehe Abb. 3).

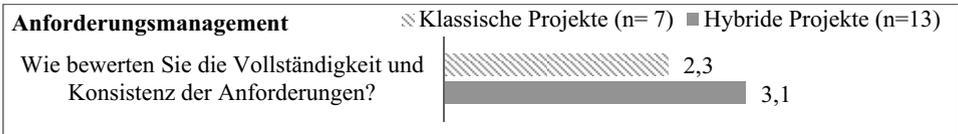


Abb. 3: Anforderungsmanagement

Die Mehrheit der Vertreter hybrider Projekte hat die Verwendung eines *Product Backlog* als erfolgsrelevant genannt. Die Umsetzung des *Requirements Engineering* in hybriden Projekten wurde ebenfalls gut bewertet, obwohl hier die Befragten größere Verbesserungspotenziale hinsichtlich der richtigen Umsetzung dieses Verfahrens, angefangen von der Methodik bis hin zur entsprechend ausgewählten Tools, sehen. Projektbeteiligte in klassischen Projekten hatten hingegen Schwierigkeiten beim Umgang mit den sich ständig ändernden Anforderungen und der beschränkten Zeit für deren Beschreibung.

### 3.2 Führung

Der Erfolg eines Projekts ist eng mit der Kompetenz und dem Handeln der Projektleitung verbunden. Die Definition von Rollen, Aufgaben, Kompetenzen und Verantwortung wurde in klassischen Projekten effektiver bewertet (siehe Abb. 4), weil sie auf größere Erfahrungen mit dem klassischen und damit weit verbreiteten Vorgehensmodell basieren.

In hybriden Projekten gab es in manchen Fällen dagegen Verständnisschwierigkeiten über die Rollendefinition und deren Akzeptanz. Die Befragten legen hier den Wert auf qualifizierte Aufklärung zu den agilen Rollen für die Mitarbeiter des Projekts. Obwohl aktuell das Thema *Agilität* immer mehr im Unternehmen akzeptiert und verstanden wird, soll sich in Zukunft hierfür entsprechende Zeit eingeplant werden. Die Akzeptanz der klassischen Rollen liegt bei klassischen und hybriden Projekten entsprechend deutlich höher als bei den agilen Rollen. Die Befragten verbinden dies allerdings nicht immer mit der Bekanntheit des klassischen Vorgehensmodells, sondern mit anderen Faktoren wie z.B. fachlich klar abgegrenzte Rollen oder tradierte Rollenmuster.

Die Anwendung eines kontinuierlichen Verbesserungsprozesses als Bestandteil des agilen Vorgehens wurde in hybriden Projekten mit dem Mittelwert 3 bewertet.

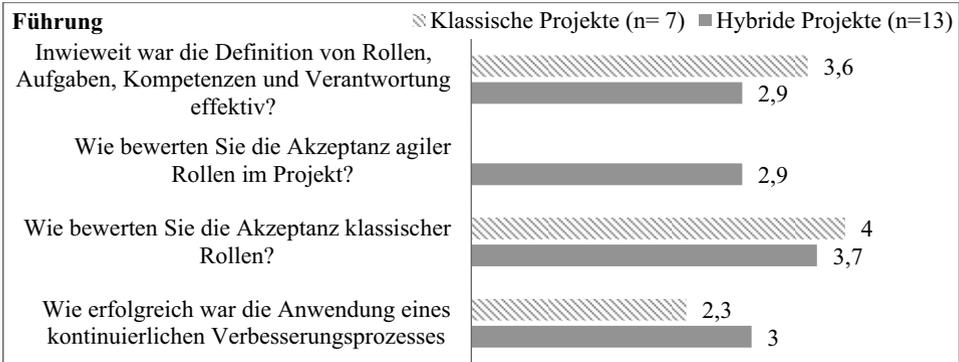


Abb. 4: Führung

Hier wurde mehrfach betont, dass nach dem durchgeführten Lessons Learned die dabei entstandenen Erkenntnisse auch umgesetzt werden sollen. Anders stellt sich die Situation in klassischen Projekten dar. Die Abschätzung der Anwendung eines kontinuierlichen Verbesserungsprozesses liegt bei 2,3 Punkten. Es ist damit begründet, dass bei den 50% der Befragten ein Lessons Learned-Prozess nur am Ende des Projekts durchgeführt wurde.

### 3.3 Unternehmenskultur

Die Umsetzung von Projektmodellen kann nicht unabhängig von der gelebten Unternehmenskultur, d.h. den Einstellungen und Werten von Unternehmensführung und Mitarbeitern bewertet werden. Die Befragung zeigt, dass die bestehende Unternehmenskultur im untersuchten Unternehmen bislang deutlich mehr die *klassischen* Projekte fördert (siehe Abb. 5).

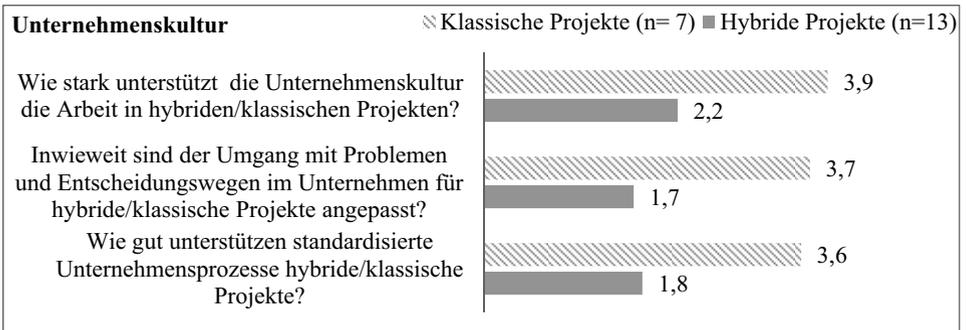


Abb. 5: Unternehmenskultur

Die Projektgremien orientieren sich heute noch überwiegend an dem klassischen Vorgehen, die Eskalationswege (z. B. hierarchische Linien) sind entsprechend gut organisiert.

Vor paar Jahren war alles im Unternehmen auf klassische Vorgehensmodelle ausgelegt. Langsam ändert sich diese Konstellation in die Richtung Agilität. Hybride Projekte werden deutlich weniger unterstützt. Als Gründe hierfür wurden mangelnde Erfahrungen mit solchen Vorgehensmodellen genannt.

Die Entwicklung der Projektmanagementkultur geht dabei in die Richtung der Agilität. Die Grundsteine für diese Vorgehensweise wurden gelegt. Des Weiteren soll das unternehmensweite Vorgehensmodell in die Richtung agil und hybrid weiterentwickelt werden. Es sollen z.B. die Quality Gates für agile Vorgehensmodelle stärker adaptiert werden.

### 3.4 Mitarbeitereinbindung

Die Mitarbeitermotivation in der Projektlaufzeit hat sich in hybriden Projekten stärker als in klassischen entwickelt (siehe Abb. 6). Der agile Anteil im hybriden Projekt erhöht die Motivation der Mitarbeiter, die Projektergebnisse werden früher sichtbar und der Austausch mit den Stakeholdern findet frühzeitiger statt. Dadurch ergibt sich die Möglichkeit, Fehler rechtzeitig zu erkennen bzw. zu beheben (Qualitätsverbesserung). Des Weiteren übernehmen die Mitarbeiter im agilen Team eine gewisse Verantwortung, welche die Motivation der Mitarbeiter im Projekt erhöht.

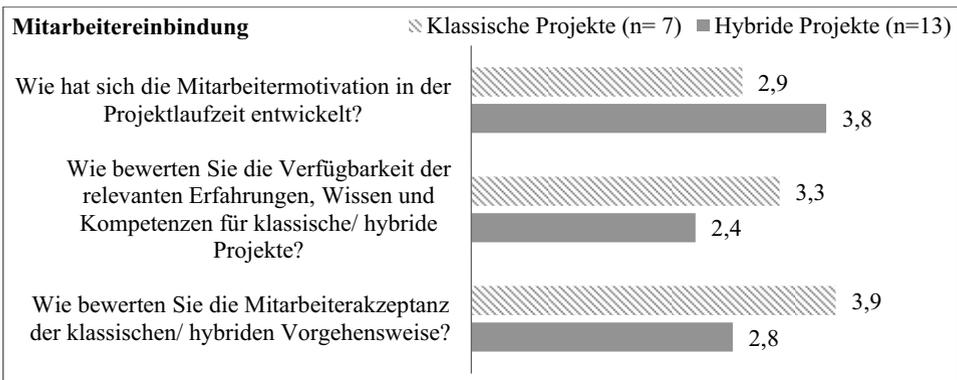


Abb. 6: Mitarbeitereinbindung

Die Verfügbarkeit der relevanten Erfahrungen, Wissen und Kompetenzen wurden in klassischen Projekten höher als in hybriden eingeschätzt. Die Mehrheit der Befragten hat das mit der Bekanntheit des klassischen Vorgehensmodells als unternehmensweiter Standard begründet. Die Mitarbeiter in hybriden Projekten hingegen begründen ihre Antworten mit mangelnder Verfügbarkeit der Erfahrungen in hybriden Projekten. Hier gilt es das Prinzip *learning by doing* zu fördern. Die Mitarbeiterakzeptanz der Vorgehensweise wurde ebenso in klassischen Projekten höher bewertet. Der Grund hierfür liegt analog, im weit verbreiteten klassischen Vorgehensmodell. Für die hybriden Projekte ergab sich hier die Herausforderung, dass die Projektmitarbeiter auf derzeitigen

Stand eine relative Unflexibilität aufweisen und mit anderen Arbeitsformen erst noch zurechtkommen müssen.

### 3.5 Prozesse

Für die Projektsteuerung sind klare Regeln erforderlich, die Transparenz und Sicherheit bezüglich der projektrelevanten Entscheidungen bieten. Bezüglich der Quality Gates in der IT und im Fachbereich zeigt sich die nachfolgend eindeutig klare Tendenz (siehe Abb. 7).

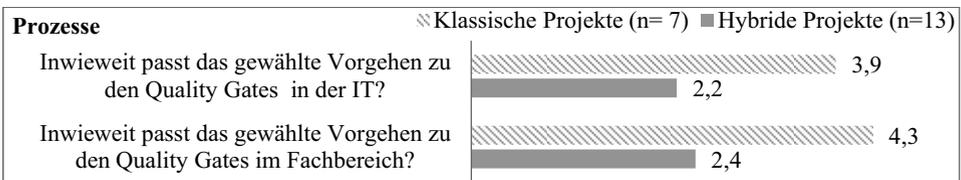


Abb. 7: Prozesse (Qualität)

Die Quality Gates wurden ursprünglich für die klassischen Projekte bestimmt und entsprechend hoch bewertet. In Bezug auf agile Vorgehensmodelle wünschen sich hybride Projekte hingegen eine entsprechende Anpassung.

### 3.6 Zusammenfassende Analyse und Interpretation der Ergebnissen

Die Ergebnisse der durchgeführten Befragung im Unternehmen zeigen, dass die klassische Vorgehensweise als standardisiertes Vorgehensmodell im Unternehmen aktuell noch weit verbreitet ist und entsprechend hoch von den Mitarbeitern sowie dem Management akzeptiert werden. Die Unternehmenskultur mit ihren standardisierten Prozessen, Gremien und Infrastruktur ist ebenfalls noch stark auf das klassische Vorgehen ausgerichtet. Allerdings ist eine Tendenz zur Implementierung agiler Elemente oder anderer Werkzeuge in klassischen Projekten zu erkennen. Immer öfter werden flexible Requirements Engineering-Methoden anstatt des Fachkonzepts gewünscht oder das Wasserfallmodell mit Iterationen verwendet.

Insgesamt wurde durch die Befragung und die vertiefenden Interviews bestätigt, dass für die hybriden Projekte die allgemeingültigen Erfolgsfaktoren des Projekts in einigen Kriterien angepasst oder erweitert werden müssen [Hü14, KK12]. In diesem Zusammenhang lässt sich insbesondere ein Anpassungsbedarf hinsichtlich einer veränderten Unternehmens- und Projektmanagementkultur nennen. Es wurde mehr Flexibilität hinsichtlich standardisierter Unternehmensprozesse, Infrastruktur und Richtlinien gewünscht. Eine Herausforderung ist die effiziente Projektstrukturplanung mit der damit verbundenen Synchronisation von Teilprojekten mit unterschiedlichen Vorgehensmodellen. Das hybride Vorgehen bringt sowohl für die Mitarbeiter und Auftraggeber als auch für die exter-

nen Dienstleister noch viel Unsicherheit mit sich. Hier wurde eine bessere Unterstützung des Managements bezüglich der Kommunikation gefordert. Weiterer Anpassungsbedarf besteht in der Umsetzung entsprechender Rollenkonzepte sowie dem Ausbau des Wissens und der Kompetenzen der Mitarbeiter bezüglich agiler und hybrider Vorgehensmodelle. Insgesamt bestehen bereits Veränderungen im Unternehmen in Richtung agile bzw. hybride Vorgehensmodelle. Jedoch fehlt hierzu ein entsprechendes Framework bzw. grundlegende Regeln sind noch nicht hinreichend verankert.

## 4 Empfehlungen zur Umsetzung hybrider Projekte

Aus den Ergebnissen der Untersuchung zu Erfolgsfaktoren hybrider Projekte lassen sich eine Reihe von Empfehlungen zur Gestaltung und operativen Umsetzung hybrider Projekte in Unternehmen ableiten, die nachfolgend unter Einbeziehung ergänzender Literaturquellen dargestellt werden.

### 4.1 Sicherstellung der Erfolgsfaktoren hybrider Projekte

Auf Basis der Erfolgsfaktoren beim Management hybrider Projekte lassen sich folgende ausgewählte Empfehlungen formulieren:

#### **Zielorientierung**

##### *Sinnvolle Kombinationsmuster für hybrides Projekt auswählen*

Es ist zu entscheiden, welche Vorgehensmodelle mit welchen Kombinationsmustern für ein konkretes Projekt passend sind. Dabei sind vor allem die Komplexität der Anforderungen und das Einsatzgebiet des Vorgehensmodells zu beachten. Außerdem ist eine genaue Analyse des Umfelds erforderlich. Nach der Auswahl der geeigneten Kombination, ist diese den Stakeholdern wie Management und Projektmitarbeitern vorzustellen und bei Bedarf abzustimmen. Ziel ist es, zu prüfen, ob das ausgewählte hybride Vorgehensmodell effizient und machbar ist [Ko12].

Außerdem ist zu definieren, ob ein hybrides Projekt nach *Prozessen oder Komponenten* geschnitten werden soll. Bei der Auswahl ist auf die Abhängigkeiten zwischen Teilprojekten und die teilprojektübergreifenden Prozesse zu achten. Eine Unterteilung des Projekts nach Komponenten macht es schwieriger, den Gesamtprozess, welcher durch mehrere Komponente läuft, abzuwickeln (abgeleitet aus Interviews).

#### **Führung**

##### *Rollenkonzept für hybrides Projekt anpassen*

Um die möglichen Rollenkonflikte zu vermeiden, soll das Rollenkonzept angepasst werden. Eine Adaption des Rollenverständnisses oder Rollenaustausch ist in einem klassischen Projekt mit agilen Anteilen sinnvoll. So soll der Product Owner nicht zugleich der Scrum Master sein. Dabei ist der Scrum Master weder der Vorgesetzte von Teammitgliedern, noch der Projektleiter. Nur mit einem Vertrauensverhältnis kann der Erfolg erreicht werden [Hü14].

## **Unternehmenskultur**

### *Unternehmens- und Projektmanagementkultur flexibler gestalten*

Die Breite des unternehmensweiten (in diesem Fall klassischen) Vorgehensmodells soll in Zukunft erweitert werden. Hierfür finden sinnvolle Ergänzungen durch den Einsatz agiler Werkzeuge statt.

- Quality Gates in der IT und im Fachbereich für hybride Projekte anpassen (hinsichtlich agiler Vorgehensmodelle);
- Hybrides Framework als integriertes Gesamtkonzept mit agilen Projektmanagement-Regeln in Kombination mit Mustern aus dem klassischen Vorgehensmodell ausarbeiten und stetig weiterentwickeln.

## **Mitarbeiterbindung**

### *Mitarbeiterakzeptanz des hybriden Vorgehensmodells steigern*

Da das hybride Vorgehen in sich noch viel Unsicherheit für die Mitarbeiter und Auftraggeber trägt, ist eine entsprechende Kommunikation und *Unterstützung der obersten Führungsebene* notwendig (abgeleitet aus dem Interview). Das Ziel liegt darin, Unsicherheiten zu senken und gleichzeitig Mitarbeiter zu motivieren [En09].

Insgesamt sind folgende Punkte zu beachten:

- Verständnis der Mitarbeiter aufbauen und keine harten Abgrenzungen zulassen, um für verschiedene Wertestrukturen bei agilen und klassischen Vorgehensmodellen zu sensibilisieren;
- Zusammenarbeit des Teams verbessern, wo eine gewisse Disziplin hinsichtlich hybrides Vorgehens notwendig ist;
- Erreichung einer Akzeptanz der Projektmitarbeiter über die ausgewählten Werkzeuge. Dabei sollen die Werkzeuge die individuell definierten Prozesse unterstützen.

## **Prozesse**

### *Gesamtprojektsteuerung für hybrides Projekt sorgfältig organisieren*

Bei der Gesamtprojektplanung eines hybriden Projekts sind folgende Aspekte zu definieren:

- Umgang mit der Komplexität bestimmen. Die Abhängigkeiten zwischen TP entsprechend managen und nach Möglichkeit eliminieren (aus dem Interview);
- Einheitliche Dokumentation und Reporting für hybride und klassische Projekte definieren (abgeleitet aus dem Interview).

*Fortschrittskontrolle für klassische und agile Anteile transparent gestalten*

Eine vollständige transparente Übersicht über die Fortschrittskontrolle soll sowohl in klassischen als auch in agilen TP gegeben sein. Somit kann z.B. ein Dashboard mit allen Anforderungen, deren Abhängigkeiten und Fertigstellungsgrad geführt werden. Auf dem Dashboard werden die Auswirkungen auf das Gesamtprojekt veranschaulicht, wenn sich die einzelnen Anforderungsumsetzungen verschieben. Somit können diese Auswirkungen entsprechend berücksichtigt werden [Ko12].

*Kontinuierlichen Verbesserungsprozess in klassischem Anteil des Projekts etablieren*

Der agile Teil des hybriden Projekts verfügt bereits über die regelmäßigen Retrospektiven. Eine Herausforderung liegt insbesondere darin, auch den klassischen Teil des Projekts in einen kontinuierlichen Verbesserungsprozess einzubeziehen (abgeleitet aus Interviews).

*Kommunikationskonzept an das hybride Vorgehen anpassen*

Die Synchronisation zwischen Teilprojekt und Gesamtprojekt muss gründlich geplant und kontrolliert werden. Alle geplanten Iterations- und Synchronisationstermine sind in einen generellen Synchronisationsplan einzutragen und im Projektverlauf einzuhalten. Als Synchronisationspunkte können Meilensteine oder Iterationen verwendet werden. Hier ist nicht nur die zeitliche, sondern auch inhaltliche Synchronisation zu berücksichtigen. Eine realistische und transparente Synchronisationsplanung sowie deren Aktualisierung sind hierbei wichtige Aspekte. Der Projektleiter hat die Aufgabe, den Mitarbeitern zu erklären und zu kommunizieren, warum die geplanten Synchronisationspunkte bedeutend sind (ähnlich in [KK12]). Die bewusste Planung und Gestaltung der Projektkommunikation ist für ein erfolgreiches Projekt erforderlich [Hi09]. Agiles Vorgehen verfügt in der Regel über ein umfangreiches Kommunikationskonzept, welches die Arbeit im Projekt transparent macht. Um die Arbeit transparent auf der Gesamtprojektebene zu gestalten, liegt die Herausforderung in der Einbeziehung von klassischen und agilen Teilprojekten in eine Gesamtprojektkommunikation. Somit lässt sich die Arbeit transparenter in Bezug auf die Gesamtprojektebene gestalten [KK12]. Hierzu können folgende Kommunikationsregeln beitragen:

- Tägliche und regelmäßige Kommunikation aller Projektmitarbeiter. Sollten die Projektstandorte verteilt sein, bietet sich eine tägliche und regelmäßige Kommunikation aller Projektmitarbeiter durch den Einsatz (digitaler) Medien. Grundsätzlich steht hierbei an erster Stelle die Informationsverteilung bzw. Informationsweitergabe im Team.
- Ermöglichung einer offenen Kommunikationskultur, indem alle Anforderungen, Unklarheiten oder Fehler zusammen abgestimmt werden. Im Falle des hybriden Vorgehensmodells ist dies besonders wichtig, da das Vorgehen einen neuartigen Charakter aufweist. Eine gemeinsame Projektfläche kann als Erfolgsfaktor hinsichtlich der Kommunikation bezeichnet werden. Dabei muss auch Raum zur Klärung von Konfliktfällen und ggf. entstehende Kritik in der täglichen Zusammenarbeit Team geschaffen werden.

- Regelmäßiger Austausch in jeder Iteration zum aktuellen Projektfortschritt, verbleibende Aufwände, nachfolgende Termine.
- Erweiterte Kommunikation, d.h. nicht nur Ergebnisse werden den Auftraggebern, dem Management oder anderen Stakeholdern vermittelt, sondern auch die zugrundeliegende Vorgehensweisen und der damit verbundenen Vorteile. Auf diese Weise verbreitet und verankert sich die Vorgehensweise [Ko12].

Die erfolgreiche Umsetzung einer hybriden Projektkonzeption erfordert eine stimmige Gesamtkonzeption mit einfach praktizierbaren Regeln. Diese bieten den Stakeholdern im Projekt hinreichend Sicherheit, andererseits bieten diese Regeln auch die Basis für eine flexible Vorgehensweise zur Lösung der Projektaufgabe.

## 4.2 Verankerung hybrider Vorgehensmodells im Unternehmen

Zur Etablierung hybrider Konzepte des Projektmanagements im Unternehmen sind eine Reihe vorbereitender Maßnahmen erforderlich. Die folgenden Maßnahmen sind überwiegend aus den Interviews abgeleitet.

### *Richtlinien an das hybride Vorgehensmodell anpassen*

Regulatorische Anforderungen und interne Richtlinien sind auf die hybriden Vorgehensmodelle anzupassen. Die rechtlichen Vorgaben wie z.B. ein Festpreisvertrag oder die Fremdarbeitskraft-Thematik sind dabei entsprechend zu regeln. Die Release-Zyklen sollen sich an beiden Bestandteilen des hybriden Vorgehensmodells orientieren, und nicht nur am Wasserfallmodell.

### *Methodenkompetenzen weiter entwickeln*

Es ist auf die Weiterentwicklung und Anwendung einzelner Methoden zu achten, um sie effizienter umsetzen zu können. Aktuell besteht der Bedarf, z.B. Requirements Engineering und agile Werkzeuge weiterzuentwickeln und im Unternehmen zu verankern. Ein systematischer Einsatz von Methoden und Tools darf nicht vernachlässigt werden.

### *Schulungen und Informationsveranstaltungen organisieren*

Vor dem Start des Projekts im Kick-off Meeting ist es erforderlich, alle relevanten Gruppen (inklusive Mitarbeiter) über die Hintergründe zur Wahl des hybriden Vorgehensmodells informieren. Außerdem sind die ausgewählten hybriden Vorgehensweisen und agile Instrumente zu erläutern.

Des Weiteren können hybride Vorgehensmodelle in unternehmenseigenen Veranstaltungen wie *Projektleiter-Forum* oder anderen Informationsveranstaltungen vorgestellt werden. Die Mitarbeiter brauchen eine klare Vorstellung, wie das hybride Vorgehensmodell in die Praxis umzusetzen ist. Dafür sind entsprechende Use Cases für Trainings auszuarbeiten. Innerhalb welcher nicht nur die theoretischen Grundlagen aufgelistet sondern auch praktische Beispiele integriert werden, wie z.B. ein Plan-Spiel. Empfehlenswert ist es, einen regelmäßigen Austausch der Mitarbeiter hybrider Projekte z.B. im Rahmen einer virtuellen *Hybride Community* im Intranet zu ermöglichen und zu unterstützen.

Die dargestellten Empfehlungen gelten verallgemeinernd für hybride Projekte. Allerdings ist in jedem einzelnen Fall zu analysieren, inwieweit das Projektverständnis, die Erfahrungen, regulatorische und rechtliche Vorgaben sich im konkreten hybriden Vorgehensmodell umsetzen lassen und welcher Anpassungsaufwand benötigt wird [KK12].

## 5 Fazit und Ausblick

Vom richtig ausgewählten Vorgehensmodell hängt maßgeblich der Erfolg eines Projekts ab. Jedes Projekt verfügt über diverse Erfolgsfaktoren. Dazu gehören allgemeingültige Erfolgsfaktoren wie z. B. eine effiziente Zusammenarbeit und Kommunikation sowie die Unterstützung von Stakeholdern. Zudem existieren an das Projektumfeld bzw. an ein konkretes Vorgehensmodell angepasste Erfolgsfaktoren. Somit verfügen hybride Projekte zusätzlich über ihre eigenen spezifischen Erfolgsaspekte. Einerseits sind die hybriden Vorgehensmodelle relativ einfach umzusetzen, da die klassischen und agilen Vorgehensmodelle im Unternehmen bereits eingesetzt werden. Andererseits sind die hybriden Vorgehensweisen als kritisch zu betrachten, weil sich diese aufgrund der Rahmenbedingungen, wie festgelegte klassische Infrastruktur, PM- und Unternehmenskultur, Vorgaben und Richtlinien nur schwer integrieren lassen. Die Ergebnisse der Analyse zeigen u.a. positive Erfahrungen in Bezug auf hybride Vorgehensmodelle. Allerdings besteht hier ein weiterer Anpassungs- und Entwicklungsbedarf für die Etablierung des hybriden Vorgehensmodells im Gesamtunternehmenskontext.

Um unter verschiedenen Voraussetzungen die Praxistauglichkeit des hybriden Vorgehens zu testen und weitere Kombinationsmuster bzw. relevante Konfigurationen hybrider Vorgehensmodelle zu bestimmen, besteht zusätzlich Forschungsbedarf im Bereich hybrider Vorgehensmodelle. Die hybriden Vorgehensweisen werden sich aufgrund ihrer Effektivität immer mehr in der Projektmanagementkultur von Unternehmen etablieren. Zusätzlich werden sich die Kombinationsmuster ständig ändern bzw. vervollständigen. Daher besteht der Bedarf, die erfolgskritischen Faktoren für die Umsetzung der hybriden Projekte weiter zu erforschen.

## Literaturverzeichnis

- [An o.J.] Angermeier, G.: Definition Erfolgsfaktoren. Glossar. Internet: <https://www.projektmagazin.de/glossarterm/erfolgsfaktoren>, Zugriff am 29.05.2014.
- [Cr12] Chroust, G.: Vorgehensmodelle – Quo vadis? Trends und Herausforderungen, In: Gesellschaft für Informatik e.V. - Fachausschuss Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik (WI-MAW), 33. WI-MAW-Rundbrief, 18 (2), Oktober 2012, S. 9-19.
- [Da01] Daum, A.: Erfolgskritische Faktoren des Projektmanagements interner Dienstleistungen. In: Steinle, C.; Bruch, H., Lawa, D. (Hrsg.): Projekt Management, 3.Aufl., Frankfurt am Main: Frankfurter Allgemeine Zeitung, 2001, S. 39-50.

- [En09] Engstler, M.: Organisatorische Implementierung von Informationssystemen an Bankarbeitsplätzen, Wiesbaden: Gabler Edition Wissenschaft 2009.
- [GP14] GPM: Deutscher Project Excellence Award. Internet: [http://www.gpm-ipma.de/ueber\\_uns/gpm\\_awards/deutscher\\_pe\\_award.html](http://www.gpm-ipma.de/ueber_uns/gpm_awards/deutscher_pe_award.html). Zugriff am 06.05.2014.
- [Ha13] Habermann, F.: Hybrides Projektmanagement – agile und klassische Vorgehensmodelle im Zusammenspiel. In: HMD – Praxis der Wirtschaftsinformatik 50 (2013), 293, S. 93-102.
- [Hi09] Hindel, B.; Hörmann, K.; Müller, M.; Schmied, J.: Basiswissen Software-Projektmanagement: Aus- und Weiterbildung zum Certified Professional for Project Management nach iSQI-Standard, 3.Aufl., Heidelberg: dpunkt, 2009.
- [Hü14] Hüsselmann, C.: Agilität im Auftraggeber- Auftragnehmer-Spannungsfeld. Mit hybridem Projektansatz zur Win-win-Situation. In: Projekt Management aktuell 1/2014, S. 38-42.
- [KK12] Kirchhof, M.; Kraft, B.: Agile und klassische Methoden im Projekt passend kombinieren. Hybrides Vorgehensmodell. In: Projekt Magazin, Ausgabe 11/2012, S. 1-11.
- [KL14] Kuhrmann, M.; Linssen, O.: Welche Vorgehensmodelle nutzt Deutschland? In: Engstler, M.; Hanser, E.; Mikusz, M.; Herzwurm, G. (Hrsg., 2014): Projektmanagement und Vorgehensmodelle 2014. Soziale Aspekte und Standardisierung, Lecture Notes in Informatics (LNI) – Proceedings, Volume P-236, Bonn: Gesellschaft für Informatik 2014, S. 17-32.
- [Ko12] Koerting, T.: Gelungener Spagat zwischen Agilität und Wasserfall am Beispiel der SAP-Einführung bei der Deutschen Bank AG. In: Engstler, M., Oestereich, B. (Hrsg.): IT-Projektmanagement 2012+ im Spagat zwischen Industrialisierung und Agilität? GlashüttenHeidelberg: dpunkt 2012, S. 75 – 82.
- [Re10] Reiss, M.: Hybride Vorgehensmodelle. In: Linssen, O.; Greb, T.; Kuhrmann, M.; Lange, D.; Höhn, R. (Hrsg.): Integration von Vorgehensmodellen und Projektmanagement, Aachen: Shaker, 2010, S. 1-14.
- [Re12] Reiss, M.: Hybrides Vorgehensmodelle, In: Gesellschaft für Informatik e.V. - Fachausschuss Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik (WI-MAW), 33. WI-MAW-Rundbrief, 18 (2), Oktober 2012, S. 33-43.
- [Re13] Reiss, M.: Hybrides Projektmanagement: Innovative Management-Architekturen zur Bewältigung neuer Komplexitätsformen in der Projektarbeit. In: Wald, A.; Mayer T.-L.; Wagner, R.; Schneider, C. (Hrsg.): Komplexität. Dynamik. Unsicherheit. Advanced Project Management (Vol.3). GPM Buchreihe Forschung, Buch Nr. F07, Nürnberg: Wünsch Offset-Druck. 2013, S. 94 – 110.
- [RF08] Ruf, W.; Fittkau, T.: Ganzheitliches IT-Projektmanagement: Wissen - Praxis – Anwendungen, München: Oldenbourg 2008
- [SBK14] Sandhaus, G.; Berg, B.; Knott, P.: Hybride Softwareentwicklung. Das Beste aus klassischen und agilen Methoden in einem Modell vereint, Berlin u. Heidelberg: Springer Vieweg, 2014.
- [Sc12] Schneegans, M.: „Klassisches“ versus agiles IT-Projektmanagement. Die Wahl der

richtigen Vorgehensweise, Whitepaper, Hamburg: amendos, 2012.

[SHE11] Schnell, R.; Hill, P. B.; Esser, E.: Methoden der empirischen Sozialforschung, 9 Aufl., München: Oldenbourg, 2011.

# Systematisierung der Auswahl von Vorgehensmodellen durch Kennzahlen

Christian Hennen<sup>1</sup>, Axel Kalenborn<sup>2</sup>, Sascha Stadlbauer<sup>3</sup>, Ingo J. Timm<sup>4</sup>

**Abstract:** Die Auswahl des geeigneten Vorgehensmodells für die Projektabwicklung stellt eine große Herausforderung in Rahmen der Vorbereitung von Software-Projekten dar. In diesem Paper soll dafür auf der Basis von Kennzahlen eine Hilfestellung gegeben werden. Die Kennzahlen dienen dem Abgleich erfolgsrelevanter Projekteigenschaften und den grundlegenden Koordinationsmechanismen der Vorgehensmodelle. Dazu wird eine Systematisierung wichtiger Kennzahlen auf Basis einer Literaturanalyse vorgenommen, die dann als Grundlage für die Modellauswahl herangezogen werden können. Die Ausarbeitungen dienen als Basis für ein noch zu entwickelndes Simulationsmodell für die Auswahl geeigneter Vorgehensmodelle.

**Keywords:** Kennzahlen, Projekte, Koordination, Auswahl von Vorgehensmodellen, Simulation

## 1 Motivation

Die Auswahl des geeigneten Vorgehensmodells stellt eine der wichtigsten Weichen im Rahmen der Vorbereitung eines Software-Projekts dar. Die Vorgehensmodelle legen die Aufbau- und Ablauforganisation eines Projekts fest, bestimmen die verwendeten Koordinationsmechanismen und sind Basis für die Erhebung und Dokumentation der Anforderungen sowie für den Dialog mit dem Kunden. (vgl. [Ti14])

Klassische Methoden, wie das Wasserfall-Modell oder das V-Modell, sind weit verbreitet, gelten jedoch als veraltet und zu starr für moderne Umsetzungsprojekte. Agile Methoden, wie Extreme Programming oder SCRUM, erfreuen sich gegenwärtig großer Beliebtheit, haben aber dort Grenzen, wo feste Projektbudgets wenig Agilität zulassen. Hybride Vorgehensmodelle sollen die Vorteile beider Welten in einem Modell vereinen und damit eine sehr hohe Praxistauglichkeit besitzen. (vgl. [BKS14]) Um die Auswahl eines geeigneten Vorgehensmodells strukturiert vornehmen zu können, ist ein Abgleich der Rahmenbedingungen des umzusetzenden Projekts mit den Koordinationsmechanismen der Vorgehensmodelle nötig. Dazu sind bereits verschiedene Ansätze entwickelt worden [z.B. HE10, Ch03, FH05], die in diesem Papers zunächst als Basis für die Zu-

---

<sup>1</sup> Universität Trier, Professur für Wirtschaftsinformatik I, Behringstraße 21, 54286 Trier, s4chenn@uni-trier.de

<sup>2</sup> Universität Trier, Professur für Wirtschaftsinformatik I, Behringstraße 21, 54286 Trier, axel.kalenborn@uni-trier.de

<sup>3</sup> Universität Trier, Professur für Wirtschaftsinformatik I, Behringstraße 21, 54286 Trier, s4sastad@uni-trier.de

<sup>4</sup> Universität Trier, Professur für Wirtschaftsinformatik I, Behringstraße 21, 54286 Trier, ingo.timm@uni-trier.de

sammenstellung relevanter Kennzahlen einfließen, um eine möglichst präzise Bestimmung relevanter Projektparameter vornehmen zu können. Diese sollen dann in späteren Arbeiten simulativ mit den Eigenschaften der Vorgehensmodelle abgeglichen werden, um so eine strukturierte Auswahl des geeigneten Modells vornehmen zu können.

## 2 Kennzahlen der Projektplanung und -steuerung

In der Literatur findet sich eine große Anzahl an Publikationen, die sich unter anderem mit der Definition von Kennzahlen zur Beurteilung der Erfolgchancen von Projekten befassen [z.B. BK13, Ti13, WM08, WG13, EK11, Jo13]. Diese beschäftigen sich mit dem Aufbau der Projekte, gehen auf Erfolgsaussichten ein, beschreiben situativ geeignete Vorgehensstrategien oder definieren Projekt-Idealtypen. In jedem Beitrag werden unterschiedliche Kennzahlen für die Strukturierung der Projekte verwendet. Auf Basis der bereits genannten Quellen und einer weitergehenden Literaturanalyse werden im Folgenden 53 Kennzahlen erarbeitet und systematisiert, die als Grundlage für die Auswahl eines geeigneten Vorgehensmodells dienen können. Die Kennzahlen fließen in ein noch zu entwickelndes Simulationsmodell ein, das bei der Auswahl des geeigneten Vorgehensmodells helfen soll. Dieser Ansatz ist motiviert von der industriellen Produktentwicklung, in der für die Optimierung des Ressourceneinsatzes und der Prozessabläufe Simulationen eingesetzt werden. In der Simulation liegt auch die Neuartigkeit des Ansatzes, dessen Basis jedoch die Kennzahlen sind, die im Fokus dieses Papers stehen.

### 2.1 Vorgehensweise bei der Kennzahlenermittlung

In die Literaturanalyse wurden Quellen aus den Themenbereichen Projektmanagement, IT-Kostenschätzung und Vorgehensmodelle einbezogen. Die in den Publikationen gefundenen Projekteigenschaften wurden mit geeigneten Wertebereichen sowie einer aussagekräftigen Definition versehen und somit zu einer Kennzahl weiterentwickelt.

Jede Kennzahl wird in der hier vorgeschlagenen Systematisierung über eine Bezeichnung, einen diskreten Wertebereich, eine Definition und der Angabe der Quellen bzw. Nennungen in der Literatur spezifiziert. Dabei werden die folgenden fünf Skalen verwendet:

- Intervallskala: z.B.  $0-\infty$  oder  $1-\infty$
- Intervallskala mit fünf Klassen: z.B.  $< 3, 3-10, 10-50, 50-150, > 150$
- Dichotome Skala: z.B. ja, nein
- 3-5 stufige Rating-Skala: z.B. sehr niedrig, niedrig, mittel, hoch, sehr hoch

Bei den Intervall- und Rating-Skalen stellt die Spezifikation der Skalen eine Herausforderung dar. So ist bspw. bei Intervallskalen die Entropie zu berücksichtigen, so dass offene Randklassen nicht zu einer überproportionalen Kumulation von Projekten und somit zu einer eingeschränkten Aussagekraft und Pragmatik der Kennzahl führen.

Eine weitere Herausforderung stellt die Bewertung der Kennzahlen durch einen Projektleiter dar. Bei einigen Projekteigenschaften erfolgt diese subjektiv, da bspw. die Motivation der Mitarbeiter oder die Dynamik der Anforderungen nicht objektiv messbar sind. Hier ist eine realistische Schätzung durch den Verantwortlichen entscheidend. Kennzahlen, wie die Dauer eines Projekts, das zur Verfügung stehende Budget oder die Anzahl der Mitarbeiter, sind hingegen objektiver Natur und *einfacher* zu messen. Da die Einschätzung der Kennzahlen aber auch von den festgelegten Wertebereichen abhängig ist, wird für jede Ausprägung eine konkrete Definition zugeordnet. Ein Beispiel: Die Ausprägung *sehr gering* der Kennzahl **Komplexität des Codes** ist definiert als *simpel, sehr hohe Wiederverwendung vorgefertigter Bausteine*.

## 2.2 Identifikation von Projekteigenschaften und Kennzahlenermittlung

Der Erfolg eines Projekts wird unter anderem von der **Anzahl der Mitarbeiter** bestimmt. Hinter dieser Hypothese steckt die Annahme, dass bei zunehmender Mitarbeiterzahl der Aufwand für die Kommunikation im Projekt ebenfalls steigt und somit eventuelle Produktivitätsvorteile vermindert oder gar aufgehoben werden. (vgl. [BEJ06], S. 38f) Als weiterer möglicher Grund kann die steigende Dynamik sozialer Prozesse in Gruppen ab sieben Personen angeführt werden (vgl. [BH08], S. 10). Buschermöhle et al. schlagen ferner vor, Projekte mit mehr als zehn Mitarbeitern in kleinere Teilprojekte zu unterteilen (vgl. [BEJ06], S. 38f). Die Anzahl der Projektteam-Mitarbeiter als wichtiger Erfolgsfaktor für agile Projekte findet sich in [CC08], S. 970 und [BK13], S. 34 und 98 wieder.

Ebenso wie die Anzahl der Projektmitarbeiter ist die **Projektdauer** ein erfolgsbestimmender Faktor. Naturgemäß erfordert die Neuartigkeit einer Aufgabe eines Projekts eine gewisse Einarbeitung, bevor die Produktivität der Mitarbeiter ihre volle Stärke erreicht hat. Je länger jedoch ein Projekt läuft, desto wahrscheinlicher wird es, dass Ermüdungserscheinungen auftreten. (vgl. [BEJ06], S. 39f.) Dies wiederum wirkt sich auf die Qualität der erarbeiteten Lösung und die Projektkosten aus.

Neben der **Anzahl der Mitarbeiter** und dem **Budget** ist die geplante **Projektdauer** ein integraler Bestandteil der **Projektgröße** ([Ti13], S. 2). Die Projektgröße ist als kostentreibender Faktor anzusehen ([Sn05], S. 83, [FA10], S. 265ff.). In Anlehnung an die Einteilung von [Ti13] unterscheidet dieser Beitrag die in Tab. 1 angegebenen Ausprägungen für die genannten Bestandteile der Projektgröße. Sie sollen als Beispiel für die erarbeiteten Kennzahlen dienen. Eine vollständige Liste aller Kennzahlen, deren Ausprägungen und Definitionen findet sich im Arbeitspapier zu diesem Beitrag. [He15]

Einen weiteren in der Literatur häufig genannten, erfolgsbestimmenden Faktor von Projekten stellt die **Motivation der Mitarbeiter** dar. Sollten Mitarbeiter des Projektteams nicht ausreichend motiviert sein, um die komplexen Arbeitsprozesse in einem Projekt zu bewältigen, schadet dies direkt und signifikant dem Projekterfolg. (vgl. [BEJ06], S. 42). Neben Buschermöhle et al. erachten auch andere Autoren diesen Zusammenhang als bedeutsam, so z.B. [BK13], S. 19, [Gr13], S. 74 oder [CC08], S. 970. Als Definition für

die Kennzahl „Motivation der Mitarbeiter“ dient in Anlehnung an [Ps02], S. 1087: „Motivation bezeichnet das auf emotionaler Aktivität beruhende Streben des Mitarbeiters [des Auftragnehmers / des Auftraggebers] nach Zielen oder wünschenswerten Zielobjekten. Sie beschreibt die Gesamtheit der Beweggründe, die zur Handlungsbereitschaft führen.“

Die häufigsten Nennungen als Merkmale eines erfolgreichen Projekts verzeichnen in der zugrunde liegenden Literatur die **Kompetenz der Mitarbeiter** und die **Erfahrung der Mitarbeiter**. Die Kompetenz der Mitarbeiter (auch Skill-Level ([EK11], S. 7) oder Know-How ([Gr13], S. 74)) hängt eng mit der Erfahrung der Mitarbeiter zusammen, da davon auszugehen ist, dass erfahrene Mitarbeiter in den meisten Fällen ein höheres Kompetenzniveau vorweisen können. Durch ein *kompetentes* Team wird die Qualität der erarbeiteten Lösung erhöht und es werden weniger Fehler gemacht. Hierdurch sinkt der Aufwand durch Nacharbeiten oder Korrekturen. (vgl. [BEJ06], S. 42)

Kennzahl	Ausprägungen					Definition
Anzahl der Mitarbeiter	< 3	3-10	10-50	50-150	> 150	Gibt an, wie viele Mitarbeiter insgesamt am Projekt mitarbeiten.
Dauer des Projekts	< 0,4	0,4-5	5-50	50-500	> 500	Die für die Umsetzung des Software-Projekts geschätzten Personenjahre.
Budget	< 0,05	0,05-0,5	0,5-5	5-50	> 50	Definiert den für die Umsetzung des Projekts zur Verfügung gestellten Geldbetrag in Mio. Euro.

Tab. 1: Kennzahlen der Projektgröße, in Anlehnung an Ti13, S. 2

Neben den zuvor genannten, eher die Projektorganisation oder die Mitarbeiter betreffenden Projekteigenschaften, ist die **Komplexität** als eine das zu erstellende Produkt betreffende Eigenschaft zu nennen. Diese beeinflusst den Dokumentationsaufwand des Produkts und hängt direkt mit dem zu leistenden Aufwand und somit auch mit der Projektdauer zusammen. (vgl. [BEJ06], S. 40) Während die meisten Publikationen nur allgemein von der Komplexität sprechen, z.B. [Sn05], S. 7, [BEJ06], S. 40, wird in [Jo13] eine Einteilung in die **Komplexität des Problems, des Codes und der Daten** (vgl. [Jo13], S. 2) vorgenommen. Dieser Unterteilung folgt der vorliegende Beitrag, um eine genauere Bewertbarkeit dieser eher abstrakten Eigenschaft zu ermöglichen.

Als eine Kennzahl auf Ebene des Produktes ist der **Innovationsgrad** zu nennen, die den Anteil an neuartigen Konzepten und Technologien im zu entwickelnden Produkt beschreibt. Diese steht im Zusammenhang mit der **Anforderungsdynamik** und dem **Umfang der Produktdokumentation**. Zudem kann ein hoher Innovationsgrad bedeuten,

dass dem Faktor Zeit und damit der Schnelligkeit der Entwicklung Vorrang gegenüber dem Produktumfang gegeben wird (**Priorität Zeit vs. Umfang**). ([EK11], S. 10).

Es wurde bereits deutlich, dass der Erfolg eines Projekts sowohl von den daran teilnehmenden Individuen (z.B. **Motivation der Mitarbeiter**) als auch von den Rahmenbedingungen (z.B. **Anzahl der Mitarbeiter**) bestimmt wird. Weitere Faktoren finden sich bei Betrachtung der sozialen Interaktionen der Projektpartner sowie der Kompetenz des Auftragnehmers.

Um die Zusammenarbeit zwischen Auftragnehmer (AN) und Auftraggeber (AG) bewerten zu können, werden Faktoren, wie die **Kenntnis des Auftragnehmers über den Kunden** oder das **Vertrauen** zwischen den beiden Parteien, herangezogen. Letzteres beschreibt das subjektive Vertrauen zwischen AN und AG und berücksichtigt ebenso die Zugehörigkeit zu einem Kulturkreis (vgl. [LL10], S. 76f. und [Ko03], S. 27). Die Ausprägungen reichen hier von sehr schwach bis sehr stark.

Um den Reifegrad einer Unternehmung (auch Prozessreifegrad (vgl. [Sn05], S. 32)) messen und damit weitere Schlüsse über die Kompetenz für komplexere Projekte ziehen zu können und die Kennzahl **CMMI-Level** eingeführt. Diese nutzt Appraisals des CMMI-DEV-Modells von Initial bis Optimizing. Damit besteht die Möglichkeit, bestehende Prozesse zu bewerten und in einer Kennzahl zu formalisieren (vgl. [BK13] S. 348).

Zuletzt lässt sich festhalten, dass der Erfolg eines (IT-)Projekts vor allem auch durch die **Präzision**, die **Bekanntheit** und die **Dynamik der Anforderungen** bestimmt wird. Analog gilt dies für die Projektziele. Sollten die Anforderungen oder Ziele in einem Projekt nicht hinreichend konkretisiert oder nicht allen Mitarbeitern vollständig bekannt sein, kann sich dies negativ auf den Projekterfolg auswirken. (vgl. [Ti13] S. 4f., [LL10], S. 76f., [EK11], S. 7) Diesem Umstand wird in dem in diesem Beitrag erarbeiteten Kennzahlensystem Rechnung getragen.

### 2.3 Systematisierung der Kennzahlen

Die Systematisierung wurde mit Hilfe eines Whiteboards und dem Ziel, Abhängigkeiten zwischen Kennzahlen zu identifizieren und Aggregation von Kennzahlen zu ermöglichen, durchgeführt. Die resultierende Matrix untergliedert in den Spalten Auftragnehmer, Auftraggeber und Projekt und in den Zeilen Mitarbeiterorganisation, Projektorganisation und Produkt (siehe Abbildung 1) In den Spalten wird differenziert, ob die Erhebung der Messgröße durch den AG oder den AN erfolgt bzw. deren Zusammenarbeit betrifft. Unter Mitarbeiterorganisation fallen Kennzahlen, die Bezug zu den am Projekt beteiligten Personen haben. Projektorganisation umfasst alle dem Projektmanagement direkt zugehörigen Messgrößen, wohingegen unter Produkt alle Kennzahlen, die das zu entwickelnde Produkt direkt beschreiben, fallen.

Die Systematisierung ermöglicht eine Differenzierung verschiedener Kennzahlen, wie Motivation der Mitarbeiter in nach AG und AN getrennte Größen. Bei der Betrachtung der Kennzahlenverteilung in der Matrix ist zu erkennen, dass jene aus dem Bereich der Projektorganisation eher auf die AN konzentriert sind, während sich die Kennzahlen der Produktebene eher im Bereich des AG konzentrieren. Dies begründet sich darin, dass die Anforderungen an das Produkt typischerweise durch die Vorstellungen und die evtl. bereits vorhandenen Produkte des Auftraggebers geprägt sind, während die Ausführung des Projekts eher durch den Auftragnehmer wahrgenommen wird. Auf der Ebene der Mitarbeiterorganisation liegt der Schwerpunkt im Bereich des Projekts, welche die Interaktion beider Parteien beschreibt.

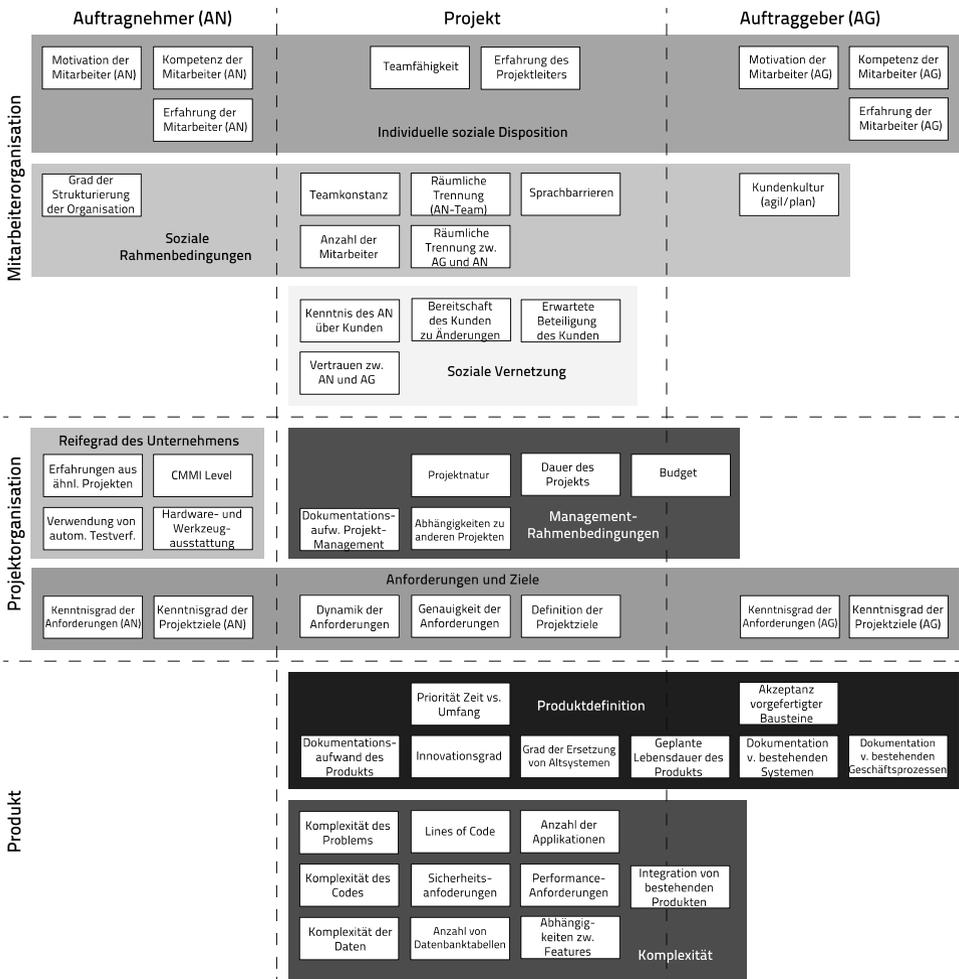


Abb. 1: Vollständiges Kennzahlensystem in Matrixform

Kennzahlen zusammenhängender Themenkomplexe werden zusätzlich in die acht farblich markierten Gruppen eingeteilt: *individuelle soziale Disposition, soziale Rahmenbedingungen, soziale Vernetzung, Reifegrad des Unternehmens, Management-Rahmenbedingungen, Anforderungen und Ziele, Produktdefinition und Komplexität*. Jede dieser Gruppen kann jeweils in einer Kennzahl aggregiert werden, so dass die Identifikation von risikobehafteten Projektbereichen und die Einleitung von entsprechenden Gegenmaßnahmen unterstützt wird.

### **3 Relevanz von Kennzahlen für die Auswahl des Vorgehensmodells**

Die hier erarbeiteten Kennzahlen sollen die Auswahl eines geeigneten Vorgehensmodells in einem konkreten Projektkontext unterstützen. Dazu werden zunächst grundlegende Koordinationsprinzipien der Vorgehensmodelle identifiziert und in einem zweiten Schritt mit den extrahierten Kennzahlen abgeglichen.

#### **3.1 Grundlegende Koordinationsmechanismen ausgewählter Vorgehensmodelle**

Zur Festlegung der grundlegenden Koordinationsmechanismen von Vorgehensmodellen wird der situative Ansatz der Organisationsforschung gewählt. Dieser erklärt die Ausprägungen von Koordinationsmechanismen in Abhängigkeit der Umweltbedingungen und beschreibt die Tätigkeit des Organisierens als Planung und Kontrolle des Einsatzes von Ressourcen und Steuerungsmechanismen zur Aufgabenerfüllung. [KK92] Damit wird den Anforderungen im Projektmanagement begegnet, die in der Wahl einer passenden Vorgehensweise in Anhängigkeit der Projektgegebenheiten bestehen. Dabei gilt der Grundsatz, dass mit steigender Spezialisierung bzw. Arbeitsteilung der Koordinationsbedarf zunimmt. [KK92] Zur Bewältigung des entstehenden Koordinationsbedarfs werden verschiedene Instrumente, wie persönliche Weisung, Selbstabstimmung und Vorauskoordination durch Pläne und Programme, eingesetzt. An dieser Stelle können jedoch nur erste Ergebnisse der Analyse vorgestellt werden, diese sollen beim Aufbau des Simulationsmodells anhand von Praxisprojekten erweitert und präzisiert werden.

#### **Koordinationsmechanismen klassischer Vorgehensmodelle**

In den klassischen Vorgehensmodellen, wie dem Wasserfall-Modell oder V-Modell, wird der Vorauskoordination großer Raum geschenkt. Die Projekte sollen in der Phase der Anforderungserhebung und des Designs so detailliert geplant werden, dass eine strukturierte und sequentielle Abarbeitung möglich ist. [FK07, S. 9f] Als wesentliche Koordinationsinstrumente kommen daher die Programmierung und die persönliche Weisung zum Einsatz. Durch Programme sollen detaillierte Abläufe für die Umsetzung, den Test sowie die Einführung der Software beschrieben und damit vorauskoordiniert werden. Durch persönliche Weisungen oder Änderung der Programme wird im laufenden Projekt auf Änderungen reagiert; die Projektleitung hat so Einfluss auf die Abwicklung. Grundlage für die detaillierte Planung sind fest definierte Anforderungen als Basis für

die Umsetzung. Änderungen an diesen Anforderungen sind nur schwer zu kompensieren.

Die Programmierung als Koordinationsmechanismus ist aufwändig und erfordert eine sehr detaillierte Vorbereitung des Projekts. Die persönliche Weisung ist ein flexibles und schnell einsetzbares Instrument. Ihm fehlt jedoch die Nachhaltigkeit, wodurch ein hoher individueller Koordinationsaufwand für die Projektleitung entsteht. Außerdem ist die persönliche Weisung bei den Projektbeteiligten unbeliebt; sie fordern Mitsprache und Partizipation an den Entscheidungen, wodurch ihre Motivation gefördert wird, die Entscheidungsfindung jedoch langwieriger wird.

### **Agile Methoden**

Bei den agilen Methoden, wie Extreme-Programming oder Scrum, soll im Projekt flexibel auf Änderungen reagiert werden. Daher wird eine Vorauskoordination nicht über einen langen Zeitraum vorgenommen und beruht eher auf der Festlegung von Planungszielen als auf der Beschreibung detaillierter Arbeitsabläufe bzw. Programme. [FK07, S. 30ff] Im laufenden Projekt erfolgt die Koordination hauptsächlich durch Selbstabstimmung der Projektbeteiligten. Diese treffen sich z.B. zum Daily-Scrum und stimmen sich über aktuelle Probleme und die nächsten Aufgaben ab. Im Sprint werden dann die Ziele für die nächste Phase mit den Anforderungen des Kunden abgeglichen, um flexibel auf eine dynamische Umwelt und ihren Anforderungen reagieren zu können.

Während die Planung nicht eine so detaillierte Vorauskoordination wie die Programmierung erfordert und damit weniger aufwändig ist, ist das Koordinationsinstrument der Selbstabstimmung sehr zeitintensiv. Die Lösungsfindung wird im Dialog mit den Projektbeteiligten durchgeführt und dauert für den Einzelnen deutlich länger als die Vergabe eines konkreten Auftrags durch die Projektleitung. Dabei soll eine höhere Zufriedenheit der Mitarbeiter durch Partizipation und eine verbesserte Entscheidungsqualität erreicht werden, da die Projektmitarbeiter ihre hohe Problemlösungskompetenz einbringen können.

### **Hybride Methoden**

In den hybriden Methoden erfolgt eine Kombination der zuvor genannten Koordinationsmechanismen. Agilität findet sich hauptsächlich in der Umsetzungsphase, die Koordination des Gesamtprojekts erfolgt jedoch weitgehend klassisch. [BKS14] Damit werden alle oben definierten Koordinationsinstrumente eingesetzt und es hängt vom individuellen Projekt ab, welche Mechanismen zum Zuge kommen.

## **3.2 Identifikation und Selektion relevanter Kennzahlen für die Modellauswahl**

Im Folgenden sollen beispielhaft idealtypische Ausprägungen einiger der Kennzahlen für klassische und agile Vorgehensweisen diskutiert werden. Die klassischen Methoden sind auf eine frühe detaillierte Anforderungserhebung und Vorauskoordination ausgelegt. Dabei sollten vor allem die Kennzahlen des Bereiches **Anforderungen und Ziele**

betrachtet werden. Die **Definition der Projektziele** sollte möglichst genau und die **Genauigkeit der Anforderungen** möglichst hoch sein ([Ti13], S. 4f.). Dies trägt ebenso dazu bei, den hohen Aufwand für das Beseitigen zu spät erkannter Zielabweichungen zu reduzieren, wie ein hoher **Kenntnisgrad der Anforderungen** ([FA10], S. 265ff.) und eine geringe **Dynamik der Anforderungen** im Projektverlauf ([HE10], S. 137). Nach [BK13] und [HE10] sollten große Projekte eher mit klassischen Vorgehensmethoden entwickelt werden, da sie sich durch schwergewichtige Prozesse auszeichnen (**Projektgröße**). In Projekten, die nach einem klassischen Vorgehensmodell erarbeitet werden, sollte der Umfang des zu liefernden Produkts wichtiger sein als das Einhalten des Auslieferungsdatums (**Priorität Zeit vs. Umfang**). Dieses folgt aus der wasserfallartigen Konkretisierung des Produktes. Da klassische Vorgehensmodelle plangetrieben sind, sollte die Kundenkultur hier ebenfalls plangetrieben sein (**Kundenkultur agil/plan**). Das bedeutet, dass kundenseitig ein Verständnis für die klassische Vorgehensstrategie, wie für die Bedeutung von Meilensteinen und formalen Prozessen, vorliegen und eine längerfristige Planung die Regel sein sollte. ([HE10], S. 137)

Für die agilen Modelle steht vor allem das hohe Maß an Kommunikation und Selbstabstimmung im Vordergrund, um flexibel auf Änderungen reagieren zu können. Daher sind hier besonders die Kennzahlen aus den Bereichen der sozialen Rahmenbedingungen und der sozialen Vernetzung heranzuziehen. Niedrige **Sprachbarrieren** ([Ti13], S. 5) und möglichst geringe **räumliche Trennung** ([CC08], S. 970) sind dem agilen Vorgehen sehr förderlich. Zudem eignen sich nach [BK13], [HE10] und [CC08] agile Vorgehensmodelle hauptsächlich bei Projekten mit einer geringen **Anzahl der Mitarbeiter** und einer kurzen **Projektdauer** (bis zu 7 Mitarbeiter und Laufzeit bis zu einem Jahr, [BK13], S. 98). Dies wiederum lässt sich mit erhöhtem Aufwand durch den Koordinationsmechanismus der Selbstabstimmung bei einer höheren Personenzahl erklären. Ein weiterer Faktor bei der Entscheidung für eine agile Vorgehensstrategie stellt **erwartete Beteiligung des Kunden** dar. Sie sollte bei agilen Projekten im Idealfall hoch sein und es sollte ein gutes Verhältnis zwischen AG und AN herrschen ([CC08], S. 970), welches in den Kennzahlen **Vertrauen zw. AN und AG** und **Kenntnis des AN über Kunden** ausgedrückt wird. Der Kunde sollte idealerweise bereit sein, Änderungen an der ursprünglichen Planung anzunehmen (**Bereitschaft des Kunden zu Änderungen**) ([BK13], S. 98f.) und er sollte ein Verständnis für die agile Vorgehensweise mitbringen (**Kundenkultur agil/plan**) ([HE10], S. 137).

Die Aussagen über klassische und agile Vorgehensstrategien lassen sich für die jeweils andere Strategie in vielen Fällen umkehren, wie bereits am Beispiel der **Kundenkultur** oder der **Projektdauer** angedeutet. Das bedeutet z.B. für die **Priorität von Zeit vs. Umfang**, dass bei einer Priorisierung der Zeit-Komponente eher eine agile Vorgehensweise zu bevorzugen ist, während bei einer Priorisierung der Einhaltung eines gewissen Produktumfangs eher eine klassische Vorgehensstrategie zu wählen ist.

## 4 Fazit

Mit der hier vorgestellten Kennzahlenmatrix soll die Auswahl von Vorgehensmodellen für ein Projekt auf systematisch unterstützt werden. Dazu wurden 53 erfolgsrelevante Kennzahlen aus dem Projektmanagement abgeleitet, spezifiziert und systematisiert. Außerdem wurden grundlegende Koordinationsmechanismen von Vorgehensmodellen identifiziert und in Zusammenhang mit Ausprägungen bestimmter Kennzahlen analysiert. Abschließend wurde exemplarisch an idealtypische Ausprägungen die Eignung von Vorgehensmodellen diskutiert.

Die hier vorgestellte Systematik ist das Ergebnis einer Literaturanalyse und bedarf daher einer Validierung anhand tatsächlicher Projektpraxis. Dies soll in zukünftigen Arbeiten geschehen. Hierbei soll neben der Untersuchung der Kennzahlenmatrix auf Vollständigkeit und Erweiterungs-/Verkleinerungsbedarf auch eine feinere Abbildung der einzelnen Ausprägungen für die Wahl eines Vorgehensmodells vorgenommen werden.

Motiviert von der industriellen Produktentwicklung, in der Simulationen für die Optimierung des Ressourceneinsatzes und der Prozessabläufe eingesetzt werden, sollen zukünftige Arbeiten die hier vorgestellte Kennzahlenmatrix als Basis nehmen, um einen Ansatz zur Modellierung und Simulation von Softwareprojekten zu entwickeln. In diesem Paper wurde dazu durch die Vorstellung der Kennzahlenmatrix und deren Abgleich mit den wesentlichen Koordinationsmechanismen ausgewählter Vorgehensmodelle ein erster Beitrag geleistet.

## Literaturverzeichnis

- [BEJ06] Buschermöhle, R.; Eekhoff, H.; Josko, B. 2006: SUCCESS 2006. Erfolgs- und Misserfolgsfaktoren bei der Durchführung von Hard- und Software- Entwicklungsprojekten in Deutschland, BIS-Verlag, Oldenburg.
- [BH08] Becker, C.; Huber, Dr. E. 2008: Die Bilanz des (Miss)-Erfolges in IT-Projekten. Harte Fakten und weiche Faktoren. In: White Paper, Pentaeder, Ludwigsburg.
- [BK13] Broy, M.; Kuhrmann, M. 2013: Projektorganisation und Management im Software Engineering, Springer.
- [BKS14] Berg, B.; Knott, P.; Sandhaus, G. 2014: Hybride Softwareentwicklung. Das Beste aus klassischen und agilen Methoden in einem Modell vereint, Xpert.press.
- [CC08] Chow, T.; Cao, D. 2008: A survey study of critical success factors in agile software projects. In: Journal of Systems and Software, 81/6, S. 961-971.
- [Ch03] Chravant, J. 2003: Project management methodologies. Selecting, implementing and supporting methodologies and processes for projects, Hoboken, Wiley.
- [EK11] Engels, G.; Kremer, M. 2011: Situational Software Engineering. In: INFORMATIK 2011 - Informatik schafft Communities, Bd. 192.

- [FH05] Filß, C.; Höhn, R.; Höppner, S.; Schumacher, M.; Wetzel, H. 2005: Rahmen zur Auswahl von Vorgehensmodellen, In: Entscheidungsfall Vorgehensmodell 12, S. 185-229.
- [FA10] Fischer, C.; Aier, S. 2010: IT-Projektkostenschätzung. Ein pragmatischer Ansatz. In: Multikonferenz Wirtschaftsinformatik, S. 259-272.
- [FK07] Fritzsche, M.; Keil, P. 2007: Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie. In: Interner Bericht des Projekts IO-SEW TUMI0717, Technische Universität München.  
[http://www4.in.tum.de/publ/papers/TUM-I0717\\_neu.pdf](http://www4.in.tum.de/publ/papers/TUM-I0717_neu.pdf), Stand 15.05.2015.
- [Gr13] Gross, B. 2013: Projekte: Chancen, gefahren Erfolgsfaktoren, In Grau, N.; Wagner, R. Hrsg.: Basiswissen Projektmanagement: Grundlagen der Projektarbeit, S. 65-76.
- [He15] Hennen, C. et al.: Systematisierung der Auswahl von Vorgehensmodellen durch Kennzahlen. Arbeitspapier, [https://www.uni-trier.de/fileadmin/fb4/prof/INF/W11/Publikationen/Arbeitspapier\\_SAVK.pdf](https://www.uni-trier.de/fileadmin/fb4/prof/INF/W11/Publikationen/Arbeitspapier_SAVK.pdf), Stand: 20.05.2015
- [HE10] Heinemann, M.; Engels, G. 2010: Auswahl projektspezifischer Vorgehensstrategien, In Linssen, O. et al. (Hrsg.), Shaker, Aachen.
- [Jo13] Jones, C. 2013: Function points as a universal software metric. In: ACM SIGSOFT Software Engineering Notes 38.4 2013, S. 1-27.
- [KK92] Kieser, A.; Kubicek, H. 1992: Organisation, 3. Auflage, Mannheim, Bremen.
- [Ko03] Kollischan, K. 2003: Erfolgsbewertung von Vorgehensmodellen. In Petrasch et al. (Hrsg.): Praxistauglichkeit von Vorgehensmodellen, S. 15-34, Shaker.
- [LL10] Ludewig, J.; Lichter, H. 2010: Software Engineering. Grundlagen, Menschen, Prozesse, Techniken, dpunkt. verlag,.
- [Ps02] Pschyrembel, W. 2002: Klinisches Wörterbuch, De Gruyter, Berlin.
- [Sn05] Sneed, H. 2005: Software-Projektkalkulation. Praxiserprobte Methoden der Aufwandschätzung für verschiedene Projektarten, Carl Hanser Verlag, München.
- [Ti13] Tiemeyer, E. (Hrsg.) 2013: Handbuch IT-Management: Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. Carl Hanser Verlag.
- [Ti14] Tiemeyer, E. 2014: Handbuch IT-Projektmanagement. Vorgehensmodelle, Managementinstrumente, Good Practices, Hanser Verlag.
- [WG13] Wagner, R.; Grau, N. (Hrsg.) 2013: Basiswissen Projektmanagement - Projekte planen, Risiken erkennen, Symposium Publishing.
- [WM08] Wiczorrek, H.-W.; Mertens, P. 2008: Management von IT-Projekten. Von der Planung zur Realisierung. Springer, 2008.



# IT-Systementwicklungsprojekte der öffentlichen Hand: Der Einfluss des Vergaberechts auf die Verwendung agiler Methoden

Thomas Süptitz<sup>1</sup>, Felix Ruppert<sup>2</sup>, Torsten Eymann<sup>3</sup>

**Abstract:** Das Standard-Vorgehensmodell der öffentlichen Verwaltung, das V-Modell XT, erlaubt umfangreiche Tailoring-Möglichkeiten. Dies schließt insbesondere die Integration agiler Methoden ein. Grenzen erfahren diese Anpassungsmöglichkeiten jedoch durch das Vergaberecht, wenn dieses eine „eindeutige“ und „erschöpfende“ Leistungsbeschreibung fordert. Dieser Gedanke steht damit im Widerspruch zum „Agilen Manifesto“: Diesem sind die ursprüngliche Leistungsbeschreibung oder eine umfangreiche Planung weniger wichtig als eine intensive Einbindung des Kunden. Der vorliegende Beitrag erläutert das Dilemma und stellt denkbare Auswege vor.

**Keywords:** Vergaberecht; IT-Systementwicklungsprojekte; V-Modell XT; agile Methoden; Scrum

## 1 Einleitung

Die Ergebnisse der CHAOS Studie 2013 [St13] legen nahe, dass nur 39 % aller IT-Projekte erfolgreich abgeschlossen werden; die überwiegende Mehrzahl der Projekte wird hingegen teurer, dauert länger, verfehlt den geplanten Funktionsumfang (43 %) oder scheitert völlig (18 %). Die Ursachen werden auf ehrgeizige Zeitpläne, die Komplexität oder sich häufig ändernde bzw. unklare Anforderungen zurückgeführt [Me09].

Um diesen Herausforderungen zu begegnen sollen Vorgehensmodelle eine systematische Planung und Durchführung von IT-Projekten ermöglichen. So ist das V-Modell XT der Standard für Behörden der Bundesverwaltung [Hö08]. Agile Vorgehensmodelle wie Scrum versprechen hingegen eine deutlich produktivere Softwareentwicklung [Gl10]. Bestrebungen beide Modelle miteinander zu verknüpfen verwundern deshalb kaum. Durch die öffentliche Hand sind jedoch vergaberechtliche Überlegungen zu treffen, die einer Integration agiler Ansätze in das V-Modell XT entgegenstehen können.

Mit diesem Beitrag betrachten wir solche Überlegungen. Dazu werden wir im unmittelbaren Anschluss einen kurzen Überblick über das V-Modell XT und agile Methoden geben (Abschnitt 2). Im dritten Abschnitt werden wir die vergaberechtlichen Problem-

---

<sup>1</sup> Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik Universitätsstraße 30, 95447 Bayreuth, Thomas.Sueptitz@uni-bayreuth.de.

<sup>2</sup> Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik Universitätsstraße 30, 95447 Bayreuth, Felix.Ruppert@uni-bayreuth.de.

<sup>3</sup> Universität Bayreuth, Lehrstuhl für Wirtschaftsinformatik Universitätsstraße 30, 95447 Bayreuth, Torsten.Eymann@uni-bayreuth.de.

stellungen erläutern und insbesondere auf die Bedeutung der Leistungsbeschreibung eingehen. Abschließend (Abschnitt 4) fassen wir die Ergebnisse zusammen und stellen mögliche Auswege aus dem Dilemma vor.

## 2 Vom V-Modell XT und agilen Methoden

### 2.1 Das V-Modell XT

Das V-Modell XT als DAS Prozessmodell für Aufträge der öffentlichen Hand unterstützt Prozesse, indem Abläufe und Ergebnisse vorgegeben werden. Ein Projekt wird dazu in Produkte und diese verändernde Aktivitäten gegliedert. Mehrere dieser Aktivitäten werden gruppiert, um so Bausteine für die modulare Zusammensetzung des V-Modell XT zu bilden [RF08]. Dabei unterscheidet das Vorgehensmodell folgende drei Projekttypen [BR05]:

- (I) Das Systementwicklungsprojekt eines Auftraggebers, das beschreibt wie Auftraggeber eine Ausschreibung vorbereiten und anhand der Angebote einen Auftragnehmer auswählen.
- (II) Das Systementwicklungsprojekt eines Auftragnehmers, um ein Angebot zu erstellen und ein System zu entwickeln. Für die Systementwicklung selbst werden die inkrementelle, komponentenbasierte und prototypische Entwicklung als Entwicklungsstrategien vorgesehen.
- (III) Ein Projekt zur Einführung und Pflege eines organisationsspezifischen Vorgehensmodells in einer Organisation.

Das V-Modell XT sieht vor, dass durch den Auftraggeber die Anforderungen hinsichtlich des Liefer- und Leistungsumfangs festzulegen und in einem Lastenheft zu beschreiben sind [Ba10]. Das Vorgehen setzt daher eine Planungsphase voraus und erfordert bereits zu Beginn einen genauen Kenntnisstand über die Struktur des gewünschten Endprodukts.

### 2.2 Agile Methoden

Neben dem V-Modell XT erfreuen sich agile Methoden, für die Scrum stellvertretend stehen kann [Gl10], großer Beliebtheit.<sup>4</sup> Sie versprechen eine erfolgreiche Verwendung insbesondere auch bei Großprojekten [Oe08]. Dazu konzentrieren sich agile Methoden auf vier Grundsätze:<sup>5</sup> So geht eine funktionsfähige Software einer umfassenden Dokumentation vor,<sup>6</sup> während die Zusammenarbeit mit dem Kunden und das rasche Reagieren auf Änderungen im Vordergrund stehen und wichtiger sind als Vertragsverhandlungen

---

<sup>4</sup> Untersuchungen zeigen, dass das V-Modell XT und Scrum zusammengenommen in nahezu der Hälfte aller Fälle als Prozessmodell Einzug gehalten haben [KL15].

<sup>5</sup> Siehe dazu auch das Agile Manifesto (<http://agilemanifesto.org/iso/de/>).

<sup>6</sup> Wenngleich das nicht bedeutet, dass überhaupt nicht dokumentiert wird, sondern nur anders.

oder das starre Befolgen eines Plans. Damit beschreiben sie ein Vorgehen mit geringem Regelungsumfang, wenig Bürokratie und dem Hauptaugenmerk auf einer gesteigerten Interaktion [Tr12].

So ist auch die Planungsphase auf elementare Grundbedingungen beschränkt oder entfällt gänzlich. Kompensiert wird dies durch einen ständigen Austausch der Beteiligten und der gemeinsamen, durchführungsbegleitenden Erarbeitung von Meilensteinen, die es zu erreichen gilt [BS10]. Ziele werden gemeinsam festgelegt und auf geänderte Vorstellungen kann unmittelbar reagiert werden. Kurze Planungs- und Entwicklungsphasen wechseln sich ab oder laufen zum Teil parallel. Das Produkt wird schrittweise verbessert, wobei diese Entwicklung in kurzen Abschnitten erfolgt. Je nach Ansatz werden diese als Zyklen, Sprints oder Iterationen bezeichnet. Am Ende eines solchen kurzen Abschnitts erhält der Kunde lauffähige, getestete Funktionen und kann das Ergebnis bewerten [HRS04].

### 2.3 Die Integration agiler Methoden

Dass es – in Abhängigkeit der Aufgabenstellung - durchaus sinnvoll sein kann beide Welten zusammenzuführen zeigt u. a. [Ha13]. Grundsätzlich unterstützt das V-Modell XT traditionell eine sequentielle Durchführung eines Projekts. Kennzeichnend für eine sequentielle Durchführung ist die Gliederung in Phasen, welche nacheinander durchlaufen werden. Erst nach Abschluss einer Phase kann das einzelne Produkt geprüft und die nächste Phase begonnen werden [RF08]. Dennoch werden auch iterative und inkrementelle Ansätze unterstützt [Hö08]. So ist die prototypische Entwicklungsstrategie für die Einsatzzwecke agiler Methoden geeignet. Diese beruht schließlich auf der Erkenntnis, dass es nicht immer möglich ist alle Anforderungen an eine IT-Lösung im Vorfeld zu beschreiben. Dabei werden die Entscheidungspunkte bis einschließlich der Projektbeauftragung übernommen. Als Schnittstelle zur agilen Methodik dient nun der Entscheidungspunkt „Iteration geplant“. Der bisherige Zyklus vom Systementwurf zur Systemintegration kann durch entsprechende agile Zyklen „ersetzt“ werden. So kann beispielsweise die V-Modell-Iteration als Release im Sinne von Scrum begriffen werden. Ein Verständnis als Sprint kommt in dieser Abstraktionsstufe noch nicht in Betracht, da die V-Modell-Iteration eine Folge von Entscheidungspunkten beinhaltet, die nicht in einem zeitlich kurz bemessenen Sprint erreicht werden können [CRW11].

Unabhängig von diesen Detailüberlegungen stellt sich die Frage, ob das Systementwicklungsprojekt des Auftraggebers und ein „agiles“ Systementwicklungsprojekt eines Auftragnehmers unter den geschilderten Prämissen zusammenpassen; schließlich erfordert die Durchführung einer Ausschreibung eine umfangreiche Planungsphase. Wesentliche Ursache dafür sind die folgenden vergaberechtlichen Erwägungen.

## 3 Der Einfluss des Vergaberechts

### 3.1 Grundlagen des Vergaberechts

Die Träger der öffentlichen Verwaltung sind verpflichtet bei der Beschaffung von Mitteln und Dienstleistungen die vergaberechtlichen Normen zu beachten. Insbesondere müssen Aufträge ausgeschrieben werden, sofern sie einen bestimmten Schwellenwert übersteigen.<sup>7</sup> Diese Grenzen liegen bei 134.000<sup>8</sup> bzw. 207.000 EUR<sup>9</sup> und richten sich nach dem (geschätzten) Auftragswert ohne Umsatzsteuer.

Aufträge unterhalb der Schwellenwerte werden nach dem ersten Abschnitt der „Vergabe- und Vertragsordnung für Leistungen – Teil A“ (VOL/A) vergeben. Obwohl die Verordnung nur verwaltungsinterne Geltung hat, gewährt die Rechtsprechung dem benachteiligten Bieter Rechtsschutz in Form eines Unterlassungsanspruchs.<sup>10</sup> Aus diesem Grund sollte die VOL/A auch bei Vergaben unterhalb der Schwellenwerte beachtet werden [HP10, MK12]. Oberhalb der Schwellenwerte erlangt der zweite Abschnitt der VOL/A (sog. VOL/A-EG) den Rang einer Rechtsverordnung.<sup>11</sup> Daher sind durch die Vergabestellen die Bestimmungen der VOL/A-EG einzuhalten [He13].

Inhaltlich werden durch die VOL/A drei wesentliche Grundsätze normiert:<sup>12</sup> Aufträge sind regelmäßig im *Wettbewerb*, im Wege *transparenter* Vergabeverfahren und bei *Gleichbehandlung* aller Bieter zu vergeben. Als zentrales Element fungiert der Wettbewerbsgrundsatz [Lu09]. Das Transparenzgebot erfordert eine klare und eindeutige Vergabebekanntmachung mitsamt aller Bedingungen und Modalitäten. Ferner sind die Wertungskriterien der Vergabeentscheidung offenzulegen. Schließlich hat auch die, der Vergabeentscheidung zugrundeliegende, vergleichende Wertung der Angebote sachlich nachvollziehbar zu sein [Fe11].

---

<sup>7</sup> § 97 Abs. 1 GWB.

<sup>8</sup> Für obere und oberste Bundesbehörden.

<sup>9</sup> Für die übrigen Träger der öffentlichen Verwaltung.

<sup>10</sup> OLG Düsseldorf, Beschluss vom 16. November 2010 (VII-Verg 50/10), OLG München, Beschluss vom 6. Dezember 2012 (Verg 25/12).

<sup>11</sup> Durch einen direkten Verweis in § 4 VgV.

<sup>12</sup> § 2 VOL/A bzw. § 2 VOL/A-EG.

### 3.2 Die Bedeutung der Leistungsbeschreibung

Die Vergabestellen sind prinzipiell angehalten öffentlich auszuschreiben [MK12].<sup>13</sup> Dazu wird eine unbegrenzte Anzahl von Unternehmen öffentlich zur Angebotsabgabe aufgefordert. Jeder Interessent ist berechtigt ein Angebot abzugeben. Eine Einengung des Bieterkreises ist nicht möglich. Der öffentlichen Ausschreibung entspricht bei einer Vergabe oberhalb der Schwellenwerte das „offene Verfahren“. Auch dieses ist gegenüber anderen Vergabearten vorrangig.

Die VOL/A bzw. VOL/A-EG<sup>14</sup> verlangen für die öffentliche Ausschreibung nach einer *eindeutigen* und *erschöpfenden* Leistungsbeschreibung. Diese ist somit Grundlage der späteren Angebote. „Eindeutig“ und „erschöpfend“ wird regelmäßig als gründlich sowie vollständig verstanden [We13]. Zudem muss die Leistungsbeschreibung so beschaffen sein, dass ohne intensive Auslegungsbemühungen seitens der Bieter keine Restbereiche verbleiben dürfen, welche nicht schon klar umrissen sind.<sup>15</sup>

Diese Überlegungen haben zur Folge, dass bereits vor der eigentlichen Systementwicklung sämtliche Anforderungen an die spätere Anwendung feststehen müssten.

### 3.3 Die funktionale Leistungsbeschreibung

Neben der geschilderten „konventionellen“ Leistungsbeschreibung kennen VOL/A und VOL/A-EG<sup>16</sup> die Möglichkeit einer funktionalen Leistungsbeschreibung. Diese ist dadurch gekennzeichnet, dass sie nicht den Weg zur Leistung, sondern „nur“ deren Zweck beschreibt [We13]. Obwohl die funktionale Leistungsbeschreibung den Weg der Umsetzung offen lässt ist es allgemein anerkannt, dass das Erfordernis der „eindeutigen“ und „erschöpfenden“ Leistungsbeschreibung auch hier Anwendung finden muss [Sc11]. Kennzeichen der funktionalen Leistungsbeschreibung ist jedoch, dass Teile der Planung auf den Bieter übertragen werden um dessen Expertise zu nutzen [Be09]. Dennoch darf der Auftraggeber nicht die gesamte Planungstätigkeit auf die Bieter verlagern. Er muss vielmehr seine eigene Planung insoweit betreiben, als dass die Kriterien für die spätere Bewertung, das Leistungsziel, die Rahmenbedingungen sowie die wesentlichen Einheiten der Leistung in der Weise bekannt sind, dass mit Veränderungen nicht mehr zu rechnen ist [We13]. Der Beschreibungsumfang gilt als gewahrt, wenn in jeder, sich aus den Vergabeunterlagen ergebenden Hinsicht vergleichbare Angebote zu erwarten sind [Be09].

---

<sup>13</sup> Unter Umständen kann auch eine beschränkte Ausschreibung/ein nicht offenes Verfahren oder eine freihändige Vergabe durchgeführt werden.

<sup>14</sup> § 7 Abs. 1 VOL/A bzw. § 8 Abs. 1 VOL/A-EG.

<sup>15</sup> OLG Saarbrücken, Beschluss vom 29 September 2004 (1 Verg 6/04).

<sup>16</sup> § 7 Abs. 2, S. 2 a) VOL/A bzw. § 8 Abs. 2, S. 1, Nr. 2 VOL/A-EG.

Die Wahl dieser Form der Leistungsbeschreibung liegt im Ermessen der Vergabestellen<sup>17</sup> und ist zulässig, wenn die zu beschreibende Leistung komplexerer Natur ist und detailliert kaum abzubilden wäre [Be09]. Dies kann für IT-Projekte der Fall sein.

Betrachtet man den eigentlichen Vergleich der Angebote, so vermag dies zunächst für eine umfängliche Planungsphase sprechen: Wer klar definiert, was er genau haben möchte braucht letztlich nur den Preis zu vergleichen. Die funktionale Leistungsbeschreibung setzt ihrer Art entsprechend jedoch eine gewisse Eigeninitiative der Bieter voraus und sorgt so für divergierende Angebote [We13]. Diese können regelmäßig nicht einzig bezüglich des Faktors „Preis“ verglichen werden. Dieser Mangel an Vergleichbarkeit ist jedoch hinzunehmen, solange er aus den unterschiedlichen operativ-konzeptionellen Ansätzen der Bieter resultiert [We13]. Konsequenz ist eine Abkehr vom Zuschlagskriterium „Preis“ zugunsten anderer Kriterien wie beispielsweise dem Konzept des Bieters oder damit einhergehender Effizienz sowie Qualität der Leistungsbeschreibung.

## 4 Zusammenfassung

Im Vergleich zu klassischen sequentiellen Vorgehensmodellen verläuft eine „agile Planung“ anders; statt einmal vollständig, wird häufiger und situativ geplant [Tr14]. Dem steht allerdings der Trias des deutschen Vergaberechts aus „Wettbewerb“, „Transparenz“ und „Gleichheit“ entgegen. Neben dem Wettbewerbsprinzip als bieterschützende Institution fordert das Transparenzgebot von den Vergabestellen eine übersichtliche und klar abgestufte Verfahrensgestaltung. Dazu gehört auch die eindeutige Vergabebekanntmachung. Hinzu kommt der Gleichbehandlungsgrundsatz, der eine willkürfreie Auswahl des Angebots sicherstellen soll.

Mithin muss eine Leistungsbeschreibung den Bietern eine zweifelsfreie Kalkulationsgrundlage bieten, um so die Vergleichbarkeit der Angebote zu sichern. Fehlerhafte Leistungsbeschreibungen ermöglichen den Bietern ein Nachprüfungsverfahren. Die Ergebnisse eines solchen reichen von einer Wiederholung<sup>18</sup> bis zur Aufhebung des Vergabeverfahrens.<sup>19</sup> Zudem können Bieter auch Schadenersatzansprüche geltend machen.<sup>20</sup>

Eine erhebliche Reduktion der Planungsphase kann durch die Begrenzung auf eine funktionale Leistungsbeschreibung erreicht werden. Mit dieser werden lediglich die Grundfunktionen des Endprodukts beschrieben, während diese gerade nicht jeden einzelnen Schritt vorgibt. Folge ist eine erhöhte Eigeninitiative der Bieter.

---

<sup>17</sup> OLG München, Beschluss vom 10. Dezember 2009 (Verg 16/09).

<sup>18</sup> OLG Koblenz, Beschluss vom 13. März 2006 (14 W 164/06).

<sup>19</sup> OLG Celle, Beschluss vom 2. September 2004 (13 Verg 11/04).

<sup>20</sup> OLG Naumburg, Beschluss vom 15. Dezember 2005 (1 U 5/05).

Einen weiteren möglichen Ausweg beschreibt auch [Me09], der ein dreistufiges anstatt eines einstufigen Vergabeverfahrens vorsieht. Maßgebliches Ziel dieses Vorschlages ist es, den zeitlichen und inhaltlichen Rahmen eines Projekts realistisch zu definieren. Dazu wird der eigentlichen Ausschreibung, die der Auswahl eines Realisierers dient, eine „funktionale Ausschreibung“ vorangestellt. Diese richtet sich unmittelbar an IT-nahe Beratungsunternehmen. Nach Auswahl eines Dienstleisters obliegt ihnen die Aufgabe, die vorliegende Grobkonzeption weiter zu verfeinern. Am Ende dieses Prozesses sollen daraus abgeleitete und verfeinerte Funktionalitäten sowie Leistungen stehen.

Dies bedeutet aber auch, dass weder die funktionale Leistungsbeschreibung noch ein mehrstufiges Vergabeverfahren den Verzicht auf eine Planungsphase im Sinne agiler Methoden erlauben. Erreichbar sind lediglich eine Reduktion des Planungsaufwands und eine Steigerung der Planungsqualität.

## Literaturverzeichnis

- [Ba10] Bauer, N.: Handbuch IT-Projektmanagement. Vorgehensmodelle, Managementinstrumente, Good Practices. Hanser, München, 2010.
- [Be09] Bernhardt, U.: § 7 VOB/A, Rn. 14. In (Ziekow, J.; Völlink, C. Hrsg.): Vergaberecht. GWB/VgV/VOB A/VOL A. C. H. Beck, München, 2009.
- [BR05] Broy, M.; Rausch, A.: Das neue V-Modell XT. In: Informatik-Spektrum, 2005, 28, S. 220-229.
- [BS10] Beardwood, J. P.; Shour, M.: Risk Management and Agile Software Development: Optimizing Contractual Design. In: Computer Law Review International (CRI), 2010, S. 161-170.
- [CRW11] Canditt, S.; Rauh, D.; Wittmann, M.: Brückenschlag. Das V-Modell XT mit Scrum inside. In: OBJEKTSpektrum, 2011.
- [Fe11] Fehling, M.: § 97 GWB, Rn. 65. In (Pünder, H.; Alexander, C. Hrsg.): Vergaberecht. GWB, VgV, SektVO, VOL/A, VOB/A, VOF, Haushaltsrecht, öffentliches Preisrecht - Handkommentar. Nomos, Baden-Baden, 2011.
- [GI10] Gloger, B.: Scrum. In: Informatik-Spektrum, 2010, 33, S. 195-200.
- [Ha13] Habermann, F.: Hybrides Projektmanagement. Agile und klassische Vorgehensmodelle im Zusammenspiel. In: Praxis der Wirtschaftsinformatik (HMD), 2013, 50, S. 93-102.
- [He13] Hertwig, S.: Praxis der öffentlichen Auftragsvergabe. Systematik, Verfahren und Rechtsschutz. C. H. Beck, München, 2013.
- [Hö08] Höhn, R.: Das V-Modell XT. Anwendungen, Werkzeuge, Standards. Springer, Berlin, Heidelberg, 2008.

- [HP10] Holleben, M. K. v.; Probst, M.: IT-Verträge der öffentlichen Hand - Änderungen durch die Vergaberechtsreform. In: Computer und Recht (CR), 2010, 26, S. 349-352.
- [HRS04] Hruschka, P.; Rupp, C.; Starke, G.: Agility kompakt. Tipps für erfolgreiche Systementwicklung. Spektrum Verlag, Heidelberg, Berlin, 2004.
- [KL15] Kuhrmann, M.; Linssen, O.: Vorgehensmodelle in Deutschland: Nutzung von 2006 - 2013 im Überblick. In: Management der Anwendungsentwicklung und -wartung (WI-MAW), 2015, 21, S. 32-47.
- [Lu09] Luber, H.: Der formalistische Angebotsausschluss, das Wettbewerbsprinzip und der Grundsatz der sparsamen Mittelverwendung im Vergaberecht. In: Vergaberecht (VergabeR), 2009, 9, S. 14-25.
- [Me09] Mertens, P.: Schwierigkeiten mit IT-Projekten der öffentlichen Verwaltung. In: Informatik-Spektrum, 2009, 32, S. 42-49.
- [MK12] Müller-Hengstenberg, C.; Kirn, S.: Öffentliches Vergaberecht und moderne IT-Softwareentwicklung. In: MultiMedia und Recht (MMR), 2012, 15, S. 3-8.
- [Oe08] Oestereich, B.: Agiles Projektmanagement. In: Praxis der Wirtschaftsinformatik (HMD), 2008, 45, S. 18-26.
- [RF08] Ruf, W.; Fittkau, T.: Ganzheitliches IT-Projektmanagement. Wissen, Praxis, Anwendungen. Oldenbourg, München [u.a.], 2008.
- [Sc11] Schellenberg, M.: § 7 VOL/A, Rn. 99. In (Pünder, H.; Alexander, C. Hrsg.): Vergaberecht. GWB, VgV, SektVO, VOL/A, VOB/A, VOF, Haushaltsrecht, öffentliches Preisrecht - Handkommentar. Nomos, Baden-Baden, 2011.
- [St13] The Standish Group International, Inc.: CHAOS Manifesto 2013. Think Big, Act Small. <https://www.standishgroup.com>, 2013.
- [Tr12] Trepper, T.: Agil-systemisches Softwareprojektmanagement. Springer, Wiesbaden, 2012.
- [Tr14] Trümmer, J.: Was ist dran an Agilen Mythen. In: Management der Anwendungsentwicklung und -wartung (WI-MAW), 2014, 20, S. 56-58.
- [We13] Weyand, R.: Vergaberecht. Praxiskommentar zu GWB, VgV, SektVO, VSVgV, VOB/A 2012, VOL/A, VOF. C. H. Beck, München, 2013.

# Hybride Vorgehensmodelle in der Versionserstellung – ein Praxisbeitrag

Dimitri Petrik<sup>1</sup>

**Abstract:** abilex GmbH ist ein KMU, welches neben anderen Dienstleistungen „Software-Test as a Service“ anbietet. Einen großen Teil der Beauftragungen bilden Tests spezifischer produktdatenintensiver Point-of-Sale Software im Enterprise Bereich. Diesen Service fasst abilex als „Versionserstellung“ zusammen, wobei die Aufbereitung jeder neuen Version von abilex als ein Projekt betrachtet wird. Dabei werden, zum Teil bewusst und zum Teil unbewusst, sowohl klassische- als auch agile Vorgehensmodelle des Projektmanagements angewandt. Daher hat dieser Praxisbeitrag das Ziel, die Herausforderungen und Besonderheiten der Anwendung von hybriden Projektmanagement-Methoden im Bereich des Managed Software Tests in unternehmensübergreifenden Konsortien zu liefern.

**Keywords:** Projektmanagement, hybride Vorgehensmodelle, hybrides Projektmanagement, agiles Projektmanagement, Projektmanagement in Teilprojekten, Projektmanagement im Konsortium, Projektmanagement im Wertschöpfungsnetzwerk, Software-Test, Testing as a Service, Managed Software Testing, Versionserstellung

## 1. Einleitung

### 1.1 Allgemeines

Viele Projektmanagement-Ansätze, sind in der IT-Branche entstanden und entweder als klassische oder als agile Ansätze bekannt. Um Gründe für Probleme in Projekten aufzuzeigen, wurde das sequentielle Wasserfallmodell entwickelt, das als klassisches Modell gilt [Sc10]. Später entstanden agile Methoden, um bekannte Probleme klassischer Modelle zu lösen. Ein bekannter Vertreter des agilen Projektmanagements ist „Scrum“, das sich am agilen Manifest für die schlanke Softwareentwicklung orientiert. [He10]. Doch die agilen Methoden haben ebenfalls ihre Grenzen, so bieten sie bspw. weniger Hilfe in Form von unterstützenden Tools und Prozessen oder einer expliziten Risikosteuerung während eines Projekts. Einen Überblick über mögliche Einschränkungen beider Vorgehensmodelle liefert die Übersicht von Sandhaus, Berg und Knott, welche bekannte agile und klassische Vorgehensmodelle vergleicht.

---

<sup>1</sup> abilex GmbH, MBKS-Versionserstellung, Löffelstr. 4, 70597 Stuttgart, Dimitri.Petrik@abilex.de

Kriterium	Wasserfallmodell	V-Modell	XP	Scrum
Phasenmodell	✓	✓	X	X
Skalierbarkeit	X	X	✓	✓
Flexibilität	X	X	✓	✓
Vorgehen	Sequentiell	Sequentiell	Iterativ	Iterativ
Methoden	X	✓	✓	✓
Werkzeuge	X	✓	X	X
Unterstützende Prozesse	X	✓	X	X
Praxisnähe	X	✓	✓	✓
Risikosteuerung	X	✓	X	X
Qualitätssicherung	✓	✓	✓	✓

Abb. 1: Stärken und Schwächen von klassischen und agilen Vorgehensmodellen im Projektmanagement [KK14]

Die Praxis zeigt, dass sequentielle Modelle wie das V-Modell nach wie vor in vielen Projekten eingesetzt werden. Unterschiedliche Anforderungsspezifitäten erfordern unterschiedliche Vorgehensmodelle für den erfolgreichen Projektabschluss [Hü14] – so eignen sich klassische Modelle besser bei unerfahrenen Teams und eher klar definierten Anforderungen, wohingegen ein eingespieltes Team selbst bei unklaren Anforderungen oder neuen Technologien im agilen Vorgehen bessere Ergebnisse liefert [Hü14]. Doch findet eine strikte Trennung in der Praxis statt? Oftmals werden vorteilhafte Merkmale beider Vorgehensweisen im gleichen Projekt angewendet. Die Vermischung ist unter dem Begriff „hybride Vorgehensmodelle“ bekannt. [SBK14]. Dies zeigt eine 2011 durchgeführte Untersuchung.

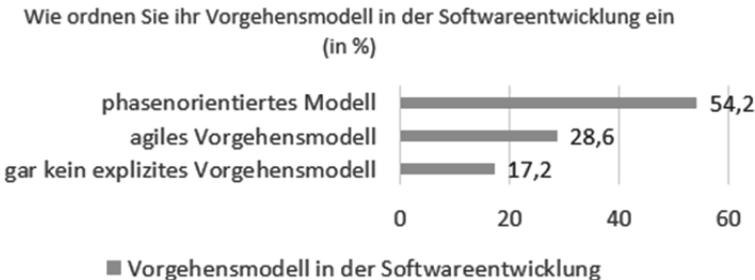


Abb. 2: Umfrage zur Verbreitung von Vorgehensmodellen bei der Softwareerstellung [SU11]

Dabei dominiert bei sequentiellen Modellen das allg. V-Modell mit 36,6% und Scrum bei den agilen Modellen mit 57%. An zweiter Stelle kommen bei beiden Vorgehensarten eigene Modelle vor, was auf hybride Vorgehensmodelle hindeutet. Mit diesem Beitrag

möchte der Autor, anhand der Vorgehensweise in Projekten seines Arbeitgebers im Bereich des Softwaretests, die Anwendung der hybriden Methoden untersuchen und die Herausforderungen oder gar Konflikte zwischen der Praxis und der Theorie zu hybriden Vorgehensmodellen im Projektmanagement aufzeigen.

## 1.2 Einordnung der Versionserstellung im PM und in der Praxis

Die abilex GmbH ist ein KMU, das aus dem Umfeld des Produktinformationsmanagements entstanden ist. Die Datenimplementierung und der begleitende Softwaretest von produktdatenintensiver Point-of-Sale-Software (PoS-Software) in der Automobilindustrie wurden als Kernkompetenz erarbeitet. In der Dienstleisterrolle untersucht abilex die Kundensoftware auf Fehlerwirkungen und Defekte, es findet jedoch kein Debuggen im Sinne der Beseitigung der Fehlerzustände statt. [SL06] Die gefundenen Fehlerwirkungen und Defekte werden analysiert und anschließend an den Kunden oder an den Softwarehersteller kommuniziert. Die Dienstleistung bietet somit im Rahmen des Outsourcings den Software-Test als Service an. Geschuldet den besonderen Softwaremerkmalen der A-Kunden, hat sich abilex im Rahmen des Softwaretests ebenfalls auf die Produktdatenprüfung und –aufbereitung spezialisiert. In diesem Zusammenhang ist der Begriff der Versionserstellung entstanden – als Sonderform des produktdatenintensiven Softwaretests. So wird korrekten Datenbeständen innerhalb der Versionserstellung eine Top-Priorität eingeräumt. Besonders wichtig sind marktspezifische Anforderungen an die Software, da sie international eingesetzt wird und marktspezifische Unterschiede aufweist, wie zum Beispiel landesspezifische Gesetzgebungen, Währungen, Steuern, Ausstattungen oder Fahrzeugkonfigurationen. Deshalb gehören zur Versionserstellung von PoS-Software die korrekte Produktdatenaufbereitung und die Lokalisierung der marktspezifischen Anforderungen.

Die Versionserstellung entspricht allen Merkmalen eines Projekts nach DIN-Norm und wird deshalb in Projektform geführt.<sup>2</sup> Fehlerfreie Markteinführung markiert die Zielvorgabe, während der Kunde eine Zeitvorgabe definiert. Ebenfalls werden der Preis und der Leistungsumfang im Vorfeld definiert und priorisiert und die begrenzte Anzahl der Projektmitglieder für die Versionserstellung vereinbart. Zudem wird die Versionserstellung klar von anderen Aktivitäten des Unternehmens abgegrenzt. Somit eignet sich die Versionserstellung als valides Beispiel für ein Projekt.

---

<sup>2</sup>Vorhaben, das im Wesentlichen durch die Einmaligkeit aber auch Konstanz der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie zum Beispiel Zielvorgabe, zeitliche, finanzielle, personelle und andere Begrenzungen; Abgrenzung gegenüber anderen Vorhaben; projektspezifische Organisation.

Insgesamt umfasst die Versionserstellung folgende sequentiellen Prozesse: Vorbereitung, Dateneingangsprüfung, Aufbereitung von Datenständen, Test und die Marktauslieferung. Die nächste Abbildung ordnet die Versionserstellung innerhalb der Softwareerstellung ein und verdeutlicht die einzelnen Aufgaben - sog. Arbeitspakete innerhalb des Versionserstellungsprojekts.

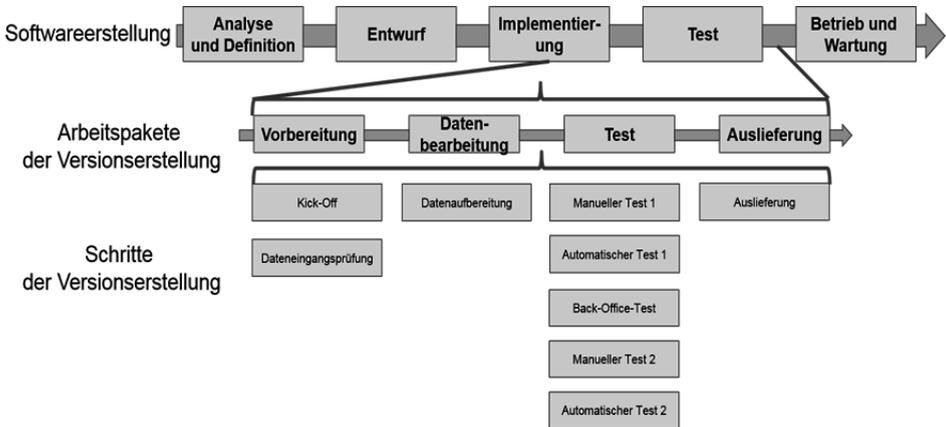


Abb. 3: Einordnung im Softwareerstellungsprozess und Darstellung der einzelnen Subtätigkeiten im Rahmen der Versionserstellung

Dabei wirft die sequentielle Vorgehensweise folgende Fragen auf:

- Können in der Vorbereitung alle Fragen geklärt werden?
- Was passiert wenn Bugs erfasst werden?
- Ist ein einmaliger Testdurchlauf ausreichend?

Die Bewältigung dieser offenen Punkte erfordert eine iterative Vorgehensweise im Projekt. Deshalb ist eine Untersuchung dazu sinnvoll, wie agil die sequentielle Vorgehensweise in der Versionserstellung ist. Darüber hinaus ordnet die Abbildung 3 die Versionserstellung als Teilprojekt innerhalb des ganzheitlichen Softwareprojekts der PoS-Software aus der Kundensicht ein, was zur Forschungslücke führt.

### 1.3 Forschungslücke

Die Wissenschaft konzentriert sich bei hybriden Vorgehensmodellen auf ganzheitliche Softwareerstellungsprojekte – die Möglichkeit einer horizontalen Kooperation zwischen unterschiedlichen Serviceanbietern wird nicht beachtet. Doch genau solche Konstellationen, bestehend aus rechtlich unabhängigen, agilen KMUs, kommen in der Realität oft vor. Die in der Literatur dargestellte Situation findet man eher bei simplen Software-Projekten (simple Smartphone Apps) oder bei großen Softwareanbietern, welche in der Lage sind, die Softwareerstellung ganzheitlich allein zu bewältigen. Aus vielerlei be-

kannten Gründen wird den Unternehmen dennoch empfohlen, sich auf die Kernkompetenzen zu konzentrieren. Unter dieser Prämisse agiert auch abilex als Anbieter von Managed Software Testing in einem Verbund aus unterschiedlichen Unternehmen und übernimmt für seine Kunden aus der Automobilindustrie den Softwaretest. Neben Kostenvorteilen hat diese in der Praxis durchaus verbreitete Lösung den Vorteil von höherer Qualitätssicherung der Software – unabhängige Testteams liefern in der Regel Software von höherer Qualität ab, weil trotz eines relativ hohen Einarbeitungsaufwands, externe Experten spezielles Know-How besitzen und eine emotionale Bindung gegenüber den Entwicklerteams nicht gegeben ist [SL05]. Im vorliegenden Praxisbeitrag wird von Großkonzernen als Kunden der Software-Test als eigenständiges Projekt an Dienstleister vergeben. Diese Aufgabenverteilung zwischen Unternehmen begünstigt die Anwendung hybrider Vorgehensmodelle. Da aus der Sicht des Autors die Anwendung hybrider Projektmanagementmethoden bei Unternehmenskooperationen in der Wissenschaft bisher wenig behandelt wurde, deutet dies auf eine vorhandene Forschungslücke hin.

#### **1.4 Aufbau und Literaturanalyse**

Im vorliegenden Praxis-Beitrag wird die Versionserstellung aus der Projektmanagement-sicht beschrieben und systematisch auf das Vorhandensein der Merkmale klassischer und agiler Vorgehensmodelle untersucht, um daraus die Anwendung hybrider Modelle abzuleiten. Im Unterschied zu bereits erschienen Publikationen wird ein Bottom-Up Ansatz verfolgt, da der Schwerpunkt der Untersuchung am Beispiel der Versionserstellung als Dienstleistung und Teilprojekt betrachtet werden Managed Software Tests als Dienstleistung stellen einerseits als Kernkompetenz einen hohen Wert für abilex dar und leistet einen hohen Qualitätsbeitrag zum erfolgreichen Abschluss des Gesamtprojekts aus der Kundensicht – der erfolgreichen Markteinführung einer neuen PoS-Software. Die Projektuntersuchung umfasst ebenfalls die Rollen der Projektbeteiligten und die eingesetzten Tools. Zum Schluss wird das Vorgehen in der Versionserstellung schematisch dargestellt und in das ganzheitliche Projekt zur PoS-Software eingeordnet. Anschließend werden Empfehlungen für weitere Forschungsrichtungen zu hybriden Vorgehensmodellen im Projektmanagement abgeleitet.

#### **1.5 Ergebnisse der Literaturanalyse**

Der Mangel an Publikationen zur Anwendung hybrider Vorgehensmodelle in Unternehmensverbänden wird anhand der Literaturanalyse deutlich. Durchsucht wurden die Datenbanken Scencedirect, IEEE, ACM und Ebscohost, wobei die Suche über Ebscohost auch die Suche bei Business Source Premier, Academic Search Premier und EconLit beinhaltet. Die Literaturrecherche umfasste nur Papers und Journals, erschienen ab 2004. Erste Literaturrecherche mit folgenden Suchbegriffen<sup>3</sup> ergab keine Treffer. Lediglich

---

<sup>3</sup> String1: „project management“ AND „hybrid approach“ AND „value network“

eine Vernachlässigung des Suchbegriffs „value network“ im Titel, Abstract und Keywords brachte Ergebnisse zu hybridem Projektmanagement. So beschreiben Lehman und Sharma eine hybride Projektmanagementform für „Softwareentwicklung as a Service“ bei IBM, welche agile Modelle für die Prototypenerstellung und die anfängliche Entwicklung anwendet und die dann nach Erreichen eines bestimmten Reifegrads zu klassischen nach Plan geführten Modellen übergeht [LS11]. Bedingt durch den Mangel an Veröffentlichungen wurde eine weitere systematische Literaturliteraturanalyse anhand der Titel, Abstracts und Keywords durchgeführt<sup>4</sup>, wobei die Suche sowohl hybrides als auch agiles Vorgehen berücksichtigt hat. Überraschend war die unterschiedliche Einordnung des hybriden Projektmanagements. So sehen einige Paper hybride Vorgehensmodelle als Vorstufe zu agilem Projektmanagement, während andere hybrides Projektmanagement für geeigneter und daher notwendiger als rein agiles Vorgehen halten. So zeigen Serrador und Pinto in einer empirischen Untersuchung dass die Erfolgsaussichten eines Projekts mit dem Anteil agiler Elemente zusammenhängen und positiv korrelieren. In dem Artikel weist die Mehrheit der untersuchten Projekte (aus unterschiedlichen Branchen und zum Teil unternehmensübergreifend) von 65% agile Merkmale gemischt mit klassischem Vorgehen auf, womit sie als hybrid gelten. Dabei wird hybrides Vorgehen als temporärer Zustand auf dem Weg zu vollständig agilen Projekten angesehen [SP15]. Diese Einordnung wird ebenfalls von Stettina und Hörz vertreten, die in ihrer Untersuchung zum agilen Portfoliomanagement von zeitlichen Lernprozessen bei Unternehmen sprechen, die nötig sind um agil vorzugehen. Für agiles Vorgehen müssen Routinen konsolidiert werden und während dieses Lernprozesses wird nicht komplett agil vorgegangen. Somit werden hybride Vorgehensmodelle erneut als vorübergehender Zustand auf dem Weg zu vollständig iterativen Vorgehensmodellen betrachtet. [SH14] Weiteres Paper zu großen outsourceten Projekten liefert zwei wichtige Merkmale zur Übertragung auf die Versionserstellung. So behaupten die Autoren, dass agiles Projektmanagement neue Werte im Unternehmen integriert, wofür Zeit benötigt wird. Außerdem kommen die Autoren zur Ansicht, dass agiles Vorgehen weitere Prozesse im Unternehmen auslöst und sogar die Einbindung der Kunden in das Projekt verstärkt und die Kundenkommunikation agil werden lässt. [Da12]. Weitere Untersuchung zu Projekten mit Beteiligung mehrerer Agenten von Domann et al. zeigt, dass klassische Methoden den Projekterfolg nicht sichern können und besonders zur Steuerung von Projekten mit einer Mehrzahl an Akteuren braucht es agile Werkzeuge und Tools. Laut dem Paper können klassische Vorgehensmodelle kleine agile Projekte mit kleinen Projektteams, dynamischen Anforderungen und vielen Stakeholdern schlecht steuern. Andererseits, passen Werkzeuge agiler Vorgehensmodelle oftmals nicht zu Projekten mit mehreren Akteuren (unabhängige Unternehmen oder Beziehungen zwischen Kunden und Dienstleistern) [Dj14]. Zur Effizienz von agilen Projekten sagen Serrador und Pinto, dass agile oder hybride Projekte ähnlichen Planungsaufwand im Vorfeld benötigen wie die klassischen Projekte verursachen. Sollte grundlegende Planung erst während des Projektablaufs stattfinden, so erhöht sich der Planungsaufwand bei agilem Vorgehen gegenüber dem klassischen

---

<sup>4</sup>String2: "Project" AND ("Agile" OR "Hybrid\*" "network" "value\*")

[SP15]. Martakis und Daneva haben mittels einer empirischen Untersuchung festgestellt, dass hybride Vorgehensmodelle am besten skalierbar sind und die Anforderungsabhängigkeiten als Risiko am besten lösen. Zudem wenden große Unternehmen Risikosteuerungsmethoden an, während kleine Unternehmen sich mehr auf Kommunikation zwischen Projektmitgliedern verlassen [MD13]. Spundak hingegen sieht klassische Vorgehensmodelle als besser geeignet zur Durchführung großer Projekte an, unabhängig davon ob die Projekte groß hinsichtlich ihrer Dauer, Teamgröße, Anzahl an Anforderungen oder Komplexität sind. Weiteres Paper von van Waanderburg und van Vliet beschreibt, dass auch große Unternehmen in Projekten versuchen agil vorzugehen. Die Autoren sehen hybrides Vorgehen bei großen Unternehmen als eine Koexistenzphase, während der Transformation von klassisch zu agil [WV13]. Diese Einordnung deckt sich mit der Sicht von Serrador und Pinto. Außerdem behaupten van Waanderburg und van Vliet, dass agiles Projektmanagement sich ebenfalls für große Unternehmen eignet, obwohl in großen Unternehmen Strukturen und Positionen existieren, die Mehraufwand während des Projektverlaufs generieren. Trotzdem können sie nicht so schnell umgestaltet werden. Aus diesem Grund seien hybride Vorgehensmodelle in IT-Projekten der befragten Großkonzerne verbreitet [WV13]. Aus dieser Einordnung heraus stellen die Autoren mangelndes „Business Involvement“ als Hindernis auf dem Weg zu agilem Projektmanagement dar, da es vom Management oder Kunden ausgeht. Weiterhin beschreiben die Autoren die Bottom-Up Ausbreitung des agilen Projektmanagements bei beiden befragten Großkonzernen – es erfolgte in einzelnen Teams, um später weitere Bereiche der Organisation zu beeinflussen. Weitere Merkmale klassischer, agiler und hybrider Vorgehensmodelle wurden anhand deutscher Quellen bei Springerlink und aus dem ProjektMagazin herausgearbeitet und finden Anwendung im Kapitel 3.

## 2. Hybride Vorgehensmodelle in der Versionserstellung

### 2.1 Projekttablauf in der Versionserstellung

In diesem Kapitel wird der Projekttablauf am Beispiel der Versionserstellung einer PoS-Software beschrieben. Wie in der Abbildung 4 zu sehen, wird bei abilex (auch Dienstleistungsunternehmen) das Projekt in vier Arbeitspakete unterteilt. Die Unterteilung sieht zwar nach einer statischen Planung mit sequentiellen Paketen aus, aber es besteht ebenfalls aus diversen Aspekten des agilen Projektmanagements.

Für den erfolgreichen Projektabschluss werden Kompetenzelemente gemäß der National Competence Baseline der GPM angewendet. Schon vor dem Projektbeginn werden durch die Versionserstellung als Dienstleister die Stakeholder analysiert - es sind überwiegend die Auftraggeber als Kontroll- und Koordinationsinstanz und die Verkäufer als zukünftige Nutzer der Software. Die Weiterentwicklung der PoS-Software im Konsortium wird vom Softwarelieferanten durchgeführt. Bereits im Vorfeld wird ein Release-Plan vom Kunden in Absprache mit dem Dienstleistungsanbieter vorbereitet und ein Rahmenplan für die Fertigstellung einer bestimmten Version erstellt. Aus der Kunden-

sicht ist es deshalb ein nach der Anforderungsspezifikation aufgestellter Plan, der aus der Sicht der Versionserstellung bereits vor Projektbeginn feststeht.

Für die Versionserstellung fängt das Projekt mit einem landesspezifischen Kick-Off Termin an, an dem landespezifische Vertriebspartner des Kunden teilnehmen. Beim Kick-Off werden mögliche Projektverzögerungen und Risiken besprochen – sei es aufgrund einer verspäteten Datenlieferung, welche vom ursprünglichen Release-Plan abweicht, dem Urlaub wichtiger Projektmitglieder oder nationaler Feiertage der jeweiligen Länder. Demnach werden im Kick-Off Risiken frühzeitig erkannt, [GPM08] um die qualitativen und zeitlichen Ziele des Versionserstellungsprojekts nicht zu gefährden. Aus der Kundensicht (Top-Down) ist der Kick-Off mit der Versionserstellung aufgrund der vorgelagerten Weiterentwicklung der Software und der Datenaktualisierung zwar mitten im Projektverlauf anzusiedeln, für abilex (Bottom-Up) markiert der Kick-Off jedoch den Projektanfang.

Nach dem Kick-Off erhält die Versionserstellung die benötigten Datenupdates oder auch eine neue Programmversion. Bei neuen Programmversionen oder auch nach Kundenwunsch fängt anschließend der Basistest an, um die Basisfunktionalitäten der Software zu prüfen. Wenn die Versionsersteller Fehlerwirkungen in der Basisfunktionalität nachweisen, kann es als Kriterium für den Abbruch der Versionserstellung gelten. Gefundene Fehlerwirkungen werden über ein Online-Ticket System an den Hersteller kommuniziert. Die Kommunikation kann auch der Kunde mitverfolgen und er kann so zeitnah über mögliche Verzögerungen informiert werden. Der Basistest deckt die Fehlerwirkungen sowohl in einzelnen Komponenten als auch im gesamten System auf. Abhängig von der Auslieferung des Entwicklungspartners, werden einzelne Komponenten der Software - ergänzt um Platzhalter - oder das komplette System inkl. Schnittstellen getestet.

Bei erfolgreich bestandenem Basistest oder bei einem Minor-Update (Datenaktualisierung aufgrund neuer Produkte, Preise, Steuern, Wechselkurse) fängt die umfassende Aufbereitung der Testversion an, welche zu Beginn automatische Tests beinhaltet und mit der Auslieferung der Testversion endet. Bei der Versionserstellung der Testversion wird ebenfalls die Datenlieferung vom Vertriebsdatenmanagement des Kunden aufbereitet und auf fehlerhafte Werte und Produktdaten getestet. Dabei wird der Datenstand zuerst manuell, anschließend automatisiert getestet. Sollten große Probleme im Datenstand aufgedeckt werden, werden diese über ein Ticketsystem dem Kunden umgehend mitgeteilt. So haben die Verantwortlichen für die Produktdaten auf der Kundenseite bereits während des ersten Tests die Möglichkeit, die Fehlerzustände der ersten Datenlieferung zu beheben.

Anschließend wird der Datenstand gemäß dem „Vier-Augen-Prinzip“ von einem unabhängigen Tester nochmals getestet. Das Testen durch zwei unterschiedliche Personen sichert die Unabhängigkeit der Testfallbewertung und verringert die Wahrscheinlichkeit eines falsch verstandenen Testfalls, womit es zum Erfüllen der Qualitätsziele beiträgt. Nach dem ersten Testdurchlauf wird die Testversion dem Vertriebsinnendienst des Kunden zum fachlichen Test übergeben. Hierbei wird gemäß der Anforderungsspezifikation

eine anwendernahe Sicht durch neue Testfälle simuliert und nach Abschluss mitsamt Feedback wieder an das Dienstleistungsunternehmen übergeben.

Nach dem fachlichen Kundentest wird, um zeitnahe Datenkorrekturen ergänzt, ein finaler Test gestartet. Dieser dauert ebenfalls vier Wochen und beinhaltet die gleichen Testfälle wie der erste Testlauf. Am Ende des zweiten Testlaufs übernimmt die Versionserstellung das Release der Software und stellt die Software für die Zielmärkte des Kunden bereit. Abschließend werden die im Projekt entstandenen Probleme und deren Lösungen in einem Review-Meeting zusammengefasst. Das Review-Meeting wird inkl. Protokollführung durchgeführt, um die Testfälle anzupassen und die Kernkompetenz im Managed Testing zu schärfen. So werden im Review-Meeting wenig effektive Testfälle angepasst oder aussortiert und der Vorgang im Sinne von „Lessons Learned“ festgehalten. Im Rahmen der Versionserstellung entstandene Probleme werden in einem umfassenderen Abschluss-Meeting diskutiert, wobei der Schwerpunkt nicht beim Testmanagement liegt, sondern alle Vorkommnisse des letzten Projekts besprochen werden. Zusätzlich werden über tägliche Meetings während der Projektlaufzeit - moderiert vom Teamleiter - der aktuelle Stand einzelner Versionsersteller ausgetauscht und mögliche Probleme diskutiert. Zusätzlich werden vom Kunden fünf bis sechs Scope-Meetings in Anwesenheit aller verantwortlichen Fachbereiche veranstaltet, um die Erreichung der qualitativen oder zeitlichen Ziele zu besprechen. An den Scope-Meetings nimmt der Teamleiter der Versionserstellung regelmäßig teil.

## 2.2 Wichtige Rollen während des Projektablaufs

Neben dem Versionsersteller sind im Projekt zwei weitere Personen von besonderer Bedeutung – der Testmanager und der Teamleiter bzw. der Projektleiter. Der Testmanager nimmt eine unterstützende Rolle ein und agiert während der Versionserstellung als Vermittler bei Problemen zum Testfallverständnis oder zu Modalitäten rund um eine Software-Version oder zu den Datenständen. Nach dem Abschluss des ersten Testdurchlaufs werden Berichte aus der Testmanagement-Software von Versionserstellern exportiert und dem Testmanager zur Auswertung weitergereicht. Am Ende des zweiten Testlaufs leitet der Testmanager das Review Meeting, um die Testfälle zu überarbeiten und die Änderungswünsche der Versionsersteller aus dem „Lessons-Learned“ zu priorisieren und bis zum nächsten Projekt zu implementieren. Damit trägt der Testmanager zur Qualitätssicherung bei, um wenig effektive Testfälle, welche keine Fehlerzustände aufdecken oder den Versionsersteller verunsichern (falsches Verständnis), zu eliminieren. Insgesamt agiert der Testmanager als fachlicher Ansprechpartner und Facilitator in Bezug auf Testprozesse. Er hat keine organisatorisch schützende Funktion über andere Tester während des Projekts und leitet auch keine Daily Meetings. Diese Aufgaben werden vom Teamleiter erledigt.

Der Teamleiter hat hingegen die Befugnis, Mitarbeiter vor anderen Aufgaben abzusichern, da die Mitarbeiter auch in anderen Projekten eingesetzt werden können. Zusätzlich beruft der Teamleiter die täglichen Meetings ein und leitet diese. Außerdem

werden vom Teamleiter zur Maximierung des Produktwerts die Anforderungen an die Versionserstellung durch Teilnahme an Kundenterminen priorisiert. Der Teamleiter pflegt ebenfalls die Kundenanforderungen und verantwortet die Aufnahme der Produktmerkmale. Da die Versionserstellung nach dem V-Modell am Ende des Gesamtprojekts anzusiedeln ist, hat der Teamleiter keinen Einfluss auf die Priorisierung der Kernfunktionalitäten der Software. Diese werden zwischen dem Kunden und dem Entwicklungsunternehmen definiert, womit die Priorisierung durch den Teamleiter in der Versionserstellung sich nur auf den Software-Test und die Datenimplementierung bezieht.

### **2.3 Tools zur Unterstützung des iterativen Vorgehens**

Während der Projektlaufzeit zum Test der PoS-Software werden diverse Tools eingesetzt. So werden fachliche Verbesserungen in gemeinsam gepflegten Excel-Dokumenten aufgenommen und anschließend werden die Ergebnisse per E-Mail, kollaboratives Intranet oder in Teammeetings kommuniziert. Die Kick-Offs, die ständig verbesserten Testfälle, die Kommunikation mit dem Kunden und die Meetings während der Versionserstellung haben relativ feste Zeiten und führen normalerweise zu keinen Abweichungen. Im Falle einer längeren Krankheit kann ein anderer Versionsersteller die Vertretung übernehmen. Dazu trägt ebenfalls ein Wiki bei, welches Wissen rund um Testprozesse beinhaltet und Wissensmonopole verhindert. Über Opensource Ticketsysteme und Gruppen-E-Mail-Konten wird ein enger Kontakt zum Kunden und zum Softwarelieferanten gepflegt, um Projektverzögerungen minimal zu halten. Da die Prozesse im Projekt eine bestimmte sequentielle Abfolge haben, werden GANTT-Diagramme zur Verfolgung des Fortschritts verwendet. Zur Durchführung von Kick-Off Veranstaltungen werden Telefonkonferenzen mit Bildschirmübertragung durchgeführt, wodurch kollaborative Arbeit ermöglicht wird. Für die iterative Abstimmung zwischen Versionserstellern und dem Projektleiter wird im Intranet eine Anwendung zur Erfassung des Fortschritts und der Zeitangaben in Tabellenform gepflegt. Die Notwendigkeit eines für Scrum empfohlenen Burndown-Charts [Ha10] besteht hingegen nicht, da die Projektverzögerungen fast ausschließlich aufgrund der sequentiellen Arbeitspakete von vorgelagerten Stellen (Softwarelieferant oder Kunde) des Konsortiumprojekts verursacht werden und ein Burndown-Chart keine Lösung schafft, sondern nur bekanntes aufzeigt. Da das Projektvorgehen bei den Partnern aus der Sicht des Dienstleistungsunternehmens eine Black-Box ist, macht eine unternehmensübergreifende Kooperation den Einsatz dieses agilen Tools überflüssig. Klassische Scrum-Dashboards werden auch nicht eingesetzt, aber eine historisch entstandene Intranet-Anwendung übernimmt die Funktionen des Scrum-Dashboards.

### 3. Anwendung auf die Versionserstellung

Der zuvor beschriebene Projektablauf wird nun auf die Merkmale bekannter Vorgehensmodelle überprüft.

<b>Klassisch Vorgehensmodelle</b>	<b>Agile Vorgehensmodelle</b>
Wasserfall-Modell	Scrum
V-Modell	Extreme Programming
Spiral-Modell	
<b>Hybride Vorgehensmodelle</b>	

Abb. 4: Übersicht der klassischen und agilen Vorgehensmodelle

Der vom Kunden vorgegebene Release-Plan stellt für die Versionserstellung gleichfalls die Anzahl von Sprints fest und ist deshalb mit Scrum-Flows vergleichbar [Ha10]. Der Kick-Off mit Kundenbeteiligung ist typisch für den Ausgangspunkt für Projekte bei dem Dienstleistungsunternehmen und markiert zugleich den Anfang in agilen Vorgehensformen wie Scrum. Von Anfang an werden im Rahmen des Kick-Offs die Anforderungen festgehalten, um alle Beteiligten des Kick-Offs bzgl. der Projektanforderungen auf einen einheitlichen Stand zu bringen. Die festgehaltenen Anforderungen ähneln einem Product Backlog [Sc10]. Die Beteiligung des marktspezifischen Vertriebsinnendienstes, sowie der Produktdatendokumentation auf der Kundenseite und die Teilnahme des marktspezifischen Versionserstellers sowie des Teamleiters als Repräsentanten auf der Seite des Dienstleisters, markiert den Kick-Off als eine kollaborative Schnittstelle am Projektanfang. Deshalb ist der Kick-Off ein agiles Merkmal zur Planung mit Kundenbeteiligung von Scrum. Einerseits verhindert es mangelndes „Business Involvement“ [WV13], andererseits hilft es den Planungsaufwand während des iterativen Vorgehens im Laufe des Projekts zu reduzieren [SP15].

Nach dem Kick-Off und erfolgreich durchgeführten Basistest fängt aus der agilen Projektmanagementsicht der erste Sprint an. Für die Versionserstellung bedeutet der erste Sprint einen vollständigen erstmaligen Test einer Version, welcher die Vorbereitung der Version, die Aufbereitung gelieferter Daten und den manuellen Test der Version und den automatisierten Test beinhaltet. Diese Prozesse laufen sequentiell ab und die Dauer des ersten Versionserstellungslaufs beträgt vier Wochen, womit es genau der Sprint-Laufzeit im Scrum entspricht [Me14]. Zudem stellt die Zeitangabe ebenfalls eine Time-Box dar. Während der Sprintdauer werden im Format des Daily-Scrum Meetings durchgeführt (der aktuelle Stand einzelner Versionsersteller untereinander ausgetauscht und mögliche Probleme oder Fehler diskutiert [Sc10] [KK14]). Allerdings werden die Scrum-Meetings aufgrund der kleinen Projektgröße nicht täglich durchgeführt. Außerdem können aufgrund von kurzen Kommunikationswegen innerhalb der Versionserstellung kleine Probleme informell unter Projektmitgliedern ausgetauscht werden. Werden große Datenfehler aufgedeckt, werden sie über ein Ticketsystem dem Kunden direkt mitgeteilt. Damit haben der Kunde oder der Softwarelieferant während des ersten Sprints die Möglichkeit, die Fehlerzustände zu beheben – ganz im Sinne der Flexibilität von Scrum [Sc10] und

des Feedback-Wertes nach Extreme Programming [He10]. Da die Kommunikation mit dem Kunden mittels eines modifizierbaren Opensource-Tools erfolgt, ist die technische Infrastruktur für Bugtracking und unternehmensübergreifende Kollaboration vorhanden. Die Möglichkeiten des Tools zur unternehmensübergreifenden Kollaboration werden jedoch durch die Compliance-Richtlinien des Kunden unterbunden, da Projektmitarbeiter des Dienstleisters die Tickets nicht an die Projektteilnehmer auf der Kundenseite zuweisen oder namentlich adressieren dürfen. Damit wird die These von van Waardenburg und van Vliet anhand der Versionserstellung als Beispiel gestützt, wonach agile Methoden im Einsatz in großen Unternehmen (Kunde) durch organisatorische Strukturen gebremst werden und dadurch zwangsläufig hybride Vorgehensmodelle entstehen, da es schwierig ist im Projekt agile Tools anzuwenden. [WV13]. Nach Sandhaus, Berg und Knot hilft zwar ein Vertrauensaufbau beim Überwinden von bürokratischen Hürden [SBK14], dennoch wurde dieser Grad des Vertrauens in der Versionserstellung entweder noch nicht erreicht, oder der erreichte Grad genügt als Bottom-Up Wandel nicht, um konzernweite Kommunikationsregelungen mit Externen beim Kunden zu ändern.

Bisher aufgeführte Risikosteuerungsmittel basieren ausschließlich auf Kommunikation und entsprechen der Aussage von Martakis und Daneva, wonach in kleinen Unternehmen die Kommunikation zur Risikosteuerung vorherrscht [MD13].

Die Einteilung in Arbeitspakete innerhalb des Sprints mit der Time-Box von vier Wochen ist ein agiles Scrummerkmal [KK14]. Das Vier-Augen-Prinzip ist ebenfalls ein Grundprinzip des agilen Extreme Programming Modells [Me14], welches beim Dienstleistungsunternehmen für den Softwaretest adaptiert wurde. Es verhindert den alleinigen Wissensaufbau eines einzelnen Versionserstellers, der zum Beispiel im Falle einer längeren Krankheit oder des Ausscheidens aus dem Unternehmen verloren geht und entspricht dem agilen Prinzip des Pair Programming. [SBK14] Zusätzlich beugt das Wiki im Intranet der Bildung von Wissensmonopolen vor.

Der Test durch den Vertriebsinnendienst des Kunden spiegelt das Real Customer Involvement (RCI) aus dem Extreme Programming wieder [SBK14]. Da der Test am Ende des ersten Sprints erfolgt und eine aktive Kundenintegration ermöglicht, ist es unübersehbar ein iteratives Merkmal einer agilen Vorgehensweise im Projektmanagement. Insgesamt besteht in der Versionserstellung, entsprechend iterativen und agilen Vorgehensmodellen, eine zeitnahe Stakeholder-Kommunikation mit Kunden und Softwarelieferanten.

Was die oben beschriebenen Positionen des Testmanagers und des Team-/Projektleiters angeht, so übernehmen sie nur zum Teil die Scrumrollen aus der Theorie. Die Rolle des Projektleiters ist teilweise mit der Rolle des agilen Scrummasters vergleichbar, teilweise trägt sie aber klassische und hierarchische Züge eines klassischen Projektmanagers. Die Rolle des Scrummasters [He10] lässt sich aufgrund der Organisationsstruktur im Unternehmen nicht vollständig anwenden und die Aufgaben werden zwischen dem Teamleiter und dem Testmanager aufgeteilt. Ein Grund hierfür ist die personelle Situation, die den Scrummaster nicht von operativen Aufgaben befreien kann – so hat der Testmanager bspw. selbst eine Version aufzubereiten, was wiederum zu Interessenskonflikten führen

kann. Es gibt durchaus Tailoring von Scrumrollen im Praxiseinsatz [Sc10]. In der Versionserstellung ist Tailoring aus Kostengründen notwendig (kleines Team) und der Vorgehensgeschwindigkeit, sowie der Compliance des Kunden geschuldet – da beispielsweise direkter Kontakt mit dem Kunden nur von Repräsentanten erfolgen kann und dafür Teamleiter-Befugnisse notwendig sind.

Das erwähnte Review-Meeting zur permanenten Qualitätsverbesserung des Softwaretests, sowie das Abschluss-Meeting zu allgemeinen Vorkommnissen des Projekts, haben Ähnlichkeiten mit der agilen Sprint-Retroperspektive, da sie in einem begrenzten Zeitrahmen Verbesserungsvorschläge zu den letzten zwei Sprints zum Gegenstand haben [Ha10]. Da am Ende der Versionserstellung im Rahmen des Abschluss-Meetings die Versionen der PoS-Software nicht vordergründig besprochen und getestet werden, entspricht es deshalb formell nicht dem Sprint-Review nach Scrum [Ha10].

#### 4. Fazit und Ausblick

Insgesamt verläuft die Versionserstellung als ein sequentielles Phasenmodell (Anlieferung, Aufbereitung, Test 1, Test 2, Auslieferung), das durch vielfältige agile Zusatzmerkmale angereichert wird, ganz i. S. v. Hüsselmann [Hü14]. Nach Sandhaus u. a. werden die Vorteile beider Vorgehen kombiniert und einzelne Iterationen entsprechen der Sprintdauer nach Scrum, angereichert um Daily Scrums zum iterativen Austausch, sowie Kick-Off zum Austausch mit dem Kunden [SBK14]. Für die Anwendung hybrider Vorgehensmodelle nach Lehman und Sharma, abhängig vom Reifegrad des Projekts, reicht die Dauer des Projekts nicht aus und ist daher im behandelten Praxis-Beispiel kaum anwendbar. Wenn das Gesamtprojekt aus der Gesamtprojektsicht des Kunden und nicht aus der Sicht vom Dienstleister betrachtet wird, so vollzieht sich das Softwarerelease nach klassischem V-Modell. Der Test fängt sequentiell nach dem Abschluss der Programmierarbeiten an und die Kommunikation mit dem Softwarelieferanten über ein Ticketsystem ist übertragbar auf das V-Modell. [SL05].

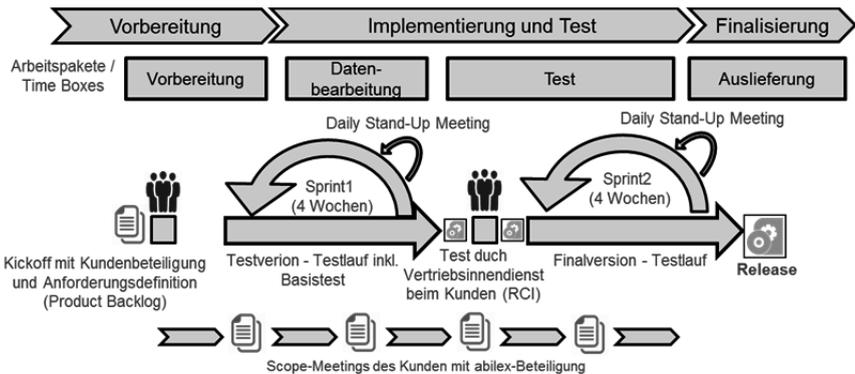


Abb. 5: Hybrides Vorgehen im Versionserstellungsprojekt bei abilex (Bottom-Up)

Der zweifache Testdurchlauf weist ebenfalls einige Merkmale des inkrementellen Spiral-Modells auf. Dennoch ist das Spiral-Modell mehr für komplexe und langwierige Entwicklungsprojekte gedacht. Die Versionserstellung ließe sich zwar in einer zweifachen Iteration darstellen, doch unterscheidet sich der Sinn der iterativen Durchläufe in dem Spiral-Modell von der Testwiederholung in der Versionserstellung. Während das Spiral-Modell eine beliebige Anzahl von Iterationen für ungewisse Risiken im Projektverkauf einhergehend mit einer Kostensteigerung zulässt, [Sc10] ist in der Versionserstellung ein zweifacher Durchlauf im Release-Plan vom Kunden vorgegeben, stellt für den Serviceanbieter für Managed Software Tests ein wichtiges Service-Merkmal dar und trägt zur Einhaltung von Projektzielen bei.

Bei einer ganzheitlichen Top-Down Betrachtung des Gesamtprojekts aus der Kundensicht, überwiegen eher klassische Merkmale, wobei abilex im sequentiellen Vorgehen des Kunden, nach dem klassischen V-Modell eingebunden ist und das Subprojekt der Versionserstellung agil steuert. Somit ist die Vorgehensweise im Gesamtprojekt auf jeden Fall hybrid – klassische Vorgehensweise im Gesamtprojekt und überwiegend agile Vorgehensweise in Teilprojekten.

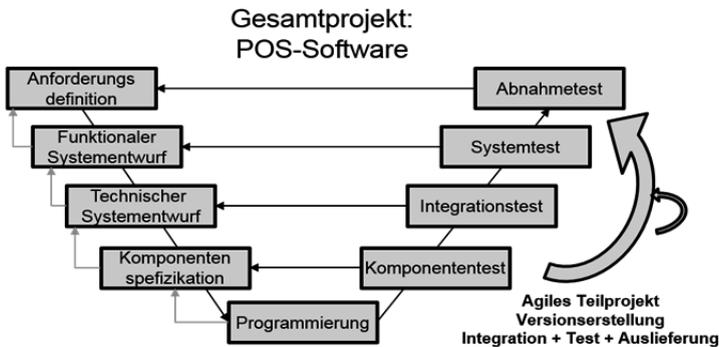


Abb. 6: Hybrides Vorgehen im Gesamtprojekt der PoS-Software (Top-Down)

Dies entspricht der Definition von hybriden Vorgehensmodellen [KK14] und dient als Beispiel für die gefundenen Thesen als Beispiel für den Einsatz von hybriden Methoden bei Großkonzernen. Diese These vertreten in ihren Artikeln Martakis und Daneva, van Waanderburg und van Vliet. Das Beispiel der Versionserstellung passt zu der These von Domann u. a., wonach Projekte mit mehreren Agenten durch klassisches Vorgehen nicht zu managen sind und den Einsatz agiler Tools erfordern. Dennoch können gewachsene Strukturen beim größten Konsortiumsteilnehmer agiles Vorgehen wie unternehmensübergreifende Kollaborationstools oder die reine Rolle des Scrummasters von den Anforderungen oder den gewachsenen Strukturen des Kunden ausbremsen. Im Fall der Versionserstellung wird der Grad der Agilität somit vom Kunden (Management des Gesamtprojekts) vorgegeben und vom Dienstleister als KMU (Management des Teilprojekts) adaptiert.

Die systematische Literaturanalyse und die Analyse des Projektablaufs in der Versionserstellung regen weitere Forschungstätigkeit zu folgenden Themen an. Die Literaturanalyse deckt unterschiedliche Einordnungen von hybriden Vorgehensmodellen auf. Deshalb ist es notwendig weitere Untersuchungen durchzuführen, um hybrides Projektmanagement systematisch einzuordnen, da hybrides Vorgehen von Unternehmen sowohl als Soll-Ziel, aber auch als vorübergehende Phase auf dem Weg zum agilen Vorgehen eingestuft wurde. Vor dem Hintergrund von unternehmensübergreifenden Projekten bietet es sich an, den Ausgangspunkt der Dynamik für hybrides oder agiles Vorgehen im Projektmanagement empirisch zu untersuchen, um einen Zusammenhang zwischen der Rolle eines Unternehmens in Kooperationen, der Unternehmenskultur und dem Vorgehen im Gesamtprojekt zu messen. Die Abhängigkeit zwischen der Partnerrolle und dem eingesetzten Vorgehensmodell bei Kunden-Partner-Beziehungen ist eine mögliche Forschungsrichtung. So wird im Praxisbeispiel der Versionserstellung das Vorgehensmodell bspw. nicht explizit vom Kunden vorgegeben, was aber in anderen Multistakeholder-Projekten anders sein kann. Dennoch hat abilex als Dienstleistungsunternehmen einige Elemente des agilen Vorgehens vom Kundenunternehmen übernommen. In diesem Zusammenhang müssen Einführungsstrategien für hybrides Projektmanagement erarbeitet werden, damit der Wandel von gewachsenen und traditionellen Organisationsstrukturen zwischen Kooperationspartnern gleichmäßig verläuft.

Die Ergebnisse der Literaturanalyse zur Notwendigkeit agiler Frameworks und deren Limitierung durch administrative Hürden, bietet die Grundlage für Untersuchung wie stark hybride Vorgehensmodelle von Tools unternehmensübergreifend unterstützt werden können und wie stark deren Einsatz von administrativen Hürden begrenzt wird. Manche Merkmale des agilen Vorgehens werden in der Versionserstellung nicht formal eingehalten, wie zum Beispiel die Meetings im Daily-Scrum-Format, die aufgrund der kleinen Projektgröße nicht täglich stattfinden. Daher bilden die Zusammenhänge zwischen Projektgröße und Skalierbarkeit agiler Tools weitere interessante Forschungsrichtungen. Anhand solcher Messungen können allgemeingültige Best Practices und Empfehlungen abgeleitet werden.

Resümierend sei angeführt, dass in unternehmensübergreifenden Projekten es zu Konflikten zwischen der Anwendung ausschließlich agiler Methoden und den gewachsenen Organisationsstrukturen, traditioneller Compliance oder den bestehenden vertraglichen Beziehungen kommen kann. Für solche Konstellationen sind hybride Vorgehensmodelle deshalb vorteilhafter als agile Methoden, da die traditionellen Komponenten zur Risikosteuerung und für einen reibungslosen reibungslosen Projektablauf bei Unternehmen unterschiedlicher Kulturen, Größen oder Branchen geeigneter sind.

## Literaturverzeichnis

- [Dj14] Domann, J. et.al.: An Agile Method for Multiagent Software Engineering, In: *Procedia Computer Science* 32, S. 928–934, 2014.
- [Da12] Daneva, M. et.al.: Agile requirements prioritization in large-scale outsourced system projects: An empirical study, In: *The Journal of Systems and Software* 86, S. 1333–1353, 2013.
- [GP08] GPM, Deutsche Gesellschaft für Projektmanagement e. V., [http://www.gpm-ip-ma.de/fileadmin/user\\_upload/Qualifizierung\\_Zertifizierung/Zertifikate\\_fuer\\_PM/National\\_Competence\\_Baseline\\_R09\\_NCB3\\_V05.pdf](http://www.gpm-ip-ma.de/fileadmin/user_upload/Qualifizierung_Zertifizierung/Zertifikate_fuer_PM/National_Competence_Baseline_R09_NCB3_V05.pdf), Stand: 25.05.2015.
- [Ha10] Hanse E.: *Agile Prozesse: Von XP über Scrum bis MAP*, Springer Verlag, Berlin, Heidelberg, 2010.
- [Hü14] Hüsselmann C., [http://www.scheer-management.com/wp-content/uploads/2014/07/Hybride-Vorgehensmodelle\\_Dr.-Claus-H%C3%BCsselmann.pdf](http://www.scheer-management.com/wp-content/uploads/2014/07/Hybride-Vorgehensmodelle_Dr.-Claus-H%C3%BCsselmann.pdf), Stand: 14.05.2015.
- [KK14] Kirchhof M., Kraft B.: Hybrides Projektmanagement. Agile und klassische Methoden im Projekt koordinieren. In: *Spotlight – Agiles Management - ein Überblick*, Berleb Media GmbH, Taufkirchen, S. 38–48, 2014.
- [LS11] Lehman T. J., Sharma A.: Software Development as a Service: Agile Experiences, In: *SRII Global Conference (SRII), 2011 Annual*, S. 749–758, 2011.
- [Ma13] Maximini D.: *Scrum - Einführung in der Unternehmenspraxis, Von starren Strukturen zu agilen Strukturen*, Springer Verlag, Berlin, Heidelberg, 2013.
- [MD13] Martakis A., Daneva M.: Handling Requirements Dependencies in Agile Projects: A Focus Group with Agile Software Development Practitioners, In: *Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference*, S. 1–11, 2013.
- [Me14] Meyerbröker P.: *Agiles Projektmanagement – eine Einführung*, In: *Spotlight – Agiles Management - ein Überblick*, Berleb Media GmbH, Taufkirchen, S. 13 – 22, 2014.
- [SBK14] Sandhaus G., Berg B., Knott P.: *Hybride Softwareentwicklung. Das Beste aus klassischen und agilen Methoden in einem Modell vereint.*, Springer Vieweg, Berlin, Heidelberg, 2014.
- [Sc10] Schatten A., Biffel S., Demolsky M., Gotischa-Franta E., Östreicher Th., Winkler D.: *Best Practice Software-Engineering, Eine praxisorientierte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen*, Spektrum Akademischer Verlag, Heidelberg, 2010.
- [SH14] Stettina C. J., Hörz J.: Agile portfolio management: An empirical perspective on the practice in use, In: *International Journal of Project Management* 33, S. 140–152, 2014.
- [SL05] Spillner A., Linz T.: *Basiswissen Softwaretest, Aus- und Weiterbildung zum Certified Tester, Foundation Level nach ISQTB Standard*. dpunkt.verlag, Bremen, 2005.

- [Sp14] Spundak M.: Mixed agile/traditional project management methodology – reality or illusion?, In: 27th IPMA World Congress, S. 939-948, 2014.
- [SP15] Serrador P., Pinto J. K.: Does Agile work? – A quantitative analysis of agile project success, In: International Journal of Project Management 33, S. 1040–1051, 2015.
- [SU11] Softwaretest-Umfrage 2011., <http://www.softwaretest-umfrage.de/2011/Vorgehensmodelle.htm>, Stand: 21.05.2015.
- [WV13] van Waardenburg G., van Vliet H.: When agile meets the enterprise, In: Information and Software Technology 55, S. 2154–2171, 2013.



# Hybrid ist Pflicht – mit Ultimate/Reliable Scrum und Critical Chain zu einer hochskalierbaren agilen Projektorganisation

Wolfram Müller

**Abstract:** Agile Methoden sind produkt- und teamorientierte Ansätze und stehen im Widerspruch zu Projekten mit hartem Terminen und Abhängigkeiten. Um die Vorteile aus beiden Welten zu nutzen muss man auch an beiden Welten Veränderungen vornehmen.

Das klassische Projektmanagement leidet an zu viel Work-in-Progress und starrer Terminplanung auf Arbeitspaketebene. Hier bietet Critical Chain Projektmanagement (CCPM) ein bewährtes Methodenset um Fluss und Agilität zu ermöglichen. Auf der anderen Seite können Agile Methoden keine Lieferumfänge zusichern und sind zu langsam. Auch hier existieren mit Ultimate/Reliable Scrum Agile Methoden der 3. Generation, die wiederum Fluss und Zuverlässigkeit ermöglichen. Damit ist der Weg frei – je nach Projektsituation, Projektphase, Teilprojekt oder Arbeitspaket genau die Vorgehensweise zu wählen, die mit geringstem Aufwand die Erfordernisse erfüllt.

In der Praxis wird das Portfoliomanagement, mit dem Konzept der „Virtual Drum“ aus der „Theory of Constraints“, massiv vereinfacht. Hier werden an einen „künstlichen/virtuellen“ Engpass die Projekte so gestaffelt, dass keine Ressource auf Dauer überlastet ist. Im Projektmanagement werden die Terminpläne durch Puffermanagement ersetzt. Zur Steuerung wird nur noch Fortschritt auf der kritischen Kette zu Pufferverbrauch herangezogen. Innerhalb von Arbeitspaketen oder Teilprojekten können nun auch agile Methoden zum Einsatz kommen. Die Teams erreichen mit Hilfe von Ultimate Scrum Boards den Zustand des „one piece flow“ und optimalen Durchsatz. Wie im Projekt wird die Zuverlässigkeit auch hier mit einem Puffer und Puffermanagement sichergestellt – namentlich Reliable Scrum. Es entsteht ein vollskalierbares agiles Projektmanagement Rahmenwerk.

Dabei handelt es sich nicht um Theorie sondern um bewährte Praxis mit enormen Potentialen.

**Keywords:** Ultimate Scrum, Reliable Scrum, Critical Chain Projektmanagement, CCPM, Theory of Constraints, TOC, Tame the Flow, Skalierbarkeit, agiles Projektmanagement

## 1 Einleitung

Wenn man das Ziel verfolgt eine hochskalierbare agile Projektumgebung zu entwickeln, dann gilt es auf der einen Seite die Projektwelt zu verbessern und auf der anderen Seite die agilen Methoden. Erst wenn beides auf der nächsten Stufe angelangt ist lässt sich beides integrieren und skalieren.

Im folgenden Beitrag wird daher zuerst aufgezeigt worin sich Projekte und agile Methoden unterscheiden. Erst wenn man diesen Unterschied kennt wird es möglich zu beurteilen wann und wo welche der Ansätze am sinnvollsten genutzt werden können.

Als nächstes wird gezeigt, wie man das Projektmanagement so entwickeln kann, dass es näher an die agile Welt reicht. Hierzu wird das Portfoliomanagement deutlich verschlankt so dass es ausgehend von einem virtuellen Engpass („Virtual Drum“) immer sichergestellt ist, dass der „Work-In-Progress (WIP)“ oder Bestand an Arbeit so eingestellt ist, dass kein Team überlastet ist. Hinzu kommt eine Projektampel, die basierend auf Fortschritt und Pufferverbrauch ein eindeutiges operatives Prioritätssignal generiert. Das Ergebnis ist eine in sich leichtgewichtige und agile Projektorganisation – namentlich Critical Chain Projektmanagement.

Nun kommen die agilen Methoden an die Reihe – auch hier gibt es Nachholbedarf. Als erstes müssen die agilen Methoden befähigt werden zuverlässig bestimmte Lieferergebnisse zu erzielen – dies geschieht durch ein Puffermanagement sehr ähnlich dem der Projekte. Dies wird als „Reliable Scrum“ bezeichnet. Darüber hinaus sind die aktuellen agilen Methoden in der Praxis zu langsam. Hierzu wird der Arbeitsfluss, wie in Lean Management, auf ein „One-Piece-Flow“ umgestellt und durch Konzepte aus der TOC ergänzt. Das Ergebnis ist optimaler Arbeitsfluss, minimaler Bestand und höchster Durchsatz. Die wird als „Ultimate Scrum“ bezeichnet.

Im letzten Teil wird das nun verbesserte Projektmanagement mit den optimierten agilen Methoden über die Arbeitspakete miteinander verknüpft. Es entsteht eine voll skalierbare agile Projektmanagement Organisation.

Dieser Beitrag kann nur an der Oberfläche kratzen und versuchen einen Überblick zu geben. Vertiefende Literatur und Quellen werden daher ausführlich genannt.

## **2 Unterschied zwischen Projekten und Produkten (Agile)**

Diese Abgrenzung hat weitreichende Auswirkung für das Projektmanagement und hilft vor allem die Unterschiede zwischen den agilen und den projektorientierten Ansätzen zu verstehen.

### **2.1 agile Methoden – Produktion?**

Von Produktion spricht man bei einem Anteil der Bearbeitungszeit („Touch Time“) zur Durchlaufzeit („Lead Time“) von kleiner als 10% [EG08]. Wenn man sich z.B. die Produktion von einer Schraube, einem Auto oder einem elektronischen Gerät betrachtet sieht das immer wie folgt aus: Das Produkt wird in vielen kleinen Teilen produziert. Viele dieser kleinen Teile bilden zusammen ein Produktionslos („Batch“). Aus Sicht des einzelnen Teils besteht der größte Teil der Durchlaufzeit aus "warten". Warten in einer

Palette, dann transportierten, entnehmen, kurz bearbeiten, wieder in eine Palette, transportieren u.s.w. - zwischen den einzelnen Bearbeitungsschritten entstehen relativ lange Wartezeiten.

Das was sich jetzt vielleicht negativ anhört hat aber große Vorteile. Durch die verhältnismäßig kurzen Bearbeitungszeiten kann man die einzelnen Zeiten recht genau schätzen. Wenn eine Bearbeitungszeit einmal überschritten wird, kann sie sich mit der Unterschreitung des nächsten ausgleichen und fällt nicht ins Gewicht. Wenn mehrere Teile parallel zu bearbeiten sind erhält man durch die langen Wartezeiten die Möglichkeit die Reihenfolge der Teile vor jedem Arbeitsschritt zu vertauschen und kann so jeden beliebigen Termin einhalten. Der Bestand („Work-In-Progress“ WIP) lässt sich leicht begrenzen, da er sehr überschaubar ist. Produktionssteuerungen sind extrem einfach, stabil und benötigen wenig Planungsoverhead.

Hierfür muss man aber einen Preis bezahlen: man muss alle Vorhaben in kleinste, relativ unabhängige Teilaufgaben zerlegen und man braucht irgendwo Warteschlangen (Puffer oder „Backlogs“) die die mittlere Durchlaufzeit erhöhen.

Wenn man sich nun die agilen Methoden betrachtet wird man feststellen, dass diese im Kern Produktionssteuerungen sind - kleine Teilaufgaben, unabhängige Teilaufgaben, Puffer auf den Taskboards und Backlog.

## 2.2 Projektmanagement

Von Projekten oder Projektmanagement spricht man wenn, der Anteil der Bearbeitungszeit zur Durchlaufzeit größer als 20% beträgt. Hierbei wird vor allem der kritische Pfad oder, wenn man die Ressourcenverfügbarkeit mit betrachtet, die kritische Kette (Critical Chain) [EG97] betrachtet. Das Ziel im Projektmanagement ist typischerweise das vereinbarte Ergebnis in der kürzest möglichen Zeit zu erbringen. Hierzu sind die echten Abhängigkeiten zwischen den Arbeitspaketen zu betrachten. Die Arbeitspakete sind hierbei optimal mit Ressourcen zu versorgen, so dass die Projektlaufzeit minimal wird.

Im Gegensatz zur Produktion strebt man nun aber einen Anteil der Bearbeitungszeit an der Durchlaufzeit von 100% an (typisch sind effektiv bis 80%). Hieraus ergibt sich, dass man folgende Eigenschaften von Projekten explizit managen muss: die Abhängigkeiten, die Streuungen in den Schätzungen und die operative Ressourcenzuordnung im Multiprojektmanagement. Projektmanagement ist deutlich komplexer im Vergleich zur Produktion und benötigt mehr Managementoverhead.

Bei Projekten muss man daher: ausreichend konzeptionieren und planen, so dass die Arbeitspakete und ihre Abhängigkeiten klar werden, muss die Aufwände schätzen um die Ressourcenzuordnung sicher zu stellen und die Streuungen am Projektende oder im Projektportfolio managen.

### 2.3 Projekte und Produktion (agile Methoden) im Vergleich

Die folgende Tabelle gibt eine Übersicht über die unterschiedlichen Systemklassen – Projekt und Produktion.

Kriterium	Projektwelt	Agile Welt (Produktion)
Touch/Lead-Time	nahe 100%	nahe oder kleiner 10%
Kopplung der Aufgaben	Starke Abhängigkeiten	Relativ unabhängig
Größe der Aufgaben	Stark unterschiedlich – großer Unterschied zwischen kleinstem und größtem	Relativ ähnlich und gleichmäßig klein,
Streuung/Risiko der Schätzungen	hohe Unsicherheiten	kleine Unsicherheiten
Menge der Aufgaben	Nur wenige kritische	Große Menge
Ziel	Lieferung eines definierten Umfangs zu einer definierten Zeit  Zeitlich begrenztes Vorhaben	Kontinuierliche Lieferung von optimalen Kundennutzen  Evolutionäres Entwickeln eines Produktes
Steuerungen	Klassisches Projektmanagement, Critical Chain Portfoliomanagement und Einzelprojektmanagement	div. Methodensets: u.a. eXtreme Programming, Scrum, Kanban, Reliable/Ultimate Scrum, Safe, Less, u.v.a.
Steuerungsaufwand	Hoch	Gering

Bei genauerer Betrachtung wird offensichtlich, dass in der Praxis beide „Welten“ immer gemischt auftreten.

In jedem Projekt sind per Definition die Charakteristika der Projektwelt vorhanden. Je nach Teilprojekte, Phasen oder Arbeitspaket können aber die Charakteristika der Produktionswelt (agile Methoden) überwiegen und somit von der einfacheren Steuerung profitieren können.

Umgekehrt weisen große agile „Vorhaben“ Charakteristika von Projekten auf. Vor allem in der Schlussphase werden Abhängigkeiten zu angrenzenden Fachgebieten stärker. In

großen Vorhaben sind Abhängigkeiten zwischen mehreren Teams zu managen und Ressourcenkonflikte mit dem Engpass aufzulösen. Damit steigt der Bedarf nach Methoden aus der Projektwelt.

### 3 Projektmanagement agilisieren – Critical Chain

Eine Projektmanagementorganisation besteht immer aus zwei Ebenen – dem Portfoliomanagement in dem die strategischen Entscheidungen (Prioritäten) gefällt werden und dem Projektmanagement darunter in dem die operative Ausführung im Vordergrund steht.

Um das bisherige Projektmanagement agiler zu gestalten werden folgende zwei Veränderungen vorgenommen.

Vorrangig wird Portfoliomanagement, mit dem Konzept der „Virtual Drum“ [UT14, HA10] aus der „Theory of Constraints“, massiv vereinfacht. Hier werden an einen „künstlichen/virtuellen“ Engpass die Projekte so gestaffelt, dass keine Ressource auf Dauer überlastet ist. Diese Situation ist bzgl. des Arbeitsflusses optimal.

Im Folgenden werden im Projektmanagement die Steuerung über Terminpläne durch Puffermanagement ersetzt [EG97, UT14]. Der Puffer wird hierzu den Arbeitspaketen entnommen und am Projektende zusammengefasst. Zur Steuerung wird nur noch Fortschritt auf der kritischen Kette zu Pufferverbrauch herangezogen.

Ausgangspunkte der von Dr. Eliyahu Goldratt entwickelten Methodik sind folgende Annahmen:

- Multitasking führt zu Mehraufwand, kostet Zeit und ist deshalb zu verhindern.
- In jeder Entwicklung gibt es einen Engpass und nur dort kann wirklich optimiert werden. Je reibungsloser es hier läuft, desto schneller der gesamte Prozess.
- Da Mitarbeiter zuverlässig sein wollen, bauen sie bei ihren Aufwandschätzungen oft bewusst oder unbewusst Puffer ein. Doch solche Reserven in einzelnen Aufgaben nützen dem Gesamtprojekt nichts. Werden sie nicht gebraucht, wird der Verantwortliche das nicht kundtun, um beim nächsten Projekt nicht unter Druck zu geraten. Dauert eine Aufgabe hingegen länger, wird der Verzug weitergegeben.

Ziel von Critical Chain ist es, schädliches Multitasking zu verhindern und die Reserven nutzbar zu machen. Die Aufgaben werden so organisiert, dass das Team/die Fähigkeit, deren Verfügbarkeit den größten Engpass bildet, konzentriert eine Aufgabe nach dem anderen abarbeiten kann.

Die Zeitschätzungen der Mitarbeiter werden um die geschätzten Puffer reduziert. Die gewonnene Zeit wird teilweise als Projektpuffer aggregiert und darf nach und nach verbraucht werden. In der Praxis entsteht ein regelrechter Wettbewerb darin, Puffer zu identifizieren und die Projektlaufzeit zu verkürzen.

Ist die Zahl der Projekte so hoch, dass schädliches Multitasking entstehen kann, wird der Bestand „Work in Progress (WIP)“ so weit reduziert, dass der Engpass gerade noch ausgelastet ist. Es entsteht eine Situation, in der alle Projekte genügend Ressourcen haben – was in klassischen Projektumgebungen meist nur für das Projekt mit der höchsten Priorität gilt. Erst wenn die kritischste Aufgabe fertig ist, wechselt der Mitarbeiter/das Team zur nächsten. Das bedeutet in der Regel, dass die Zahl der gleichzeitig aktiven Projekte sinkt, aber die Projektlaufzeiten deutlich kürzer werden.

Die Ressourcen werden nun operativ nach Fortschritt zu Pufferverbrauch der Projekte gesteuert. Das Projekt mit dem schlechtesten Verhältnis erhält die kritischen Ressourcen und die höchste Aufmerksamkeit.

Bei Projekten erweist sich oft die Integrationsphase (Integration, End-2-End-Tests, Bugfixing) als kritischer Engpass: D.h., das Unternehmen muss klar definieren, wie viele Integrationsprozesse es gleichzeitig verkraften kann, ohne dass die Projekte sich durch schädliches Multitasking gegenseitig behindern. An diesem Engpass werden die Projekte entsprechend getaktet. Die Integration hat höchste Aufmerksamkeit und Unterstützung des Managements inklusive.

Ergebnis: Die Projektlaufzeiten sinken, die Zahl der Projekte, die pro Monat oder Jahr fertig gestellt werden, steigt ebenso wie die Zuverlässigkeit – und das bei gleichen Ressourcen wie zuvor. Es kommt mehr Ruhe in die Projekte und der Stress für alle Beteiligten sinkt.

Durch die Umstellung der Steuerung auf Critical Chain reduziert sich die Durchlaufzeit initial um ca. 25% bis hin zu langfristig 50-70%. Im Gegenzug steigt der Durchsatz um entsprechende ca. 30% und darüber hinaus. Dies alles geschieht bei gleichem Ressourceneinsatz und höchster Zuverlässigkeit von ca. 90%.

Eine Übersicht über diese Erfolge findet sich auf folgenden Webseiten:

- [www.goldrattconsulting.com/Industries](http://www.goldrattconsulting.com/Industries)
- [speed4projects.net/critical-chain/Erfolgsgeschichten](http://speed4projects.net/critical-chain/Erfolgsgeschichten)

Die Art der Steuerung über den Projektpuffer und die enorme Steigerung der Geschwindigkeit sind ein deutlicher Schritt in Richtung Agilisierung.

## 4 Agile Methoden zuverlässig machen – Reliable/Ultimate Scrum

Agile Methoden haben in den letzten Jahren ihre Leistungsfähigkeit verdeutlicht. Die starke Fokussierung auf den Kunden und die Nutzung der Teamdynamik sind zwei Faktoren. Die einfache Steuerung und wenigstens temporäre Begrenzung des Work-In-Progress der zumeist größere Erfolgsfaktor.

Folgende Themen sind in der agilen Welt aber bisher nicht adressiert:

1. Zuverlässigkeit der Lieferung – keine agile Methode kann einen bestimmten Umfang zu einer bestimmten Zeit sicherstellen.
2. Alle agilen Methoden weisen hohe Bestände an Aufgaben entweder im Backlog oder als Work-in-Progress auf – daher sind alle agilen Methoden zwar scheinbar schneller – die Potentiale werden aber noch nicht ausgeschöpft.
3. Die Skalierbarkeit für große Vorhaben ist bisher nur unzureichend und unter großem Aufwand möglich.

Daher gilt es die guten Eigenschaften der agilen Methoden beizubehalten und die oben genannten Schwachpunkte zu beseitigen.

### 4.1 Zuverlässigkeit herstellen – „Reliable Scrum“ [ST13, WM12]

Ausgangspunkte sind das bekannte Scrum oder Kanban – diese werden ergänzt um zwei Elemente aus Critical Chain:

- Begrenzung des Work-in-Progress durch sachrichtiges ausbalancieren von Umfang, Ressourcen und Termin.
- Darstellung des Projektstatus als Fieberkurve mit Pufferverbrauch und Fortschritt (identisch zu Critical Chain).

Ziel von Reliable Scrum ist es dem Team eine realistische Erfolgswahrscheinlichkeit zu geben und dem Product Owner (Auftraggeber und Projektmanager) ein Werkzeug zur Steuerung des Backlogs (Work-In-Progress) zu geben, so dass zum zugesicherten Termin und die zugesicherten Funktion sicher geliefert werden.

Das Backlog wird vervollständigt und geschätzt. Die Abarbeitungsgeschwindigkeit wird ermittelt. Zusammen ergibt sich die Erfolgswahrscheinlichkeit. Das Backlog wird nun so eingestellt und verhandelt, dass die Erfolgswahrscheinlichkeit ausreichend hoch wird.

Durch diese realistische Erfolgswahrscheinlichkeit entsteht ein Puffer am Projektende. Mit diesem Puffer kann die aus Critical Chain bekannt Fieberkurve gezeichnet werden. Hier können die Stakeholder objektiv verfolgen ob das Release/Vorhaben auf Kurs ist und der „Product Owner“ (Auftraggeber/Projektmanager) kann sein Backlog immer so managen, dass die Erfolgswahrscheinlichkeit erhalten bleibt.

Wenn mehrere agile Projekte zusammen arbeiten müssen, ist oft schwer den Überblick über die einzelnen Teilstränge zu halten und die Abhängigkeiten zu managen. Die Fieberkurve kann man nun auch für ein Portfolio erstellen und so sicherstellen, dass der Großteil der Projekte „grün“ ist. In dem Moment werden Abhängigkeiten sicher eingehalten und Probleme früh entdeckt.

Ergebnis: der Auftrag wird deutlich schneller und besser geklärt. Das Team erhält dadurch einen klaren Leuchtturm, Fokus und Motivation. Der „Product Owner“ kann sein Backlog managen und den Status objektiv in Richtung Stakeholder kommunizieren. In Folge werden die „Releases“ (Projekte) deutlich schneller und zuverlässiger.

## 4.2 Ultimate Scrum [ST13, WM13]

Die Teams erreichen mit Hilfe von Ultimate Scrum Boards den Zustand des „one piece flow“ und optimalen Durchsatz, was einer Drum-Buffer-Rope (DBR) Produktionssteuerung, aus der Theory of Constraints [UT10], entspricht.

Um eine DBR als Projektsteuerung zu nutzen muss man, wie bei Scrum oder Kanban, das Projekt in kleinste Einheiten (Stories/Tasks) zerlegen. Zusätzlich muss man einen Arbeitsschritt/Ressource als Engpass („Drum“/Trommel) definieren und dann den Start von neuen Aufgaben genau an diesem Arbeitsschritt ausrichten. Hierzu wird vor/in diesem Arbeitsschritt ein Arbeitsvorrat („Buffer“/Puffer) installiert. Wenn dieser unter einen definierten Bestand sinkt werden neue Stories/Tasks gestartet – der Bestand dient als Signal („Rope“/Signalseil). Dies wird immer in Form eines Taskboards visualisiert.

Ein Ziel ist es die Sprints zu entfernen um damit die unnatürlichen Brüche am Ende der Sprints zu vermeiden und in einen kontinuierlichen Fluss zu kommen. Der Fluss ist wichtig um die Anzahl der offenen Stories und Tasks zu verringern und somit die Durchlaufzeit zu Verkürzen. Am Schluss natürlich auch der Durchsatz zu steigern.

Viele Dinge aus Scrum bleiben bestehen - nur die Steuerung wird angepasst. Eine Retrospektive alle 2-3 Wochen – ist immer noch sinnvoll. Es gibt weiterhin ein „Planning 1“ – durch „Reliable Scrum“ wird das Backlog aber in den ersten Sprints weitgehend komplett qualifiziert, so das Backlog vollständig priorisiert und geschätzt vorliegt und nur noch angepasst werden muss. Reviews werden natürlich auch gemacht - aber nicht in einem definierten Rhythmus, sondern alle 5-10 Stories, wenn das Lieferergebnis wirklich Wert aufweist. Ebenso die „Daylies“ bleiben erhalten.

Was ändert sich dann? Es gibt keine Sprints mehr! Es ist alles ein kontinuierlicher Fluss mit dem Ziel, so wenig wie möglich Stories und Tasks geöffnet haben.

Um das zu erreichen, wird der Prozess in zwei unabhängige Teile geteilt 1. das Schneiden der Stories in Tasks (auf der linken Seite des Taskpuffer) und 2. das Abarbeiten der Tasks (auf der rechten Seite).

Die Steuerung der linken Seite ist extrem einfach. Das Aufteilen von Stories in Task ist ja für eine Person nur wenige Stunden Aufwand – ungefähr so viel wie ein Task selbst. Daher wird das Schneiden von Stories ausschließlich durch die Menge der Tasks im Taskpuffer gesteuert. Wenn nur noch zwei Tasks übrig sind – wird einer der Entwickler die nächste Story aus dem Backlog nehmen und sie in Tasks aufbrechen. Es kann sein, dass die Alarmgrenze von 2 Task zu riskant ist. Sie sehen, dass - wenn ein Pufferloch auftritt – also keine Aufgabe im Taskpuffer mehr übrig ist. Wenn Pufferlöcher auftreten, muss man einfach die Alarmschwelle solange (langsam und schrittweise) erhöhen, bis keine Pufferlöcher mehr auftreten. Die Alarmschwelle sollte dabei nicht die Hälfte der Anzahl der beteiligten Entwickler überschreiten, ansonsten ist dies ein deutlicher Hinweis auf Prozessprobleme.

Das Monitoring geschieht über die, aus dem „Reliable Scrum“ bekannten Werkzeuge, also das klassische „Product Burndown Chart“ und die neue „Fieberkurve“. Diese Diagramme werden immer aktualisiert, sobald eine Story fertig gestellt wurde. Hierdurch entstehen viel mehr Messpunkte und noch feinere „Echtzeit“ Transparenz.

Und jetzt auf der rechten Seite? Normalerweise hat eine Drum-Buffer-Rope viele Prozessschritte (wie in Kanban). Hier haben wir aber einen kontinuierlichen Prozess mit zwei Schritten "Entwicklung" und "Review/Test". Das Besondere ist aber, dass es nur eine Art von Ressourcen gibt – Entwickler. Diese haben zwar Unterstützung durch Qualitätsmitarbeiter, die die Tests zu schreiben/durchführen - aber am Ende sind die Entwickler der begrenzende Faktor. Daher macht es keinen Sinn, zwischen den Prozessstufen zu unterscheiden. Beide werden durch die Verfügbarkeit der Entwickler eingeschränkt.

Die Drum-Buffer-Rope Steuerung besteht nun aus folgenden drei Teilen:

- Die Trommel - dies ist die begrenzende Ressourcen - in diesem Fall die Entwickler. Die Trommel ist wie der Herzschlag der Produktionskette. Sie gibt den Takt vor – nach ihr müssen sich alle richten.
- Dann benötigen wir einen Puffer (in der Regel vor der Trommel), aber in diesem Fall, wenn es nur einen begrenzenden Prozessschritt gibt ist der komplette Bestand (Anzahl der offenen Tasks) selbst der Puffer - spielt aber für die Steuerung keine Rolle ob ein angefangener Task im Bearbeitung ist oder in einem Puffer liegt. Keine dedizierten Puffer ist Letzt endlich sogar ideal.
- Und das Seil? Das ist die Signalleitung vom Puffer um neue Aufgaben zu starten. In diesem Fall ganz einfach - wenn eine Aufgabe abgeschlossen ist, dann darf eine neue Aufgabe gestartet werden.

Das Monitoring für die rechte Seite besteht aus einem "aggregierten Input-Output-Diagramm", oder manchmal auch "Flussdiagramm" oder „Continuous Flow Diagram (CFD)“ genannt. Das Ziel ist es, den Bestand (Differenz zwischen Ein- und Ausgangslinie) so niedrig wie möglich zu halten.

Dies kann auf sehr einfache Weise erreicht werden. Es werden keine neuen Aufgaben begonnen, bis die ersten "Pufferlöcher" entstehen. Wenn der erste Entwickler keine Task mehr hat, dann kann ein zusätzlicher Task gestartet werden und damit wird der Puffer um eins erhöht. Dies sollte aber eine Ausnahme sein und es müssen die Ursachen hierfür untersucht werden. Pufferlöcher sind voll von Informationen über Hindernisse oder verfahrenstechnische Probleme. Aber schließlich, wenn alles getan wurde und immer noch Pufferlöcher auftreten, dann muss man den Bestand erhöhen um den Durchsatz zu sichern.

Das Ergebnis ist ein kontinuierlicher Fluss ohne Prozessstörungen. Durch den minimalen WIP ist die Durchlaufzeit ebenfalls minimal und der Durchsatz erreicht das Optimum.

## **5 Das voll skalierbare agile Projektmanagement Rahmenwerk**

Mit Critical Chain Projektmanagement (CCPM) und den verbesserten agilen Methoden – Reliable und Ultimate Scrum sind die Einzelteile vorhanden. Diese müssen nur noch zu einem Ganzen integriert werden [MH14].

Das hier vorgestellte System geht daher einen großen Schritt weiter in Richtung eines Framework für vollintegriert Projekt- und Produktentwicklung. Beide Steuerungen (Projekt und agile Methoden) sind in der Praxis tatsächlich einfach zu kombinieren und zwar an der Stelle der operativen Priorität von Arbeitspaketen/Stories. Alle hier vorgestellten Elemente sind schon in unterschiedlichsten Kontexten erfolgreich im Einsatz - das Framework ist die Zusammenführung.

Das Framework soll kein Rückschritt in die tayloristische prozessuale Welt sein - sondern der Schlüssel, mit sehr wenig zentralen Informationen (strategische und operative Priorität), der zugrundeliegenden Organisation möglichst viel Freiheit für Selbstorganisation zu geben. Hierdurch können sich die positiven Wirkungen der teamorientierten agilen Welt entfalten und die Vorteile einer großen Organisation und Projekten genutzt werden.

Beschreibung:

- Es handelt sich um ein 3-Schichten-Framework. Jede Schicht hat eigene Charakteristika und Steuerungen.
- Das Portfolio-/Demandmanagement ist die oberste Schicht. Der Eingang wird über aber über Throughput Accounting Ansätze [JC04] gesteuert. Zum terminieren kommt entweder eine Stafflung an der „Virtaul Drum“ (für Projekt) oder eine simplified-Drum-Buffer-Rope (für die Kleinaufträge) zum Zuge. Damit werden Trade-Offs schnell transparent und das Board kann die strategischen Prioritäten setzen.

- Auf der untersten Eben finden sich die Teams, die „Epics“ (Teilprojekte) und Arbeitspakete („Stories“) umsetzen. Hier kommt Reliable/Ultimate Scrum zum Zuge. Reliable Scrum um „Epics“ sicher zu einem Termin liefern zu können – damit diese in Projekten funktionieren und Ultimate Scrum um einfacher Kleinaufträge und Stories aus „Epics“ mischen zu können und um schnellere Durchlaufzeiten zu erreichen und einen klaren Fokus des Teams auf Fluss zu ermöglichen.
- Der Clou ist die Mittelschicht. Die Mittelschicht dient dazu die „operative Priorität“ der Stories in den Backlogs zu liefern – so dass jedes Team genau weiß in welcher Reihenfolge es die Stories ziehen muss um die im Portfoliomanagement genannten Termine erreichen zu können. Hier gibt es zwei bewährte Methoden a) Critical Chain Buffer Management – ergibt eine rot-gelb-grün Ampel und b) die sDBR-Ampel über die Restlaufzeit bis zum Liefertermin – auch wieder rot-gelb-grün mit allen Feinabstufungen. Das heißt man kann beides mischen. Wenn die Teams sich "einigermaßen" an die Ampel halten dann werden die Termine insgesamt gehalten.

Das Framework ist ein vollintegrierter Ansatz Projekt- & Produktionsteuerung der neuesten Generation - kombiniert mit den ganzen positiven Aspekten der agilen und Lean Methoden [MH14].

## 6 Praxisbeispiele

Im Rahmen dieses Beitrags ist es leider nicht möglich von allen Praxisbeispielen hier detailliert zu berichten. Daher beschränke ich mich hier auf einen kurzen Abriss, wo dieses Framework und in welcher Ausprägung im Einsatz ist.

Projekt	Situation	Lösung/Wirkung
Entwicklung einer App (Start-Up in Berlin)	Entwicklung einer innovativen App zum sicheren verteilten Abgleich von Kontaktdaten  Zwei Firmen, vier Projektbeteiligte  Projekt nach einem Jahr massiv verzögert und Beteiligte nur noch über Anwalt in Kontakt	(1) Reliable Scrum - Klärung des restlichen notwendigen Umfang und Abschätzung eines realistischen Termins  (2) Überwachung des Termin mit Fieberkurve  (3) Ultimate Scrum – optimaler Fluss, Vermeidung von Wartezeiten  ➔ Release wurde pünktlich geliefert, Kommunikation deutlich verbessert

<p>Entwicklung Wire Swiss App/Website (Start-Up von Janus Friis ehem. Skype Gründer)</p>	<p>Entwicklung einer Kommunikationsplattform speziell Android-App</p> <p>14 Entwickler, zwei Standorte, 1 Jahr Entwicklung, laufende Verzögerungen</p>	<p>(1) Ultimate Scrum +50% mehr Durchsatz in vier Wochen</p> <p>(2) Reliable Scrum Termintreue und Verkürzung der Projektlaufzeit</p> <p>➔ App wurde früher fertig als die restlichen Bestandteile und in höchster Qualität</p>
<p>Ablösung von Softwareplattformen (1&amp;1 Internet AG)</p>	<p>Drei Großprojekte zur Ablösung komplexer Altsoftwaresysteme (10 Jahre), jeweils ca. 20 Mitarbeiter, 4 Standorte, bis zu 25 Teams</p>	<p>(1) Reliable und z.T. Ultimate Scrum</p> <p>(2) z.T. ergänzt um Portfoliomanagement nach CCPM</p> <p>➔ alle drei Vorhaben pünktlich und in vollem Scope geliefert</p> <p>➔ durch Ultimate Scrum Konzept „Continuous Integration“ etabliert</p> <p>➔ höchste Agilität – neue Funktionen in 2-3 Tage implementierbar</p> <p>➔ massive Reduzierung der Entwicklungsaufwände für Folgeprojekte z.T. um Faktor weit über 100</p>
<p>Drei Mechatronische Produktentwicklungen (Internationale Unternehmen: Medizintechnik, Komponenten für Anlagenbau, Messsysteme)</p>	<p>Mechatronische Entwicklung, 250 bis 500 Entwickler und Konstrukteure, weltweite Standorte, weltweit bekannte Unternehmen, starker Marktdruck durch Globalisierung</p>	<p>(1) vorrangig Critical Chain Portfolio- und Projektmanagement</p> <p>➔ Steigerung der Taskerledigungsrate um ca. 50-70% in 4-6 Wochen</p> <p>(2) kombiniert Ultimate Scrum in Softwareentwicklung, Elektronik und Mechanik</p> <p>➔ Massive Verbesserung des Arbeitsfluss und der Kommunikation untereinander</p> <p>➔ höhere Motivation der Mitarbeiter</p>

<p>Embedded System Softwareentwicklung (Weltmarktführer im Automobilbereich)</p>	<p>Softwareentwicklung mit bis zu 4500 Entwicklern – weltweit verteilt</p> <p>Hochkomplexe Organisation im Spannungsfeld zwischen Kundenanforderungen und Plattformentwicklung.</p>	<p>(1) Ultimate Scrum in Engpassteams</p> <p>➔ +60 bis +70% Durchsatz in 2 Wochen</p> <p>➔ hohe Termintreue</p> <p>➔ Beruhigung des Arbeitsumfeldes der Mitarbeiter, Reduktion der Vorstandseskalationen um Faktoren</p> <p>(2) sDBR-Steuerung zur Synchronisation der Ablieferungen</p> <p>➔ höchste Zuverlässigkeit</p> <p>(3) nächster Schritt Critical Chain als Projektsteuerung in der Kundenorganisation</p>
--	---	---

Abgerundet wird das Framework in der Praxis durch eine konzentrierte Vorgehensweise zur Einführung basierend auf den Strategie & Taktik Bäumen [HA10] aus der „Theory of Constraints“ [UT10].

In diesen S&T-Bäumen ist die optimale Schritt-für-Schritt-Reihenfolge beschrieben, wie der Veränderungsprozess, unter Nutzung der Selbstorganisation, schnell und nachhaltig durchgeführt werden kann.

## 7 Fazit – Hybrid ist Pflicht!

Es gibt gar keine Alternative zu hybriden Ansätzen. Die unterschiedlichen Bedarfe von Projekten und Produkten (Agile) erfordern eine differenzierte Herangehensweise.

Durch Critical Chain (CCPM) wird die Projektorganisation schneller und agiler. Auf Seiten der agilen Methoden wird durch Reliable/Ultime Scrum (advanced Agile) die Zuverlässigkeit und Geschwindigkeit erhöht, so dass beides in Kombination eingesetzt werden kann.

In Kombination mit einem Veränderungsprozess auf Basis der Strategie & Taktikbäume der TOC sowie unter Nutzung der Selbstorganisation kann der notwendige Veränderungsprozess deutlich beschleunigt werden. Das Risiko wird massiv reduziert und die Nachhaltigkeit steigt.

In der Verschmelzung von CCPM, advanced Agile und Veränderung basierend auf Selbstorganisation, liegt in der Praxis ein großes Potential mit Produktivitätssteigerungen von 50% bis über 100% bei gleichzeitiger Fokussierung auf den Kundennutzen.

## Literaturverzeichnis

- [GC84] Eliyahu M. Goldratt und Jeff Cox “The Goal: A Process of Ongoing Improvement”, North River Press, 1984
- [Go97] Eliyahu M. Goldratt “Critical Chain”, Gower Publishing Ltd , 1997
- [Go08] E. Goldratt „Standing on the Shoulders of Giants: Production concepts versus production applications. The Hitachi Tool Engineering example”, Goldratt Consulting, 2008
- [GB10] Eliyahu M. Goldratt und Alan Barnard „Projects CO Strategy & Tactics Tree“, [www.harmonytoc.com](http://www.harmonytoc.com), 2010
- [Ca04] John A. Caspari “Management Dynamics: Merging Constraints Accounting to Drive Improvement”, John Wiley & Sons, September 2004
- [HMR14] Mike Hannan, Wolfram Müller und Hilbert Robinson “The CIO Guide to Break-through Portfolio Performance”, booklocker, 2014
- [TM13] Steve Tendon und Wolfram Müller “Hyper-Productive Knowledge Work Performance: The Tameflow Approach and Its Application to Scrum and Kanban”, J. Ross Publishing, 2013
- [Te07] Uwe Techt und Holger Lörz „Critical Chain: beschleunigen Sie Ihr Projektmanagement“, Haufe-Gruppe, 2007
- [Te10] Uwe Techt „Goldratt und die Theory of Constraints: Der Quantensprung im Management“, Editions La Colombe, 2010
- [Te14] Uwe Techt “Projects that Flow: mehr Projekte in kürzerer Zeit, die Geheimnisse erfolgreicher Projektunternehmen”, Ibidem-Verlag, 2014
- [WM12] Wolfram Müller “ Scrum + Critical Chain = Reliable Scrum: Das Beste aus zwei Welten kombinieren“, [ProjektMagazin.de](http://ProjektMagazin.de), Ausgabe 18/2012
- [WM13] Wolfram Müller „Schneller geht's nicht – Ultimate Scrum: Agiles Projektmanagement“, [ProjektMagazin.de](http://ProjektMagazin.de), Ausgabe 05/2013

# Bierdeckelskizzen - Scrum ist leicht aber nicht einfach

Alexander Krieg<sup>1</sup>

**Abstract:** Vor ungefähr zwanzig Jahren wurde Scrum entwickelt. Ein Grund dafür war das Fehlen eines Rahmenwerks für die Softwareentwicklung, das die nötige Flexibilität und Schnelligkeit besitzt. Scrum ist das bekannteste Framework aus der agilen Softwareentwicklung. Der Prozess kann auf einem Bierdeckel skizziert werden. Dennoch fällt es Unternehmen und Teams schwer Scrum einzuführen. Was ist das Leichte an Scrum und warum ist es dennoch nicht einfach? Diese Fragen werden im vorliegenden Aufsatz behandelt und mögliche Lösungsansätze dargestellt. Es wird aufgezeigt, dass die Schwierigkeit nicht am Scrum Prozess selbst liegt sondern vielmehr an der Veränderung der Arbeitskultur sowie der vorherrschenden Kommunikations-, Führungs- und Denkmuster, die mit der Einführung einhergehen.

**Keywords:** Scrum Team, Scrum Master, moderne Führung, agiles Coaching, hybride Vorgehen, Kulturwandel, agiles Projektmanagement, Führung als Dienstleistung, agiles Schätzen, agiles Risikomanagement, agile Transition, Continuous Integration, IT-Kanban.

## 1 Einleitung

Scrum basiert auf Ansätzen des 1986 veröffentlichten Artikels (vergl. [TN86]) von Hirotaka Takeuchi und Ikujiro Nonaka. Sie beschreiben dort, dass sich die Regeln der Produktentwicklung verändert haben. Es geht nicht mehr nur um hohe Qualität und geringe Kosten, es kommen noch Geschwindigkeit und Flexibilität hinzu.

Ungefähr zehn Jahren nach der Veröffentlichung des Artikels „The New New Product Development Game“ entwickelten Jeff Sutherland und Ken Schwaber „Scrum“. Es gibt in Scrum nur drei definierte Rollen (Product Owner, Scrum Master und Entwickler Team), vier unterschiedliche Meetings (Daily Standup, Sprintplanungsmeeting, Sprintreview und Retrospektive) und ein sich ständig wiederholender (iterativer) Prozess (vergl. [SS13]). Aus prozessualer Sicht klingt das sehr einfach. Dennoch haben Teams und Unternehmen häufig Schwierigkeiten bei der Einführung bzw. einer späteren Ausweitung von Agilität (vergl. [Le07]) in mehrere Teams bzw. in die gesamte Organisation. Jede Scrum Rolle ist mit einem entsprechenden Empowerment ausgestattet. Das bedeutet, dass die Rollen mit einer klar zugeordneten Verantwortung und Ermächtigung ausgestattet sind, um gewisse Themen durchzusetzen bzw. einzufordern, ohne dabei eine disziplinarische Rolle einzunehmen. Die Entwicklung neuer Vorgehensmodelle, Rahmenwerke und eine Flexibilisierung der vorhandenen Strategien werden immer notwendiger, um die zunehmenden Veränderungen von Projektzielen, Rahmenbedingungen und stetig wachsender Dynamik in den Griff zu bekommen. Die klassischen Projektma-

---

<sup>1</sup> Acando GmbH, Millerntorplatz 1, 20359 Hamburg, alexander.krieg@acando.de

nagementvorgehen sind nicht flexibel und leichtgewichtig genug, um adäquat auf die hohe Veränderungsfrequenz zur Projektlaufzeit zu reagieren. Viele Projekte in der Softwareentwicklung stellen ein Team immer wieder vor neue, bisher noch nicht da gewesene Herausforderungen. Diese erfordern eine große Erfahrung der Teams und ein hohes Maß an Flexibilität und Anpassungsvermögen an die Vorgehensweisen. Softwareentwicklung ist eine relativ junge Disziplin, grundlegende Standards müssen noch weiter zu einer größeren Reife entwickelt werden. Brücken, Häuser und Straßen werden seit mehreren Jahrhunderten gebaut, entsprechend groß sind die Erfahrung und die vorhandenen Standards.

Die Darstellungen in diesem Beitrag beruhen auf Beratungs- und Projekterfahrungen des Autors. Das Spektrum der Projekte reicht von IT-Projekten, mit denen agile Methoden im Unternehmen eingeführt wurden bis hin zu IT-Großprojekten, die nach unternehmenseigenen Vorgehensmodellen durchgeführt wurden. Die benannten Erfahrungswerte spiegeln sich jedoch auch in der angegebenen Literatur wieder.

Gewisse Eckpunkte, die ein Projekt definieren bleiben gleich, egal für welches Vorgehen man sich entscheidet. Man benötigt einen Projektauftrag, der in einer bestimmten Zeitspanne und einem zuerst grob definierten Scope, einer vereinbarten Qualität und mit einer bestimmten Menge an unterschiedlichsten Ressourcen und Methoden fertig gestellt werden soll. Scrum wirbt damit, leichtgewichtig zu sein. Dies ist ein zentrales Merkmal agiler Prozesse und Methoden. Das bedeutet, es wird zum Wohl des Produkts und der Kundenzufriedenheit auf ein Übergewicht in der Organisation, Verwaltung und Dokumentation verzichtet. Das mag aus klassischer Sicht eigenartig erscheinen, besonders wenn in Erwägung gezogen wird, dass weitere agile Maxime eine kontinuierliche Verbesserung (Continuous Improvement) und ein sehr hoher Qualitätsanspruch sind und als zentrale Kulturmerkmale verstanden werden (vergl. [SS13]). Es gibt Bereiche in denen es schwierig wird, diese Leichtgewichtigkeit durchgängig zu realisieren, denkt man z.B. an die Dokumentationsanforderungen in der Pharmaindustrie. Entwickelt sich ein Unternehmen in Richtung Agilität, was die Einführung agiler Methoden, Werte und Vorgehen bedeutet, ist Scrum ein Schritt auf dem Weg der agilen Transition.

Dieser Beitrag behandelt vier Themenbereiche, die beim Arbeiten mit Scrum bzw. einer Einführung von Scrum zu beachten sind. Es sind Bereiche, in denen sich moderne Rahmenwerke und Projektvorgehen von bisher bekannten klassischen Vorgehensmodellen wie z.B. das Wasserfallmodell oder dem Rational Unified Prozess (RUP) (vergl. [Ve00]) unterscheiden. Diese vier Themenbereiche sind:

1. Vorgehen und Prozessmanagement
2. Strukturen, Rollen, Kommunikation und Empowerment
3. Werte, Kultur und Führung
4. Projektmanagement

In den beiden folgenden Kapiteln wird aus zwei unterschiedlichen Blickwinkeln auf diese Themenbereiche geschaut. In Kapitel 2) ‚Scrum ist leicht‘, betrachtet der vorliegende Aufsatz die vier Themenbereiche daraufhin, welche Punkte davon leicht umzusetzen sind. Hingegen werden in Kapitel 3) ‚Scrum ist nicht einfach‘ die vier Themenbereiche dahingehend beleuchtet, welche der Punkte nicht einfach umgesetzt werden können. In Kapitel 4) ‚Lösungsansätze zur agilen Transition‘ sind Ansätze aufgezeigt, die eine Einführung von Scrum in den Organisationsalltag vereinfachen können. Im letzten Kapitel wird ein abschließendes Resümee gebildet.

## 2 Scrum ist leicht

Scrum ist ein Rahmenwerk (engl. Framework) für die Entwicklung von Software und kein Projektmanagementvorgehen.

Das wird gerne missverstanden. In Scrum werden **Vorgehen und Prozessmanagement** oder auch sehr vereinzelt Methoden aus dem Projektmanagement beschrieben (vergl. [PI07]). Dennoch gibt es bei Scrum weder Personalbeschaffung noch Risiko- oder Stakeholdermanagement. Hingegen sind Moderations- und Kommunikationsfähigkeiten wesentliche Kompetenzen für die Durchführung von Meetings (vergl. Tab. 1) bzw. Retrospektiven. Das wohl bekannteste Bild aus Scrum ist der Scrum Prozess (vergl. Tab. 1) selbst, der auf einen Bierdeckel passt. Dieses Bild skizziert nur den grundsätzlichen Ablauf des Scrumprozesses. Der Prozess ist nicht kompliziert und kann von jedem Team nach einer kurzen Eingewöhnungsphase umgesetzt werden. Er besteht aus einer einzigen Phase, die Sprint genannt wird. Weitere gibt es nicht. Ein Sprint hat immer die gleiche Länge, denselben Ablauf und als Sprintergebnis wird immer ein lauffähiges Softwareinkrement ausgeliefert. Der Sprint wird bis zum Projektende kontinuierlich in gleicher Länge wiederholt. Hinzu kommen vier Meetings und eine Dokumentationsform (vergl. Tab. 1) für die Anforderungen. Der Sprint und auch die Meetings haben sehr klar strukturierte Inhalte und Abläufe (vergl. [Wo12]). Als Dokumentationsform für das Anforderungsmanagement gibt es die User Story. In der User Story wird nur dokumentiert, was die Entwicklung und der Testbereich für den aktuellen bzw. den nächsten Sprint benötigen. Die Basisabläufe, Strukturen und Vorgehen des Sprints, der Meetings und der User-Stories sowie die damit verbundenen Philosophie können in der Standardliteratur (vergl. [SS13]) nachgelesen und im Rahmen einer Weiterbildung zum Scrum Master und Product Owner vertieft werden.

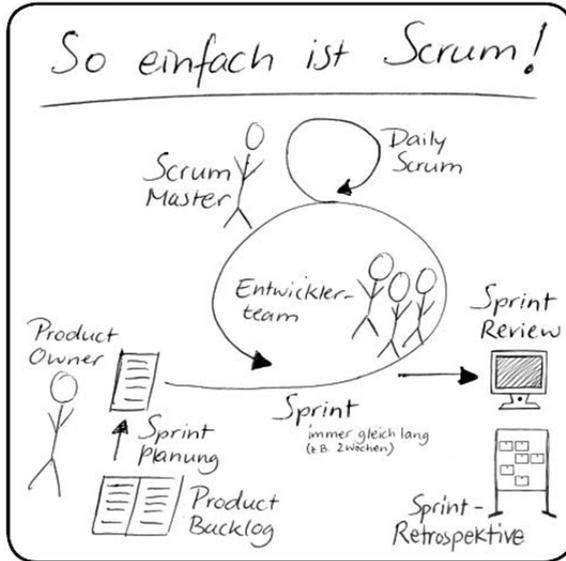


Abb. 1: Der Scrumprozess [Wo12]

**Strukturen, Rollen, Kommunikation und Empowerment** innerhalb des Scrum Teams sind klar definiert (vergl. [SS13]). Grundsätzlich sind alle Personen gleichberechtigt. Es gibt eine horizontale Struktur (vergl. Tab. 1), die die Aufgaben und die Verantwortung der Rollen (vergl. Tab. 1) definiert, aber keine disziplinarische Hierarchieebene oder Kommunikationsstruktur widerspiegelt. Die nicht vorhandene vertikale Hierarchie und die auf diverse Feedbackschleifen begründete Kommunikation erzeugt ein sehr offenes und von gegenseitigem Respekt begründetes Arbeits- und Kommunikationsklima. Dies lässt wenig Raum für Fehlinterpretationen und ist gleichzeitig die Basis für einen maximalen Wissenstransfer (vergl. Tab. 1). Das Scrum Team wird nach kurzer Zeit flexibel auf Veränderungen, die aus dem Anforderungsmanagement kommen, reagieren können. Es mag Teams oder Einzelpersonen geben, die mit derart viel Freiraum und der damit verbundenen Verantwortung und Selbstorganisation (vergl. Tab. 1) nicht sofort zurechtkommen. In der Regel entsteht jedoch nach kurzer Zeit eine intensive Gruppendynamik.

Die in Scrum gelebten **Werte, Kultur und Führung** unterstützen die Rollen, deren Aufgaben und Verantwortung. Die Werte sind im „Agile Manifesto“ (vergl. [Be01]) und „The Scrum Guide“ (vergl. [SS13]) festgehalten und werden von jedem agilen Team gelebt und auch geschützt. Ohne sie würde die agile Art zu arbeiten und zu kommunizieren nicht funktionieren. In einem Team, dessen Maxime Selbstorganisation, Lösungsorientierung, Vertrauen und Kooperation sind, gibt es nicht die große Notwendigkeit etwas „managen“ zu müssen. Das Team und jede einzelne Rolle hat eine hohe Verantwortung, höher als in klassischen Vorgehen. Der Führungsansatz (vergl. Tab. 1) des Product Owner und des Scrum Master liegen darin, die Arbeit des Scrum Teams zu fördern und zu unterstützen, damit es die hohen Erwartungen in jedem Sprint auch erreichen kann. Ent-

stehen Arbeitshindernisse, liegt es im Aufgabenbereich des Scrum Masters, diese schnellstmöglich für das Entwickler-Team zu beseitigen.

Einzelne Tätigkeiten, die sonst im klassischen **Projektmanagement** angesiedelt sind, gibt es ebenfalls. Arbeitspakete müssen priorisiert, geschätzt (vergl. [Pi07]) und auf die einzelnen Sprints eingeplant werden. Das geschieht im Sprintplanungsmeeting unter der Leitung des Product Owners, der die Kundenwünsche vertritt und den betriebswirtschaftlichen Erfolg des Projekts verantwortet. Nach jedem Sprint wird ein Sprintreview durchgeführt, neu priorisiert und entsprechend der aktuellen Velocity des Entwickler-Teams eingeplant. Auch die klare Gegenüberstellung zwischen geleisteter Arbeit und dem aktuellen Ergebnis wird durch den Product Owner erarbeitet und an die entsprechenden Stakeholder kommuniziert. Am Ende eines jeden Sprints steht ein lauffähiges Produktinkrement. Das ermöglicht eine sehr realistische Sicht auf den tatsächlichen Fertigstellungsgrad (vergl. Tab. 1). Die Aussage in klassischen Projekten, „das Projekt hat einen Fertigstellungsgrad in Prozent“, basiert auf der Fertigstellung einzelner Phasen, wie z.B. der Anforderungsphase. Diese Betrachtungsweise ist ungenauer und mit höherem Risiko zu bewerten als die Aussage über ein entwickeltes, getestetes und implementiertes Softwareinkrement. Die Tätigkeit des Product Owners im Projektmanagement konzentriert sich auf eine maximale Effizienz und Flexibilität auf Ebene der Arbeitspakete, nicht auf Verwaltungstätigkeiten und das pflegen von Plänen und Anforderungsdokumenten, die im aktuellen Projektstand noch keine Relevanz haben.

### 3 Scrum ist nicht einfach

Teams die es bisher gewohnt waren in einem klassischen Projektumfeld zu arbeiten und dadurch meist auch in Unternehmen mit vertikaler Organisationsstruktur, werden schon zu Beginn die Offenheit und die Effizienz schätzen, die sie im Scrum Alltag erleben. Allerdings wird ihnen das agile **Vorgehen und Prozessmanagement** etwas ungewohnt erscheinen. Schwierig wird es, wenn es das erste Scrum Team im Unternehmen ist und kein erfahrener agiler Coach zur Verfügung steht, der für das Team plant und auch die entsprechende Methodenkompetenz (vergl. Tab. 1) besitzt. Ein Agiler Coach ist keine explizite Scrum-Rolle. Erfahrungsgemäß empfiehlt es sich aber, die Einführung von Scrum durch einen erfahrenen Coach zu unterstützen. Dieser kann aus dem Unternehmen stammen oder extern hinzugezogen werden. Im Mittelpunkt steht seine langjährige praktische Erfahrung, die ggf. durch entsprechende Zertifikate nachweisbar sein sollte. Grundsätzlich benötigt ein Scrum Team bei den ersten Sprints ein Coaching, bis das Vorgehen, die Prozesse und die Kommunikation im Alltag vollständig gelebt werden. Der Einführungsprozess ist immer unterschiedlich und hängt von den gegebenen Projektrahmenparametern (vergl. Tab. 1) ab. Können die Teammitglieder vollständig im Scrum Team arbeiten oder werden sie über eine Matrixorganisation auch in anderen Projekten benötigt? Wie groß ist die Offenheit und Bereitschaft der Stakeholder gegenüber agilen Vorgehen? In welchem Maß ermöglicht die Organisation dem neuen Team Gestaltungsraum zur Eingewöhnung und Entfaltung? An die sehr konsequent moderier-

ten Meetings, wie z.B. dem Daily Scrum müssen sich viele erst gewöhnen. Man hat nur wenig Zeit, die Informationen der eigenen Arbeit den anderen strukturiert mitzuteilen und gleichzeitig alle Informationen der Teammitglieder für sich zu filtern und auch gezielt Fragen zu stellen. Zu Beginn wird der Scrum Master in den Meetings noch sehr viel moderieren müssen. Eine zentrale Aufgabe für den Scrum Master ist es, in jedem Sprint zuerst den vertikalen Durchstich (technische Realisierbarkeit) durch das vorliegende Produktinkrement zu schaffen. Erst danach werden Stories umgesetzt, die nicht zur zentralen technischen Achse des Produktinkrements gehören. Durch dieses Vorgehen werden Planungssicherheit und Beherrschbarkeit sowie ein zentraler Anteil am Risikomanagement unterstützt. Wird dieses Vorgehen in jedem Sprint angewandt, verspricht das die Verringerung des Gesamtrisikos und führt zu sehr frühem Erkennen von Fehlern in der fachlichen, technischen und architektonischen Planung und Konzeption. Dieses Vorgehen unterstützt insbesondere die Planungssicherheit von einander abhängigen Sprintzielen, wenn mehrere agile Teams in einem Projekt oder in hybriden Programmen (vergl. [Hi12]) koordiniert werden. Module müssen rechtzeitig fertig gestellt und integriert werden, um davon abhängige Softwareteile rechtzeitig zu bedienen.

Oft haben Organisationen die Herausforderung, für ein agiles Pilotprojekt die richtigen **Strukturen, Rollen, Kommunikation und Empowerment** zu definieren bzw. zu finden. Nicht einfach ist die Rollenbesetzung (vergl. Tab. 1) des Product Owners und des Scrum Masters. Viel zu häufig werden gerade diese beiden Positionen mit unerfahrenen Personen besetzt. Dadurch werden die ohnehin großen Herausforderungen bei der Einführung schnell zu unüberwindbaren Hürden. Oft muss das Pilotprojekt einen schnellen Erfolg vorweisen, um Akzeptanz zu finden. Scrum und andere agile Konzepte sind prädestiniert, schwierige Situationen durch hohe Effizienz und flexibles reagieren zu meistern. Das ist aber nur mit einem erfahrenen Scrum Team möglich, das auch das notwendige Empowerment (vergl. Tab. 1) besitzt. Ist das nicht gegeben, ist ein Scheitern, das dann gerne dem Vorgehen zugeschrieben wird, vorprogrammiert. Empowerment und Kommunikation sind entscheidende Punkte. In einigen Branchen werden IT und Fachbereich strikt getrennt. Auf Projektebene muss diese Trennung aber aufgehoben werden, um funktionsübergreifende Scrum Teams (vergl. Tab. 1) aufzubauen. Ein Scrum Master oder Product Owner, der nicht mit dem nötigen Empowerment ausgestattet ist oder nur wenig Erfahrung mit der sehr transparenten und offenen Art der Kommunikation agiler Teams hat, wird in der Umsetzung der erforderlichen Aufgaben nur wenig Erfolg haben. Dies gefährdet das gesamte Projekt. Oft tun sich klassische Organisationen schwer damit, einer neuen und für sie unklaren Rolle das nötige Empowerment zu geben und eine transparente Kommunikation zu fördern. Schwierig wird es auch, wenn ein bisheriger fachlicher Projektleiter als Scrum Master agiert und ein agiles Entwickler-Team anleitet oder als Product Owner ohne jegliche agile Erfahrung und Methodenkompetenz ein Scrum Team führen soll.

Gerne werden **Werte, Kultur und Führung** des agilen Vorgehens als nicht so wichtig erachtet. In zweitägigen Weiterbildungen zum Scrum Master oder Product Owner werden sie vermittelt, aber die Praxis zeigt, dass sich unerfahrene Product Owner und Scrum Master fast ausschließlich auf Methoden und Prozesse berufen, wie sie in der Fachlitera-

tur beschrieben werden. Damit handeln sie entgegen des ersten Kernsatzes des „Agile Manifesto - Menschen und Kommunikation sind wichtiger als Prozesse und Strukturen“ (vergl. [Be01]). Moderne Führung (vergl. Tab. 1) basiert auf Fördern, Vertrauen, Wertschätzung und auch los lassen. Moderne Führung sollte als Leistung im Dienst des Teams verstanden werden. Die durch agile Arbeiten entstehenden Möglichkeiten sind neben der Flexibilität und der Effizienz auch das Kreieren völlig neuer, kreativer Lösungen. Das benötigt Raum und Kommunikation - weniger Prozess und Struktur. In „Management Y“ (vergl. [Br14]) wird diese Art der Führung mit Douglas Mc. Gregors ‚Theorie X und Theory Y‘ gestützt. Das auf den Kopf gestellte Dreieck symbolisiert die neue Art des Führens. Die Mitarbeiter dienen nicht ausschließlich dem Management, sondern auch das Management dient den Mitarbeitern.

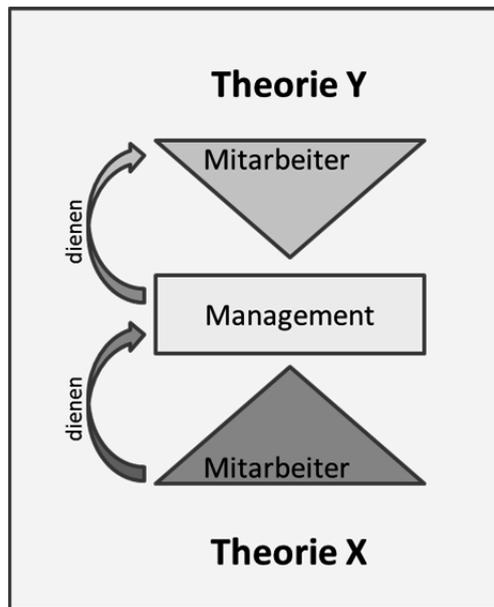


Abb. 2: Theorie X und Theorie Y [Br14]

In der gängigen Scrum Literatur gibt es wenig Aussagen zum **Projektmanagement** wie z.B. die im PMBOK (vergl. [Pm13]) beschriebenen Wissensgebiete Personalmanagement, Beschaffungsmanagement, Risikomanagement sowie Berichtswesen gegenüber Gremien und Lenkungsausschüssen. Zum Schätzen werden Methoden zur Sprintschätzung angeboten wie z.B. das Planning Poker. Eine Projektschätzung (vergl. Tab. 1) über das gesamte Projekt ist nicht vorgesehen. Geht es um ganze Programme oder die Einbettung in große Unternehmen und Konzerne, sind diese Themen unerlässlich, um ein oder mehrere Scrum Teams in die Organisation, in Multiprojekte oder Portfolien zu integrieren. Zu Beginn eines Projekts muss es eine agile Gesamtschätzung (vergl. [OW07]) aller Arbeitspakete geben. Der Erfolg hängt entscheidend von der Größe des Projekts und der

Erfahrung des Product Owners ab. Ein agiler Coach oder ein agiler Projektmanager ist an dieser Stelle einzubeziehen. Die Schätzung wird vom gesamten Scrum Team (bei z.B. Scrum of Scrums werden Vertreter aus jedem Scrum Team ausgewählt) unter der Anleitung des Product Owners bzw. des agilen Projektmanagers (vergl. [HK14]) durchgeführt. Es liegt nahe, dass diese Themen im Unternehmen von vorhandenen klassischen Projektleitern übernommen werden. Problematisch dabei ist, dass sich Scrum Teams und ihre Arbeitsweise nicht linear in eine klassische Projektplanung integrieren und steuern lassen. Beim Scrum of Scrums werden mehrere Scrum Teams gebildet, jedes Scrum Team ist für ein Produkt oder Teil eines Produkts verantwortlich und jedes Scrum Team hat auch einen Product Owner. Der Scrum Master ist in der Regel für mehrere Entwickler Teams parallel zuständig. Oft wird darüber noch ein Chief Product Owner installiert, der als zentrale Stelle alle Product Owner koordiniert. Diese Position könnte auch von einem agilen Projektleiter begleitet werden.

Oft sind unternehmensweite Vorgehensmodelle (vergl. Tab. 1) nicht auf das iterative Vorgehen agiler Teams angepasst. Sie beinhalten in der Regel Meilensteine oder Quality Gates, die zum Teil die gesamte Fachdokumentation sehr früh im Projekt erfordern. Klassische Projektleiter (vergl. Tab. 1) sind es auf Prozessebene noch nicht gewohnt in sehr kurzen Zyklen auf Anforderung, Entwicklung, Test und Integration zu steuern und zu planen. Oft fällt es schwer, die Idee des „managen“ los zu lassen und die Kontrolle für Abläufe und Kommunikation in die Teams abzugeben, zu fördern oder Hindernisse zu beseitigen (vergl. [BS08]). Das kann auch an Abteilungsgrenzen, einer strikten Trennung von IT und Fachbereich oder an Organisationshierarchien liegen. Wenn es eine Organisation gewohnt ist, große Projekte mit einem IT-Projektleiter und einem Fach-Projektleiter zu besetzen, müssten sie auf Teamebene dennoch versuchen interdisziplinäre Teams, bestehend aus Fach- und IT-Kräften, einzurichten. Ein Vorgehen nach Scrum macht sonst wenig Sinn. Projektsteuerkreise (vergl. Tab. 1) erwarten Kennzahlen, Projektstatus und Restaufwände. Diese basieren auf klassisch ablaufenden Phasen (Anforderungs-, Entwicklungs-, Test- und Integrationsphase). Ein Agiles Team hat wie in Kapitel 2) beschrieben eine veränderte Sicht auf diese Kennzahlen.

Folgende Tabelle gibt eine Übersicht über die in Kapitel zwei und drei behandelten Argumente innerhalb der vier Themenbereiche:

<b>Themenbereich</b>	<b>Scrum ist leicht</b>	<b>Scrum ist nicht einfach</b>
Vorgehen und Prozessmanagement	Scrum Prozess, Meetings, Dokumentationsform	Methodenkompetenz, Projektrahmenparameter
Strukturen, Rollen, Kommunikation und Empowerment	Rollen, horizontale Struktur, Wissenstransfer	Rollenbesetzung, Empowerment, funktionsübergreifende Scrum Teams
Werte, Kultur und Führung	Führungsansatz, Selbstorganisation	Moderne Führung
Projektmanagement	Fertigstellungsgrad	Projektschätzung, unternehmensweite Vorgehensmodelle, klassische Projektleiter, Projektsteuerkreis

Tab. 1: Wann ist Scrum leicht und wann nicht?

## 4 Lösungsansätze zur agilen Transition

Die Einführung von Scrum und agiler Methoden in ein Unternehmen sollte schrittweise durchgeführt werden. Dabei haben die Teams und Abteilungen Zeit, sich an die Methoden, Werte und Prozesse zu gewöhnen. Wird ein erstes agiles Pilotprojekt durchgeführt bzw. ein erstes Pilotteam von vier bis sieben Personen gebildet, ist im Unternehmen oft kein erfahrener Scrum Master oder Product Owner vorhanden. Die Rollen des Product Owners und des Scrum Masters sind die Schlüsselrollen in Scrum und müssen richtig besetzt werden. Interne fachliche Projektleiter können die Rolle des Product Owner übernehmen. Es kann dabei aber zu Diskussionen über Verantwortungsbereiche und Empowerment für Budget und Ressourcen kommen, was individuell zu klären ist. Etwas einfacher ist die Besetzung des Scrum Master. Es eignen sich erfahrene Entwickler und Softwarearchitekten - technisches Verständnis für die Prozesse der Softwareentwicklung sind sehr hilfreich. Zur Einführung bzw. Bildung eines agilen Teams ist es von Vorteil, dem Team einen agilen Coach zur Seite zu stellen. Ein agiler Coach kann bereits bei der Rollendefinition seine Erfahrungen einbringen und ein auf die vorhandenen agilen Kenntnisse zugeschnittenes Coaching aufbauen. Oft wird ein erstes agiles Team gebildet, aber die restlichen Stakeholder und Abteilungen sind sehr klassisch orientiert. Hier kann ein erfahrener Coach vermitteln und auch helfen, die richtigen Kommunikationsstrukturen aufzubauen. Die Unterstützung eines Coachs ist besonders zum Projektstart wichtig, wenn es um die Planung und Schätzung der Arbeitspakete und Ressourcen geht und das Team noch keine Sprinterfahrung hat. Die Positionen des Scrum Masters und des Product Owners benötigen zu Beginn ein stärkeres Coaching, das nach den ersten Sprints abnehmen wird. Die so gecoachten Rollen können ihr Wissen an andere Teams

weitergeben. So kann ein Wissenstransfer der agilen Teams beginnen. Je größer die Projekte sind die agil umgesetzt werden sollen, desto mehr Erfahrung und auch Coaching ist am Anfang notwendig. Zusätzlich müssen Rollen definiert bzw. umgestaltet werden, wie zum Beispiel ein Chief Product Owner oder ein agiler Projektleiter (vergl. [OW07]), der das klassische und das agile Vorgehen beherrscht. Er versteht es, die agilen Teams zu führen, die Gesamtprojektplanung und Koordination zu übernehmen und die erwarteten Zahlen Richtung klassischer Steuerkreis zu kommunizieren. Sollen Projekte mit mehr als zwei bis drei Scrum Teams durchgeführt werden, wird ein agiles Projektmanagement oder ein hybrides Programm zur Steuerung und Planung notwendig. Existiert ein unternehmensweites Vorgehensmodell, stellt sich die Frage, ob es agile Teams unterstützt oder evtl. eine Anpassung notwendig ist. Soll ein klassisches Projekt oder eine hierarchisch strukturierte Organisation mit agilen Methoden unterstützt werden, kann die Verwendung von IT-Kanban (vergl. [An12]) hilfreicher sein als Scrum. Ist es nicht möglich, funktionsübergreifende Teams - bestehend aus IT und Fachbereichen - aufzubauen, kann ebenfalls die Einführung agiler Visualisierungs- und Prozessverbesserungsmethoden aus dem IT-Kanban ein Lösungsansatz sein. Es hat den Vorteil, dass der Fokus auf dem Verbessern der bereits vorhandenen Prozesse und des Prozessflusses liegt und nicht auf der Veränderung von Strukturen, Rollen, Kommunikation und Vorgehen. IT-Kanban steuert und optimiert die sich im Fluss befindenden Arbeitspakete (Work in Progress ,WIP'). Eine weitere Möglichkeit ist es, in der Softwareentwicklung und den Projekten ein iteratives Vorgehen einzuführen. Der Einsatz von Kanban und die Entwicklung iterativer Vorgehen kann parallel ablaufen. Hat ein Team oder eine Organisation gelernt in Iterationen zu planen und zu koordinieren, ist der erste Schritt zu mehr Agilität bereits vollzogen und eine Entwicklung in Richtung Scrum kann die konsequente Folge sein.

## 5 Resümee

Der Aufsatz soll zeigen, dass die Herausforderung mit Scrum nicht in einem einfach zu skizzierenden Prozess liegt, sondern in der Veränderung der Arbeits-, Führungs- und Kommunikationskultur. Agile Vorgehen sind leichtgewichtig, dynamisch und flexibel. Oft beginnt die agile Transition in den Entwicklungsabteilungen und Projektebenen. Die Einführung und Integration in eine bestehende, durch hierarchische Strukturen definierte Organisation ist nicht einfach. Sie erfordert die Bereitschaft zu Beginn die richtigen Fragen zu stellen, ein klares Ziel auszugeben und die dafür notwendigen Veränderungen konsequent durchzuführen. In jedem Fall sollte die Einführung von Scrum mit erfahrenen Scrum Mastern, Product Ownern bzw. einem agilen Coach beschritten werden, um Misserfolge weitestgehendst zu vermeiden. Ein Blick in die Gegenwart der modernen Softwareentwicklung und IT-Infrastrukturen zeigt, dass in Unternehmen Begriffe wie Continuous Delivery, Continuous Integration und DevOps längst Einzug hielten. Um dies adäquat bedienen zu können, ist es von Vorteil wenn mindestens ein iteratives Vorgehen und Ansätze von agilen Teams eingeführt sind.

## Literaturverzeichnis

- [An12] Andersen, D.J.: Kanban: Evolutionäres Change Management für IT-Organisationen. dpunkt Verlag, 2012
- [Be01] Beck et al.: Manifesto for Agile Software Development. Siehe <http://agilemanifesto.org/>, 2001.
- [Br14] Brandes, U. et al.: Management Y. Campus, 2014.
- [BS08] Buhse, W.; Stamer, S.: Enterprise 2.0: Die Kunst, loszulassen. Rhombos, 2008
- [Hi12] Hilmer, S.: Hybride Vorgehensmodelle für ein unternehmensweit einheitliches, flexibles Projektmanagement. In (Linssen, O.; Kuhmann, M. Hrsg.): Qualitätsmanagement und Vorgehensmodelle, 19. Workshop der Fachgruppe Vorgehensmodelle (WI-VM) der Gesellschaft für Informatik e.V. (GI). Shaker Verlag, Aachen, S. 129-138, 2012.
- [HK14] Hilmer, S.; Krieg, A.: Standardisierung vs. Kultur: Klassisches und agiles Projektmanagement im Vergleich. In: (Engstler M.; Hanser, E.; Mikusz, M.; Georg Herzwurm Hrsg.): Vorgehensmodelle 2014. GI-Edition, Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, S. 65-76, 2014.
- [Le07] Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Addison-Wesley, 2007.
- [OW07] Oestereich, B.; Weiss, C.: APM - Agiles Projektmanagement: Erfolgreiches Timeboxing für IT-Projekte. dpunkt.verlag, 2007.
- [Pi07] Pichler, R.: Scrum - Agiles Projektmanagement erfolgreich einsetzen. dpunkt Verlag, 2007.
- [Pm13] Project Management Institute: A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide. Project Management Institute, 2013.
- [SS13] Sutherland, J.; Schwaber, K.: The Scrum Guide. Siehe: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>
- [TN86] Tekeuchi, H.; Nonaka I.: The new new product development game. Harvard Business Review, 1986.
- [Ve00] Versteegen, G.: Projektmanagement: mit dem Rational Unified Process. Xpert.press, 2000.
- [Wo12] Wolf, H.; von Solingen, R.; Rustenburg, E.: Die Kraft von Scrum. Addison-Wesley, 2012.



## Teil II

Eingeladene Beiträge der Session „Future Track“



# Nachhaltige Agile Transition: Symbiose von technischer und kultureller Agilität

Philipp Diebold<sup>1</sup>, Steffen Küpper<sup>2</sup>, Thomas Zehler<sup>3</sup>

**Abstract:** Agile Entwicklungsvorgehen stellen den Stand der Technik bei der Entwicklung von Software und Systemen in unterschiedlichen Domänen dar. Beim Einsatz agiler Methoden wie Scrum kommt es jedoch häufiger zu Problemen, da die Einführung derartiger Methoden häufig nur aus technischer Sicht betrachtet wird. Die Berücksichtigung eines, als Voraussetzung für die nachhaltige Etablierung, kulturellen Wandels der Organisation zur Agilität kommt häufig zu kurz. Unser Beitrag stellt deshalb die anzustrebende Symbiose zwischen technischer und kultureller Agilität in den Vordergrund. Wir erläutern, wie eine schrittweise Transformation hin zum richtigen Grad an Agilität aussehen kann, wenn er das Gleichgewicht von Technik und Kultur berücksichtigt.

**Keywords:** Agile Softwareentwicklung, technische Agilität, kulturelle Agilität, Transition.

## 1 Einleitung und Motivation

In der Softwareentwicklung ist die agile Entwicklung in den letzten 20 Jahren den Kinderschuhen entwachsen, hat sich sukzessive weiterentwickelt [Ab03] und ist vom Stand-der-Wissenschaft zum Stand-der-Technik gereift. Agile Vorgehensweisen sind besonders in ausgewählten Domänen, wie der Webentwicklung und bei bestimmte Projekttypen mit enger Zusammenarbeit mit dem Kunden, mittlerweile weit verbreitet, während ihrer Verbreitung in anderen Bereichen (z.B. Automotive) größere Vorbehalten gegenüberstehen. Dessen ungeachtet wird die agile Entwicklung in Unternehmen zumeist mit traditionellen Vorgehen kombiniert [BT04]. Dies wird durch eine Reihe aktueller Studien untermauert, die sich mit aktuell verwendeten Entwicklungsprozessen beschäftigen [VO15], [HK14].

Agile (Software-)Entwicklung lässt sich auf verschiedenen Ebenen betrachten. Wohl bekannt sind die **Agilen Methoden**, wie Scrum, Kanban, DSDM, XP, etc., die sich ihrerseits meist aus verschiedenen **Agilen Praktiken** zusammensetzen, z.B. PairProgramming, Daily StandUps, Sprint Planning, etc. Da sich die Umsetzung der Praktiken der Umwelt nach außengrößtenteils durch festlegen von Regeln, (Prozess-)Abläufen und Artefakten auszeichnet, insbesondere durch die Verwendung von technischen Lösungen, wird die Menge dieser Elemente im Folgenden als *technische Agilität* bezeichnet. Auf der anderen Seite spielen **Agile Prinzipien**, benannt im Agilen Manifest aber auch wei-

---

<sup>1</sup> Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, philipp.diebold@iese.fraunhofer.de

<sup>2</sup> TU Clausthal, IPSE, Julius-Albert-Straße 4, 38678 Clausthal-Zellerfeld, steffen.kuepper@tu-clausthal.de

<sup>3</sup> Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, thomas.zehler@iese.fraunhofer.de

ter wie Verantwortungsübernahme, aktive Anwender-Einbindung und positive Fehlerkultur eine wichtige Rolle. Sie beeinflussen die Kultur bzw. das Verhalten sowie die Entscheidungen einer Gruppe. Wahrnehmbar für Andere werden Prinzipien beispielsweise durch Rituale, Leitbilder oder das sichtbare Verhalten der Mitglieder einer Gruppe. Diese sind Ausdruck eines tieferen, kollektiven Gefühls für das Richtige. Das Gefühl selbst wird geprägt von Grundannahmen, die nicht mehr hinterfragt werden (vgl.: „Ebenen der Kultur“ von E. Schein [Sc04]). Im Folgenden wird diese als *kulturelle Agilität* bezeichnet. Gemeinsam ist **beiden**, dass sie eng mit den **Agilen Werten** (engl. Core Values) des Agilen Manifests [Be01] verbunden sind.

*Technische* und *kulturelle Agilität* sind insofern verknüpft, als dass die technische Agilität das Vorhandensein gewisser kultureller Werte im Unternehmen zur vollständigen Implementierung voraussetzt. Wir erachten es als sinnvoll, beide Aspekte stets zusammen zu betrachten. Wobei die technische Agilität in der Praxis nach unserem Dafürhalten zumeist die führende Rolle einnimmt.

## 2 Projekterfolg durch Agilität

Verschiedene Studien [Di15] haben gezeigt, dass bei der Einführung agiler Softwareentwicklung auf technischer Ebene durch Methoden wie Scrum, im Nachhinein häufig Änderungen bzw. Adaptionen notwendig sind, um die jeweilige Methode an den konkreten Unternehmenskontext anzupassen.

Für uns ist daher ein schrittweiser und wohlfundierter Prozess bei der Einführung agiler Vorgehensweisen in einem Unternehmen einem durch Nachjustieren zum Ziel führenden Big-bang-Vorgehen vorzuziehen [DZ14]. Dafür schlagen wir das folgende Vorgehen vor: (1) Ausgangspunkt der Überlegungen zur Einführung agiler Elemente im Unternehmen sind in der Regel auftretende Probleme mit dem gegenwärtigen Entwicklungsvorgehen. Für die angestrebte Prozessverbesserung lassen sich daraus ein oder mehrere Ziele ableiten, beispielsweise die Erhöhung der Effizienz oder der Flexibilität. (2) Diese Ziele beeinflussen die Adaption des Prozesses maßgeblich und müssen daher vorab identifiziert werden, ebenso wie die relevanten Stakeholder. (3) Auf dieser Basis und weiterer verschiedener unternehmensspezifischer Rahmenbedingungen, z. B. einzuhaltende Standards, Kundenabsprachen oder Dokumentationspflichten, ermöglicht es die Agile Potenzialanalyse [DZ14], eine Liste an geeigneten Agilen Praktiken als Ausgangspunkt für das weitere Vorgehen zu extrahieren. Deren Einführung in das Unternehmen muss anschließend durch das Unternehmen erfolgen.

Die Agile Potenzialanalyse impliziert, wie auch in der gegenwärtigen Praxis weit verbreitet [BT04], dass eine Transformation des Unternehmens nur für die eigene Vision der Agilität erforderlich ist. Damit die identifizierten Probleme und angestrebten Ziele adressiert werden. Dies kann auch bedeuten, dass nicht die Out-of-the-Box-Einführung einer existierenden agilen Methode wie Scrum die am besten geeignete Herangehensweise darstellt. Stattdessen kann die schrittweise Einführung einer unternehmensspezifischen Menge agiler Elemente sinnvoller sein. Wie sie sich unter anderem als Output der agilen Potentialanalyse darstellt.

### 3 Technische und kulturelle Aspekte bei der Agilen Transition

Wie in Kapitel 2 erwähnt erachten wir eine Schritt-für-Schritt Einführung von agilen Elementen im Unternehmen für sinnvoller als einen Big-bang mit anschließendem Nachjustieren. Dies erfordert eine aufeinander aufbauende Einführung einzelner agiler Praktiken aus einem Set geeigneter Praktiken, das seinerseits in Gänze die Erfüllung des definierten (Verbesserungs-) Ziels unterstützt. Bei einer auf technische Agilität fokussierenden Einführung von Praktiken ist darauf zu achten, dass die jeweilige kulturelle Agilität mitberücksichtigt wird und durch die Implementierung beeinflusst wird. Damit soll ein ideales Gleichgewicht zwischen beiden Aspekten der Agilität geschaffen werden.

Angelehnt an das S-Kurven-Konzept der Technologiereife [So08] erklärt sich das Vorgehen (Abbildung 1, blaue Linie) bei der Transition des Entwicklungsvorgehens durch den anfänglichen Fokus auf die *technische Agilität*, d.h. auf agile Praktiken, die einen geringen Anteil an kulturellem Wandel benötigen. In der Phase 1 Einführungsphase - wird durch die Konzentration auf kleinteilige Änderungen auf *technischer Ebene* leichter gelingen, den Entwicklungsprozess im Unternehmen zu verändern. Neben der technischen Agilität muss jedoch der *kulturellen Agilität* ebenso Beachtung geschenkt werden. Agile Praktiken und Methoden lassen sich nur dann sinnvoll einführen, wenn dafür gleichzeitig die zugrundeliegenden agilen Werte und Prinzipien in der Organisation verankern, d.h. eine unterstützende Kultur geschaffen wurde. Auf dem Fundament von Phase 1 lässt sich in der nachfolgenden Phase 2 Wachstumsphase – aufbauen. In der geht es darum, die *kulturelle Agilität* auszubauen. Am Sinnvollsten sollten nun agile Praktiken eingeführt werden, die einen großen Wandel der *kulturellen Agilität* erfordern. In dieser Phase liegt der Fokus dediziert auf der umfassenden Einführung und Verbreitung *kultureller Agilität*. In der dritten und abschließenden Phase - Reifephase - liegt der Schwerpunkt wieder eher auf der *technischen Ebene* und die Vermittlung zusätzlicher *kultureller Aspekte* tritt erneut in den Hintergrund.

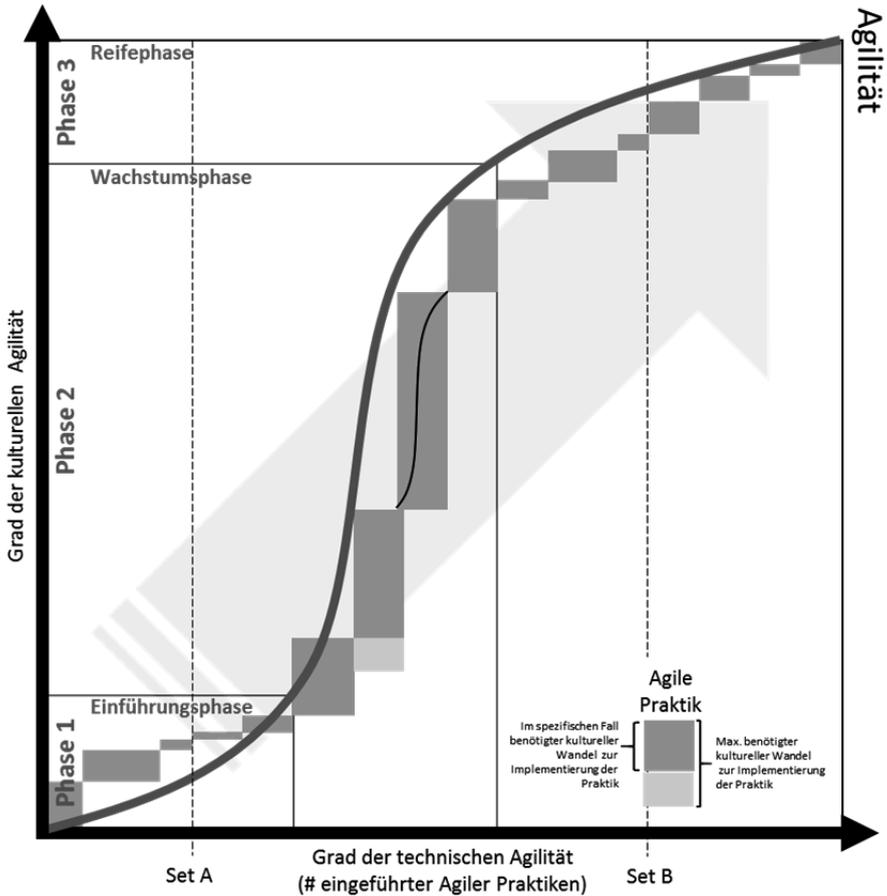


Abb. 1: Integration von technischer (x-Achse) und kultureller (y-Achse) Agilität

Generell stellt das beschriebene Vorgehen einen theoretischen Weg dar. In der Realität ist meist eine gewisse Agilität (vorwiegend technische Agilität) bereits inhärent im Unternehmen existent; die Transition beginnt somit nicht bei Null (Abbildung 1, links unten). Zudem ist wichtig, dass das Unternehmen stets eine eigene Vision der Agilität angepasst für das Unternehmen und seinen jeweiligen Kontext anstrebt; wie es in der Agilen Potenzialanalyse empfohlen wird. Was bedeutet, es ist durchaus legitim, sich mit einer nicht vollständigen Ausprägung der Agilität im Unternehmen zufrieden zu geben.

In den vorangestellten Ausführungen wird der komplette Weg des Unternehmens in seiner Veränderung bis hin zum gesetzten Agilitätsziel betrachtet. Ein weiterer wichtiger Teilaspekt auf eben diesem Weg ist die Betrachtung einzelner agiler Praktiken und ihrer Reihenfolge. Auch für Praktiken gilt der oben geschilderte Zyklus in gleicher oder zu-

mindest ähnlicher Form, jedoch auf einer anderen Detailstufe und in Abhängigkeit der bisher eingeführten Praktiken. Jede Praktik benötigt ein gewisses kulturelles Bewusstsein, um sie im Unternehmen einzuführen und zu etablieren. Ein gutes Beispiel hierfür ist das Daily Meeting, wie in Scrum beschrieben: Wichtig ist nicht nur die reine Umsetzung des täglichen Treffens zum Statusabgleich. Die Entwicklung eines tieferen Verständnisses dafür, warum man sich trifft, um so die Praktik in der täglichen Routine zu verankern, ist ebenfalls notwendig. Auch dieser Weg kann schrittweise angegangen werden: In einem ersten Schritt wird das Meeting mit seinen drei Kernfragen als Regeltermin im Kalender eingetragen. Im nächsten Schritt werden die Fragen nach einer gewissen Zeitspanne gelöscht, nachdem sie in die Denkweise der Beteiligten übergegangen sind. Im dritten und abschließenden Schritt wird der periodische Termin selbst aus dem Kalender gelöscht, sobald das volle *technische und kulturelle* Bewusstsein dieser Praktik erreicht ist: die Praktik hat sich damit selbst verstetigt.

## 4 Zusammenfassung

Wir konnten aufzeigen, dass *technische* und *kulturelle Agilität* eng miteinander verknüpft sind und dass beide Aspekte im Zuge der Einführung bzw. Anpassung des (agilen) Entwicklungsvorgehens im Unternehmen gemeinsam betrachtet werden sollten. In dieser Verknüpfung spielt die *technische Agilität* nach unserem Dafürhalten die führende Rolle, wobei jedoch stets eine gewisse *kulturelle Agilität* erforderlich ist, um die Veränderungen nachhaltig in den Köpfen der Beteiligten zu verankern. Es reicht nicht aus, eine agile Vorgehensweise unter rein technischen Gesichtspunkten im Unternehmen einzuführen. Die Betroffenen müssen diese Veränderung bewusst unterstützen, indem agile Werte und Prinzipien reflektiert werden; sich somit als Wert und Grundannahme verstetigen. Im Umkehrschluss macht die Einführung einer losgelösten *kulturellen Agilität* (d.h. einer ausschließlichen Propagierung der agilen Werte und Prinzipien) wenig Sinn, wenn diese nicht durch die dazugehörigen und passenden agilen Methoden und Praktiken „zum Leben erweckt werden.“

Unsere Betrachtungen zeigen, dass es lohnenswert ist, die Kombination beider Aspekte unter Forschungsgesichtspunkten weiter zu betrachten. Ein Ansatzpunkt bietet hierbei die dedizierte Betrachtung einzelner agiler Praktiken (z.B. aus den Methoden Scrum oder XP), um herauszufinden, welche konkrete kulturelle Agilität zur nachhaltigen Einführung der jeweiligen Praktiken notwendig ist, damit sie sich im Unternehmen und den Köpfen der beteiligten Personen verankern. Mit diesem Schritt würde der Zyklus, beginnend mit der Zieldefinition der agilen Transition, über die Potentialanalyse zur Identifikation geeigneter agiler Praktiken, bis hin zur schrittweisen Einführung einzelner Praktiken im Unternehmen geschlossen. Ein weiterer interessanter Forschungsaspekt ist die Frage: Welche typischen Konflikte und Probleme treten auf, wenn sich *technische Agilität* und *kulturelle Agilität* nicht auf dem Gleichgewichtspfad befinden?

## Literaturverzeichnis

- [Ab03] Abrahamson, P. et al.: New Directions on Agile methods: A Comparative Analysis. In Dillon, L., Tichy, W. (Hrsg.): Proc. of 25th International Conference in Software Engineering, Portland Oregon 2003. S. 244-254, 2003.
- [Be01] Beck, K. et al.: Manifesto for Agile Software Development, <http://agilemanifesto.org/>, Stand: 18.05.2015.
- [BT04] Boehm, B.; Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley, 2004.
- [Di15] Diebold, P. et al.: What Do Practitioners Vary in Using Scrum? In Lassenius, C. et al. (Hrsg.): Proc. of XP2015, Helsinki 2015. Springer LNBIP 212, S. 40-52, 2015.
- [DZ14] Diebold, P.; Zehler, T.: Top-Down vs. Bottom-Up – Agilität in regulierten Umgebungen. SQ Magazin 32 (Sept. 2014), S. 30-31, 2014.
- [VO15] Version One: 9th Annual State of Agile Development Survey. <http://info.versionone.com/state-of-agile-development-survey-ninth.html> . 2015.
- [Sc04] Schein, E. H.: Organizational Culture and Leadership, 3<sup>rd</sup> Edt., John Wiley & Sons, S. 25 – 37, 2004.
- [So08] Sood, A.: Understanding Technology Evolution: The Fallacy Of The S-Curve, <http://mthink.com/book/understanding-technology-evolution-fallacy-s-curve/> . 2008
- [HK14] BPM-Labor der Hochschule Koblenz: Internationale Studie: Status Quo Agile 2014. [www.status-quo-agile.de](http://www.status-quo-agile.de)

# Erkenntnisse und Impulse eines Projektmanagers

Torsten J. Koerting<sup>1</sup>

**Abstract:** Kein Projekt gleicht dem anderen, und speziell für komplexe und herausfordernde Projekte. Jedes Projekt ist anders. In meinen aktuellen Projekten gibt es so viele Ideen, Impulse und Herangehensweisen, die hilfreich sind und ganz individuell und spontan das Umfeld adressieren, aus dem sie hervorgehen. Diese wertvollen Impulse möchte ich beschreiben, reflektieren und mit vielen anderen Menschen teilen. Ich möchte aber nicht nur Erkenntnisse sammeln, sondern auch Bekenntnisse machen: Jeder Projektmanager erlebt in herausfordernden Situationen, dass die Dinge, die er anstößt, nicht zu dem gewünschten Ergebnis führen – vielleicht sogar, dass sie die Situation noch weiter verschlimmern. Auch diese Bekenntnisse möchte ich reflektieren und teilen.

**Keywords:** Projektmanagement, Erkenntnisse und Impulse, komplexe Projekte und Projekte im Schiefelage.

## 1 Start empty !

*Auch wenn es erstaunlich scheinen mag – ein Urlaub und ein komplexes, herausforderndes Projekt stellen an ihre Hauptdarsteller dieselben Anforderungen: Sie müssen alles hinter sich lassen.*

### 1.1 Eine Projektkrise lässt keine Freiräume

Wenn ein Projektmanager in ein neues Projekt startet, dann sollte er auf eines sehr stark achten: Dass er aus keinem seiner vorherigen Projekte noch Verpflichtungen mitbringt oder Restaktivitäten zu erledigen hat. Gerade die Anfangszeit eines neuen Projekts verlangt dem Projektmanager alles ab. Es ist eine sehr arbeitsintensive Zeit, die ihn auch psychisch stark fordern wird. Zu glauben, dass man in dieser Zeit noch übriggebliebene Aufgaben aus einem Vorläufer-Projekt erledigen kann, ist ein absoluter Trugschluss. Es lassen sich auch nicht irgendwelche Parallelprojekte stemmen, seien sie noch so klein – vergessen Sie das einfach. Ein Tag in einem Krisen-Projekt lässt keinerlei Freiräume für andere Aktivitäten. Punkt.

---

<sup>1</sup> [www.torstenkoerting.com](http://www.torstenkoerting.com), Mondorfer Weg 1d, 61352 Bad Homburg, [results@torstenkoerting.com](mailto:results@torstenkoerting.com)

## 1.2 Ein gutes Gewissen ist die beste Erholung

Das Gleiche gilt für den Urlaub: Wer wirklich Urlaub machen und seine Zeit mit seiner Familie, Freunden, dem oder der Liebsten so richtig genießen will, der sollte weder unerledigte Arbeit mitnehmen, noch den Vorsatz, gemütlich am Pool die ganze Fachliteratur zu wälzen, die das ganze Jahr über liegen geblieben ist. Auch wenn das heißt, dass die Tage und Nächte vor Urlaubsantritt stressig werden, weil sich sonst nicht alles erledigen lässt – das ist egal. Oder sagen wir es mit Mark Twain: „Gäbe es die letzte Minute nicht, so würde niemals etwas fertig.“ Der Punkt ist: Nur ein freier Kopf mit einem guten Gewissen kann sich wirklich erholen und Kraft tanken für das, was ihm nach der Rückkehr bevorsteht. Deshalb gilt auch hier: Start empty.

## 2 Die wichtigste Aufgabe in einer Krisensituation

*Ein Projektmanager hat viel zu tun – seitenlange To-do-Listen lassen sich mit dem füllen, was er jeden Tag in Angriff zu nehmen hat. Seine wichtigste Aufgabe jedoch fehlt auf den meisten To-do-Listen – ein großer Fehler. Denn ohne sie geht gar nichts.*

### 2.1 Resignierte Projektmitarbeiter

Übernimmt ein Projektmanager ein Krisen-Projekt, dann sieht er sich meist mit zwei unterschiedlichen Haltungen der Projektmitarbeiter konfrontiert: Die einen haben eine sehr hohe Erwartung an ihn und sind froh, dass er endlich da ist, weil mit ihm ein neuer frischer Wind aufkommt und ab jetzt hoffentlich alles besser wird. Die anderen – und das ist die weitaus größere Gruppe – haben dagegen schon längst resigniert, sind desillusioniert, leiden stark unter den Konflikten im Team und unter der vergifteten Atmosphäre, die es dort auch oft gibt.

Diese Menschen zu motivieren, sie mitzunehmen auf eine große Reise, ist deshalb die wichtigste Aufgabe eines Projektmanagers, vor allem in der ersten Zeit des Projekts. Durch seine Präsenz und Sichtbarkeit und durch seine Kommunikation muss er deutlich machen, welches große gemeinsame Ziel sie alle zusammen ansteuern. Um im Bild zu bleiben: Er darf es nicht dabei belassen, ihnen zu erklären, wie sie ein Boot bauen, sondern er muss ihnen die Insel am Horizont zeigen, die es zu erreichen gilt.

### 2.2 Entscheidend: Drei Leadership Principles

Doch womit motiviert der Projektmanager in einer solchen Situation die Projektmitarbeiter? Für mich haben sich dabei folgende drei Leadership Principles bewährt:

- **Consistent Delivery without excuses:** Der Projektmanager hält seine Zusagen immer zu hundert Prozent ein.

- **Walk the Talk:** Der Projektmanager lebt das vor, was er sagt, und auch das, was er von anderen erwartet.
- **Create Wow!-Moments:** Durch viele kleine Gesten erzeugt der Projektmanager immer wieder ein spezielles Zusammengehörigkeitsgefühl. Er zeigt dem Team, dass es etwas Besonderes ist – so tragen er eine Energie in das Projekt hinein, die alle Beteiligten anstecken wird.

Diese Wow-Momente – sie leben im Wesentlichen davon, dass ein Projektmanager in der Lage ist, sich empathisch auf seine jeweiligen Gesprächspartner einzustellen und ihnen zu zeigen, dass er sie und ihre Bedenken, ihre wie auch immer geartete aktuelle Situation versteht. Er hat keine Schwierigkeiten damit, die Projektmitarbeiter zu loben oder ihnen seine Anerkennung zu zeigen – weder im persönlichen Gespräch noch in jeglicher schriftlicher Kommunikation. Außerdem beherrscht er auch die Kunst der kleinen Aufmerksamkeiten, sei es nun in Form von Schokoriegeln, die er zu einer Besprechung mitbringt oder in Form von Getränken, die er genauso in den Meetingraum trägt wie alle anderen auch. Der Projektmanager weiß, dass er in seiner Rolle immer auch derjenige ist, der alle anderen unterstützt und dafür sorgt, dass sie ihre Arbeit tun können – eine der besten Motivationsfaktoren überhaupt.

### 3 You can't manage what you don't measure

Es ist fast schon eine Binsenweisheit – dass man nur das managen kann, was man auch misst. Das Komische daran: Obwohl dies bekannt und hinlänglich nachgewiesen ist, halten sich viele Unternehmen und Projekte nicht daran. Mit fatalen Konsequenzen.

#### 3.1 Wissen, was getan werden muss

Nicht nur in Krisen-Situationen, sondern auch in „gesunden“ Situationen ist in vielen Unternehmen im Sinne von laufenden Prozessen bzw. Projektaktivitäten nicht transparent, was man erreicht hat, was aktuell passiert, welche Fortschritte das Projekt macht, wie die laufenden Lieferaktivitäten einer bestimmten Einheit aussehen. Ganz zu schweigen von Prozesskosten und Prozesseffizienz, die ebenfalls nicht überwacht und gemessen werden.

KPI (Key Performance Index) oder Dashboards sind die dazugehörigen Methoden bzw. Instrumente dafür. Ich fasse sie gerne unter dem Begriff „Optics“ zusammen, weil sie Dinge sichtbar machen. Wer sie anwendet, hat gleich dreimal etwas davon:

- 1 Man sieht und versteht, wie es um das Projekt oder die Liefersituation bestellt ist – jenseits einer auf Rot, Grün oder Gelb stehenden Projektampel.

- 2 Aus den Ergebnissen der „Optics“ lassen sich Maßnahmen ableiten, die das, was da möglicherweise in Schiefelage geraten ist, wieder auf die richtige Bahn setzen.
- 3 Die Ergebnisse dieser Maßnahmen lassen sich ebenfalls messen und dadurch sichtbar machen – das ist entscheidend, um herauszufinden, ob die aufgesetzten Maßnahmen auch zu dem gewünschten Ergebnis führen.

### **3.2 Ungenutztes Verbesserungspotenzial**

In vielen Projekten und Liefersituationen finde ich zwar durchaus die Situation vor, dass KPI und Dashboards definiert sind und auch dargestellt werden – das heißt aber nicht, dass damit schon automatisch alles gut ist. Denn oft sind die falschen KPI etabliert – oder die Zahlenwerke werden falsch betrachtet bzw. interpretiert. Wichtig es hier, die jeweiligen „Optics“ entsprechend zu adjustieren, damit das Unternehmen genau die „Optics“ bekommt, die es braucht und die es ermöglichen, beispielsweise die Schiefelage noch besser zu verstehen, sie transparenter zu machen, korrekt zu messen und natürlich auch die entsprechenden Maßnahmen abzuleiten und die (erwarteten) Ergebnisse messbar und sichtbar zu machen.

Denn Fakt ist: Wer nicht erfasst, was er tut, wird immer nur ins Blaue hinein wirtschaften und – was noch viel schlimmer ist – jegliches Verbesserungspotenzial vergeuden.

## **4 Wahrnehmen, Reflektieren, Handeln**

*Nicht nur in Krisen-Projekten, sondern auch in allen anderen Projekten gibt es sie: kleine Anzeichen dafür, dass die Projektwelt doch nicht so heil ist, wie alle tun. Diese Anzeichen gilt es wahrzunehmen und entsprechend darauf zu reagieren – in einem immerwährenden Zyklus.*

### **4.1 Öffnen Sie Ihre Sinne**

Gerade am Anfang eines Projekts ist es wichtig, mit einer konstruktiven Geisteshaltung den Blick auf die Projektsituation zu schärfen. Öffnen Sie ganz bewusst Ihre Sinne, achten Sie sehr sensibel auf die Realität, auf Veränderungen und ziehen Sie geeignete Konsequenzen daraus. Nur beim Erkennen dieser Veränderungen darf es jedoch nicht bleiben! Es muss immer eine bewusste Handlung folgen.

Es geht also darum, dass Sie hier einen kleinen Kreislauf aus Wahrnehmen (Sense), Reflektieren (Reflect) und Handeln (Act) einrichten, den Sie immer wieder durchlaufen bzw. wiederholen – analog zum Zyklus des Rapid Prototyping (Build – Measure – Learn). Denn nur so ist es möglich, dass Ihre Wahrnehmungen sich auch tatsächlich auf

das Projekt auswirken und so eventuell noch verhindern, dass aus dem Projekt ein Krisen-Projekt wird.

## 4.2 Zweimal pro Woche eine Stunde

Um den Kreislauf aus Wahrnehmen, Reflektieren und Handeln konsequent und effektiv zu durchlaufen, ist es wichtig, dass Sie sich dafür ganz bewusst Zeit nehmen – zweimal pro Woche eine Stunde reicht. Dies täglich zu tun, ist nicht empfehlenswert, denn dann wird es Ihnen schwerfallen, sich vom Tagesgeschäft zu lösen und einen übergeordneten Blick auf die Projektgeschehnisse zu werfen. Dies gelingt Ihnen besser, wenn das Durchlaufen des Zyklus nicht zu Ihrem Alltagsgeschäft gehört.

Nehmen Sie wie gesagt eine Stunde Zeit für einen Zyklus und beantworten Sie sich selbst dieser Fragen – am besten schriftlich:

- Wie ist die Stimmung aktuell im Projekt? Ist sie besser oder schlechter als vor drei Tagen?
- In welchen Situationen habe ich ein Störgefühl?
- Was hat sich generell verändert?
- Was läuft besser, was läuft schlechter als vor drei Tagen?

Überlegen Sie dann, woran es liegen könnte, dass gewisse Dinge besser und andere schlechter laufen – und entwickeln Sie Maßnahmen, die den schlechten Entwicklungen entgegenwirken und die guten Entwicklungen verstärken. Und drei Tage später fangen Sie mit diesem Vorgehen wieder von vorne an. ☺

## 5 Never eat alone ... oder ... “Guten Appetit!”

*Ein Projektmanager ist permanent mit neuen Kunden- und Dienstleistersituationen konfrontiert, ständig trifft er auf neue Menschen, die er nicht kennt und mit denen er aber eng zusammenarbeiten muss. Wichtig in solchen Situationen: Erstmal essen gehen!*

### 5.1 Auf ein starkes Netzwerk kommt es an

Kennen Sie das Buch „Geh nie alleine essen!“ von Keith Ferrazzi? Es ist ein Buch darüber, wie man ein gutes persönliches Netzwerk aufbaut und wie man in Verbindung mit seinen Kontakten bleibt – auch wenn es viele Hundert sind.

Projektmanager müssen immer wieder innerhalb kurzer Zeit viele neue Menschen kennenlernen, die für ihre tägliche Arbeit wichtig sind. Sie erleben und erfahren viele Charaktere, versetzen sich in neue Denkweisen hinein und leiten daraus unterschiedliche

Dinge ab: Beim Team geht es darum, schnell herauszufinden, was diese Menschen können, für welche Aufgaben sie gut geeignet sind und was entsprechend getan werden muss, um sofort wirksame Krisenmaßnahmen zu finden oder um das Team auf neue Beine zu stellen. Auf Kunden- und Auftraggeber-Seite gilt es, starke Bande zu knüpfen, eine Beziehung zu bilden und aufrechtzuerhalten, um Solidarität und Unterstützung für Entscheidungen zu finden.

## **5.2 Nutzen Sie jede Gelegenheit**

„Geh nie alleine essen!“ – besonders diese „Titel-These“ des Buches von Keith Ferrazzi sollten sich Projektmanager zu Herzen nehmen. Denn gerade sie müssen jede Gelegenheit nutzen, den Menschen, mit denen sie es im Rahmen eines Krisen-Projekts jeden Tag zu tun haben, intensiv zu begegnen. Intensiv deshalb, weil das Projekt in einer Krise steckt und damit in einer außergewöhnlichen Situation ist, und intensiv auch deshalb, weil Projektmanager im Tagesablauf wenige Freiräume haben – diese wenige verbleibende Zeit gilt es also, effektiv zu nutzen. Selbst wenn die Pausen für Mahlzeiten kurz und kaum im Voraus planbar sind – ganz im Gegensatz zum „normalen“ Arbeitsleben, in dem man meist zur selben Zeit mit denselben Kollegen zum Essen geht oder sich schon Tage im Voraus dafür verabredet: Irgendjemand aus dem Umfeld hat immer Zeit und ist ähnlich spontan und flexibel unterwegs wie der Projektmanager selbst.

## **5.3 Neuer Kommunikationsstil**

Deshalb: Essen Sie nie allein! Nehmen Sie immer jemanden mit – jemand, der in Ihrem unmittelbaren Umfeld arbeitet; jemanden, der vielleicht im selben Büro sitzt; jemanden, den Sie entfernt kennen, weil Sie ihn bisher erst einmal in einem Meeting getroffen haben und danach nie wieder; jemanden, den Sie noch gar nicht kennen und den Sie die ganze Zeit schon kennenlernen wollten. Essen Sie also gemeinsam mit diesem Menschen – Sie werden neue Informationen bekommen, neue Einblicke, Sie werden immer tiefer in die Materie einsteigen und den Gesamtkosmos Ihres täglichen Umfeldes immer besser verstehen.

Wenn Sie es schaffen, dann dehnen Sie diese Haltung auch auf andere Bereiche aus – auf die Kaffeepausen, auf das Abendessen. Sie werden schnell feststellen, dass Sie sehr viel intensiver kommunizieren, dass sich Ihr ganzer Kommunikationsstil ändern wird. Und Sie werden außerdem feststellen, dass Sie sich deutlich stärker vernetzen, als das im normalen Berufsalltag der Fall ist.

## **6 Jedes (Projekt-)Feuer braucht einen Maßnahmenplan**

*Wenn der Wald brennt, kommen Flugzeuge und löschen den Brand. Wenn es im Projekt brennt, rennen alle hilflos durcheinander und wissen nicht, was sie tun sollen. Dabei ist es doch ganz einfach.*

## 6.1 Ein Krisen-Projekt ist wie ein Wald

Als Projektmanager werden Sie einem Bereich, einem Projekt zugeordnet, das sich in einer Krisensituation befindet. Diese Situation sollen Sie nun stabilisieren. – Stellen Sie sich Ihr Einsatzgebiet als ein Waldstück vor, das eigentlich Erträge erwirtschaften soll, aber seit Jahren ungepflegt und verwahrlost brachliegt. Es gibt niemanden mehr, der das tote Unterholz herausholt, sich um kranke Bäume kümmert, um den Ungezieferbefall oder darum, dass auch immer wieder junge Bäume aufgeforstet werden.

Und wenn Sie hineingehen in diesen Wald, dann kommen Sie nie an sein Ende. Immer wenn Sie denken „Ah, da hinten ist er, der Zaun, der die Grundstücksgrenze markiert“, dann stellen Sie beim Näherkommen fest, dass Sie nur einer optischen Täuschung aufgesessen sind und der undurchdringliche Wald grenzenlos zu sein scheint – immer neue Ebenen, Täler und Hügelketten tun sich vor Ihnen auf.

Wenn Sie nun anfangen, diesen Wald aufzuräumen und einige Zeit darin verbringen, werden Sie schnell feststellen, dass es in diesem Wald auch immer wieder brennt. Manche dieser Brände sind klein und glimmen nur so vor sich hin, andere sind größer – den Rauch dieser Brände sieht man kilometerweit. Und es gibt Brände, die unter der Erde schwelen und ganze Waldgebiete im Verborgenen zerstören.

## 6.2 Wer erledigt was bis wann?

Der Punkt ist: Wenn Sie diese vielen Brände nicht gründlich aufspüren und löschen, dann werden sie den kompletten Wald zerstören. Ihn wieder aufzuforsten und Ertrag aus ihm zu generieren, können Sie dann vergessen.

Und in einem Krisen-Projekt ist das ganz ähnlich: Wenn es Probleme gibt, große oder kleine Konfliktherde, Dinge, die nicht rund laufen, dann müssen sie gelöst werden – und zwar nicht irgendwie, sondern immer in Form eines Maßnahmenplans mit Aktivitäten, Verantwortlichen und festgelegten Zielterminen: Wer erledigt was bis wann? Ganz wichtig dabei: Jeder Maßnahmenplan muss von einer Person, die klar und eindeutig benannt ist, überwacht und gesteuert werden. Nur so ist es möglich, aufflammende Probleme nachhaltig zu lösen und das Projekt zu stabilisieren.

## 7 Wer lacht, ist souverän

*Keine Frage: Menschen, die uns freundlich und mit einem Lächeln begegnen, empfinden wir als deutlich angenehmer und sympathischer als diejenigen, die griesgrämig durch die Gegend schleichen. Das ist auf der Straße nicht anders als in einem Krisen-Projekt.*

## 7.1 Lachen steckt an

Haben Sie schon einmal ganz bewusst ausprobiert, was geschieht, wenn Sie die Straße entlang laufen und Sie den Mitmenschen, die Ihnen begegnen, freundlich ins Gesicht schauen und sie anlächeln? Ich probiere das sehr häufig aus. Und 95 Prozent der Menschen, denen ich ein Lächeln schenke, lächeln mich ebenfalls freundlich an. Ihr Gesichtsausdruck verändert sich, er hellt sich förmlich auf. Sogar ihre Körperhaltung verändert sich. Sie gehen aufrechter und beschwingter. Sie können sich gar nicht dagegen wehren, das geschieht unbewusst. Meine feste Überzeugung ist: Alle Menschen fühlen sich besser, wenn sie angelächelt werden. Kennen Sie dieses Video dazu? Schauen Sie es sich einmal an: Lachen ist ansteckend. Sie werden vor dem Rechner sitzen und gar nicht verhindern können, dass Sie mitgrinsen oder gar lachen.

## 7.2 Ein Geist breitet sich aus

In einem Krisen-Projekt ist das nicht anders. Projektmanager und Autor Peter Taylor hat ein neues Buch geschrieben, es heißt „The Project Manager who smiled“. Er beschreibt darin eine sehr wichtige Haltung, die ich uneingeschränkt teile: Als Projektmanager haben wir eine Vorbildfunktion. Wenn ich mit heruntergezogenen Mundwinkeln die Flure entlang gehe und für niemanden einen freundlichen Blick übrig habe – dann werde ich es schwer haben, ein Team hinter mich zu bringen und das Projekt aus der Schiefelage zu retten. Wenn ich dagegen lächle, strahle ich etwas ganz anderes aus: Offenheit und Souveränität – meine ganz persönliche Geisteshaltung. Und weil Lächeln ansteckt, weil es sich fortpflanzt wie die Wellen, die ein ins Wasser geworfener Stein verursacht, „infiziere“ ich auch andere mit meiner ganz persönlichen Haltung. Und die wiederum stecken weitere Menschen im Projektumfeld damit an. Und so breitet sich eine Haltung, ein Geist über das ganze Projekt aus, der am Ende dafür sorgt, dass wir *ein* starkes Team sind, das gemeinsam an *einem* Strang zieht und das Projekt dreht.

Es gibt auch in einem Krisen-Projekt viele, viele Gelegenheiten, Spaß zu haben – und ohne Spaß, ohne Lachen geht es nicht. Als Projektmanager gilt es, Emotionen zu wecken und das Team mit auf die Reise zu nehmen. Und das ohne den Druck, der natürlich in solch einem Projekt herrscht, übermächtig werden zu lassen. Lächeln Sie! Strahlen Sie dadurch Offenheit und Souveränität aus! Das Verrückte dabei ist: Wir stecken uns selbst mit unserem Lächeln an. Denn selbst, wenn uns mal so gar nicht nach Lachen zumute ist – wer es trotzdem tut und seinen Mitmenschen ein Lächeln schenkt, erfährt spätestens durch die Reaktion der anderen, wie ansteckend das sein kann!

## 8 Nachhaltige Nachhaltigkeit nachhalten

*Zugegeben: Ein bisschen schwindelig kann einem bei dieser Überschrift schon werden. Dennoch drückt sie genau das aus, um was es mir hier geht: Ein Projektmanager muss unter allen Umständen dafür sorgen, dass geplante Aktivitäten auch tatsächlich umge-*

setzt werden. Und diese Nachhaltigkeit dann auch noch nachhalten. Wie das geht, lesen Sie hier.

## 8.1 Nachhaltigkeit vorleben

**Erstens:** Ein Projektmanager muss aufgesetzte Meilensteinpläne, Aktivitäten und Maßnahmen, die in unterschiedlichen Teams und Konstellationen entwickelt wurden, nachverfolgen, am besten in aufgesetzten Regelterminen und Jour fixes.

**Zweitens:** Er muss dieses Nachhalten delegieren und die Kollegen, die für die einzelnen Aktivitäten und Maßnahmen verantwortlich sind, dazu bringen, dies selbst zu übernehmen, sprich: Listen zu führen, zu tracken, nachzuhalten und dafür zu sorgen, die die Maßnahmen umgesetzt werden.

**Und drittens:** Er muss diese nachgehaltene Nachhaltigkeit anderer nachhalten. Das heißt: Selbst noch mal diejenigen tracken, die schon damit beschäftigt sind, die Aktivitäten anderer nachzuhalten.

Allen Beteiligten muss klar sein, dass das Management von Maßnahmen sehr nachhaltig umgesetzt wird – und dass der Projektmanager diese Nachhaltigkeit seinerseits nachhält. Wenn die Teammitglieder feststellen, dass es quasi keine Ausweichmöglichkeit gibt, dann passiert nämlich eins: Sie sorgen dafür, dass keine ihrer Maßnahmen, die sie entweder selbst verantworten bzw. die in der Verantwortung ihres Nachhaltens liegen, später geliefert wird bzw. nicht zu dem Zeitpunkt fertig ist, zu dem sie fertig sein sollte.

## 9 Erfolgsfaktoren: Statusreporting und Managementsystem

Ein wichtiger Erfolgsfaktor für diese „nachgehaltene nachhaltige Nachhaltigkeit“ ist ein klassisches Tool aus dem Projektmanagement: das Statusreporting. Dort wird regelmäßig Nachhaltigkeit eingefordert, und deshalb ist es auch sehr entscheidend, die Termine für das Statusreporting weder zu verschieben noch ausfallen zu lassen.

Ebenfalls ein wichtiger Erfolgsfaktor: ein effizientes Managementsystem, das sowohl der Projektmanager als auch die Kollegen, an die er das Nachhalten delegiert hat, haben sollten. Sie müssen ohne Wenn und Aber in der Lage sein, die erforderliche Nachhaltigkeit umzusetzen.

Dem Projektmanager kommt bei alledem natürlich eine Schlüsselrolle zu: Er muss diese Nachhaltigkeit klar zum Ausdruck bringen, vorleben und es muss auch Konsequenzen haben, wenn er feststellt, dass die Nachhaltigkeit nicht eingehalten wird, sprich: wenn zu liefernde Ergebnisse nicht zum vereinbarten Termin vorliegen oder vereinbarte Aktivitäten wie Telefonate, Termine, Gespräche etc. nicht stattgefunden haben. Nur dann lässt sich ein Projekt aus der Krise führen.

## 10 Was Drecks-Slides und Top-Shots gemeinsam haben

*Stellen Sie sich vor: In einer Präsentation gibt es ein Slide, auf dem eine Service-Level-Grafik abgebildet ist, die Ihren Bereich alles andere als gut aussehen lässt. Wie nennen Sie dieses Slide? „Slide, auf dem unser Bereich nicht so gut aussieht“? Ich hätte da einen Vorschlag für Sie.*

### 10.1 Was steht hinter einem Wort?

In einem Krisen-Projekt bewegen die Team-Mitglieder sehr viel, sie agieren intensiv und sie haben es fast ständig mit denselben Menschen zu tun. Da geschieht es unweigerlich, dass sich in vielen Bereichen so etwas wie ein eigener Sprachcode herausbildet – situationsbezogen, mit einer Bedeutung aufgeladen, die nur diejenigen verstehen, die dabei waren. Und so kommt es, dass ein Slide mit einer Service-Level-Grafik, die den betreffenden Bereich nicht so gut dastehen lässt, dann eben „Drecks-Slide“ heißt. Und alle wissen, was sich dahinter noch verbirgt: nicht nur die Tatsache, dass der jeweilige Bereich schlecht dasteht, sondern auch alles, was in diesem Zusammenhang damit besprochen wurde, was noch ansteht, welche Schwierigkeiten auf das Team zukommen. „Drecks-Slide“ fasst das alles schön zusammen.

Ein anderes Beispiel: Sie wissen, dass Sie nicht genügend Projektmitarbeiter haben, um eine ganz bestimmte Aufgabe zu bewältigen. Sie rufen also Ihr Team zusammen, erklären, dass Sie noch einen weiteren hochqualifizierten Mitarbeiter brauchen, beschreiben, welche Fähigkeiten er haben muss und warum Sie und das Team gerade diese speziellen Fähigkeiten brauchen, welches Budget Sie zur Verfügung haben und so weiter. Und irgendwann lassen Sie das Wort „Top-Shot“ fallen – und setzen damit den oben erwähnten emotionalen Anker. Wenn irgendwann nach diesem Meeting das Wort „Top-Shot“ fällt, dann weiß jeder der Beteiligten in derselben Sekunde, was sich hinter diesem Begriff verbirgt (Notwendigkeit der Einstellung eines neuen Mitarbeiters, Budget dafür, Ziele, die damit erreicht werden sollen etc.).

### 10.2 Setzen Sie Anker

Es hat viele Vorteile, solche Begriffe zuzulassen und zu benutzen: Sie sind emotional aufgeladene Anker, die einen komplexen Sachverhalt in einem Wort auf den Punkt bringen. Weil die Worte an sich – wie „Drecks-Slides“ beispielsweise – durchaus provokant sind, rufen sie starke Reaktionen hervor und bleiben so auch viel besser in den Köpfen der Mitarbeiter haften. Und nicht nur die Worte bleiben haften, sondern auch die ihnen zugeordneten Ideen. Die Notwendigkeit, bestimmte Anker zu setzen, ist übrigens nicht nur in Krisen-Projekten gegeben, sondern auch in ganz anderen Bereichen, im Marketing etwa. „Made to stick“ lautet auch dort die Devise – und unter genau diesem Titel haben Chip Heath und Dan Heath in einem Buch beschrieben, warum manche Ideen sang- und klanglos untergehen und andere Erfolg haben: <http://www.amazon.de/Made-Stick-Ideas-Survive-Others/dp/0812982002/>

## 11 Was haben Jongleure und Projektmanager gemeinsam

*Gleich zu Beginn eines Krisen-Projektes wartet Schwerstarbeit auf den Projektmanager: Er muss viele verschiedenen Themen adressieren und noch mehr Maßnahmen anstoßen. Wie ein Jongleur wirft er quasi viele Bälle in die Luft und versucht, sie alle gleichzeitig oben zu halten. Doch was passiert, wenn einer oder zwei Bälle abstürzen? Ganz einfach: Er tut das, was ein Jongleur auch tun würde – er lässt sie fallen.*

### 11.1 Immer schön die Übersicht behalten!

Steigt ein Projektmanager in ein Projekt ein, das sich in Schieflage befindet, oder in eine andere Projektsituation, die zu stabilisieren ist, sieht er sich vielen Aufgaben gegenüber, die gleichzeitig zu erledigen sind: Er beginnt mit der Analyse, muss das Team kennenlernen, mit unterschiedlichen Stakeholdern aus verschiedenen Bereichen sprechen, sei es aus dem Projekt selbst, beim eigenen Unternehmen, auf Kunden- oder Auftraggeberseite. Er setzt außerdem diverse Maßnahmen und Initiativen in Gang. Und es gibt vielleicht auch schon Quickwins, die er an dieser Stelle anstoßen oder gar schon realisieren kann.

Am Beginn eines solchen Projekts startet er also viele Maßnahmen und Initiativen gleichzeitig, eine ergibt sich aus der anderen, und viele verschiedene bilden ein Bündel, das wiederum neue Initiativen nach sich zieht. Wie ein Jongleur in der Zirkusmanege wirft er also einen Ball nach dem anderen in die Luft. Er startet mit einem Ball, dann kommen ein zweiter und dritter hinzu. Alles noch gut zu überschauen. Der vierte, der fünfte, sechste, siebte – jetzt nur keinen Fehler machen! Und immer schön die Übersicht behalten!

Die Gefahr dabei ist noch nicht einmal, dass ein oder zwei Bälle abstürzen – die Gefahr ist vielmehr, dass Jongleur oder Projektmanager sich von den abstürzenden Bällen bzw. Maßnahmen so aus dem Konzept bringen lassen bzw. versuchen, sie wieder auf die richtige Bahn zu bringen, dass das ganze Gebilde zusammenbricht.

### 11.2 Zwei verlorene Bälle sind zu verkraften

Was kann ein Projektmanager in diesem Fall tun? Er braucht vor allem einen scharfen Blick auf die unterschiedlichen Initiativen, die er angestoßen hat, und muss aus übergeordneter Perspektive bewerten,

- wie viele Initiativen er gestartet hat,
- ob er nicht zu schnell zu viele Maßnahmen angestoßen hat
- ob seine eigenen Fähigkeiten ausreichen, viele Bälle gleichzeitig in der Luft zu halten

- ob sich die Maßnahmen und Initiativen in ihren „Flugbahnen“ gegenseitig behindern und dadurch das gesamte Vorhaben Gefahr läuft, abzustürzen

Entscheidend ist für ihn jedoch, dass ihm zu jedem Zeitpunkt eines bewusst ist: Sobald ein Ball fällt bzw. eine Maßnahme schiefeht oder nicht greift, kann er nur eins tun – es mehr oder weniger ignorieren. Kümmert er sich zu intensiv darum, droht das gesamte Gebilde abzustürzen – und der Schaden wäre ein viel größerer.

# Self-Enforcing Networks als Tools zur Auswahl eines geeigneten (ggf. hybriden) Vorgehensmodells in IT-Projekten

Christina Klüver<sup>1</sup> und Jürgen Klüver<sup>2</sup>

**Abstract:** Die Auswahl eines geeigneten Vorgehensmodells (VM) für bestimmte Projekte oder als institutionelle Vorgabe erfordert nicht nur die Kenntnis der zahlreichen VM sondern auch die Zuordnung zu den Anforderungen des jeweiligen Projekts. Unser Artikel beschreibt die Möglichkeiten, die für diese Probleme ein von uns entwickeltes selbstorganisiert lernendes neuronales Netz bietet. Dies Netz, das Self Enforcing Network SEN, wird in seiner Grundlogik beschrieben; seine praktischen Verwendungsmöglichkeiten werden in verschiedenen Szenarien demonstriert. Die wesentlichen praktischen Vorteile von SEN sind Flexibilität sowie leichte Bedienbarkeit und Interpretation der Ergebnisse.

**Keywords:** Self-Enforcing Network, Selbstorganisiertes Lernen, Vorgehensmodelle

## 1 Einleitung

Die Auswahl eines geeigneten Vorgehensmodells stellt die Verantwortlichen bekanntlich vor große Herausforderungen: Die Vorgehensmodelle müssen generell bekannt sein, die Vorgehensweise in IT-Projekten weisen häufig besondere Charakteristika auf, die Kunden haben ihrerseits bestimmte Vorstellungen, die nicht unbedingt mit dem Auftrag kompatibel sind, und für Zertifizierungen müssen bestimmte Vorgaben eingehalten werden – um nur einige der wesentlichen Probleme zu nennen. Erschwerend kommt noch dazu, dass es seit einiger Zeit eine Vielfalt von Vorgehensmodellen gibt (u.a. [KL14]), die für verantwortliche Manager nur noch schwer zu durchschauen ist. Eine Unterstützung der entsprechenden Auswahlprozesse durch geeignete Softwaretools wäre demnach für die betriebliche Praxis eine wesentliche Hilfe.

Bekannt sind mittlerweile unterschiedliche Klassifizierungen (u.a. [WM10]) wie insbesondere konzeptionelle, sequentielle, inkrementelle, iterative, spiralförmige und agile Vorgehensmodelle, die sich im Laufe der Jahre durchgesetzt haben.

Wir stellen in diesem Artikel ein Softwaretool vor, das Entscheidungsunterstützungen bei der Auswahl von Vorgehensmodellen ermöglicht, die für ein neues Projekt am besten geeignet sind. Das Tool basiert auf einem von uns entwickelten künstlichen neuronalen

---

<sup>1</sup> Universität Duisburg-Essen, Fakultät WiWi, Universitätsstr. 12, 45117 Essen, christina.kluever@uni-due.de

<sup>2</sup> Universität Duisburg-Essen, COBASC-Research Group, Universitätsstr. 12, 45117 Essen, juegen.kluever@uni-due.de

len Netz, das auf der Basis des selbstorganisierten Lernens operiert – das Self Enforcing Network (SEN). Das Tool kann flexibel gemäß unterschiedlichen Bedürfnissen eines Benutzers eingestellt werden und liefert in einem Visualisierungsteil anschaulich die Resultate als Empfehlungen für die Benutzer. Es zeigt nach Bedarf auch, welche unterschiedlichen Vorgehensmodelle kombinierbar sind.

Die folgende Sektion beschreibt das methodische Vorgehen, um eine geeignete Datenbasis für das SEN zur Verfügung zu stellen. Im dritten Kapitel wird das Self Enforcing Network näher dargestellt; im vierten Kapitel wird die Operationsweise des SEN und die Visualisierung der Resultate an einigen praktischen Beispielen demonstriert. Wünschenswerte Erweiterungen des Tools und zusätzliche praktische Möglichkeiten werden am Schluss skizziert.

## 2 Der methodische Ansatz zur Auswahl von Vorgehensmodellen

Der hier dargestellte Prototyp des SEN enthält vierzehn Vorgehensmodelle – klassische wie agile -, wie z.B. „Wasserfall“, „Scrum“ und „RUP“. Diesen werden insgesamt sechzig Attribute zugewiesen, die als Kriterien bzw. als Charakteristika der jeweiligen Vorgehensmodelle fungieren. Um eine Zuordnung der entsprechenden Attribute zu den Modellen transparenter zu gestalten, werden die Kriterien in zwölf Gruppen aufgeteilt, die unterschiedliche Aspekte berücksichtigen. In Tabelle 1 werden die Gruppen mit einigen der Aspekte aufgeführt:

Gruppe	Anzahl Kriterien	Beispiele
Ökonomische Kriterien	6	Budget, Projektdauer, (Schulungs-)Aufwand...
Flexibilität	6	Einsatz, Benchmark-Fähigkeit, Unterstützung durch Benchmarksysteme (CMMI, ITIL...)
Projektcharakter	7	Erfordert Kreativität, diplomatischer Umgang im Projekt, Anzahl der Arbeitspakete
Umwelt	2	Beachtung neuer Entwicklungen, Änderungen innerhalb des Projektverlaufs
Dokumentation	3	Detailierungsgrad, Häufigkeit, Beteiligte
Risikomanagement	9	Verzögerung, Kundenkontakt, Änderung des Codes
Team	4	Größe, Erfahrungsgrad, Anzahl der MA
Kommunikation	5	Häufigkeit, Ort, Kommunikationswege
Akzeptanz	4	Offenheit für neue Methoden, Rollenmodell, Hierarchie-Support
Phasenbindung	6	Qualitätsmanagement, Methodenbindung, festgelegte Prozesse
Technik	4	Herstellerabhängigkeit, Entwurfsobjektbindung
Formalisierung	4	Formalisierungsgrad, QS-Unterstützung, Zertifizierungssupport

Tab. 1: Beispiele für die Zuordnung der Kriterien

Die Auswahl der Kriterien entstand auf der Basis von Literaturrecherche und durch Diskussionen mit Managern bzw. Mitarbeitern verschiedener Unternehmen, die mit unterschiedlichen VM vertraut sind und über praktische Erfahrungen mit mindestens einem VM verfügen.

Zunächst wurde für jedes VM eine Liste mit den wesentlichen Charakteristika erstellt. Anschließend wurden die Listen zusammengeführt und Doppelungen eliminiert, so dass alle Kriterien nur einmal aufgeführt werden.

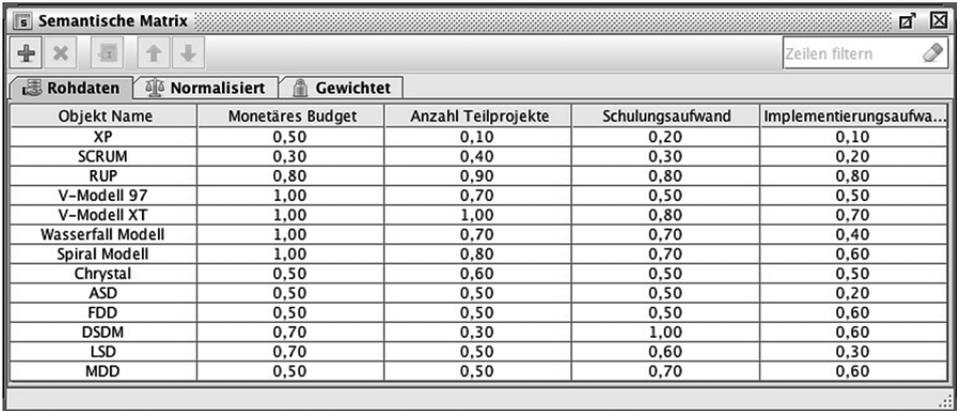
Im nächsten Schritt erfolgten die Bewertungen der Charakteristika für alle VM mit Werten zwischen 0.0 und 1.0; 1.0 bedeutet, dass das Charakteristikum für das VM sehr wichtig ist, 0.0 natürlich, dass es völlig unwesentlich ist. Beispielsweise wird die "*Dokumentation*" hinsichtlich des Detaillierungsgrads bei RUP mit 1.0 bewertet, hingegen bei Build & Fix mit 0.0. Bei dem Kriterium "*Monetäres Budget*" wird angenommen, dass Unternehmen, die das V-Modell XT, das Spiral- oder Wasserfallmodell verwenden, über ein sehr hohes Budget verfügen können; in diesem Fall steht eine 1.0 wenn das Budget mehr als 1.000.000 € betragen kann, entsprechend eine 0,1 wenn es sich um ein Projekt handelt, das maximal über 5.000 € verfügt. "*Projektdauer*" wird mit 0.3 bewertet, wenn es sich um Projekte handelt, die maximal für 30 Tage geplant werden und für die als VM eher XP oder Scrum in Frage kommen. Bei dem V-Modell XT oder Wasserfall Modell wird die Dauer mit 1.0 bewertet, da die Projekte mit einer Laufzeit von mehr als fünf Jahren vorgesehen werden können.

Auf dieser Weise wurde eine Matrix entwickelt mit vierzehn Zeilen für die VM als Objekte und sechzig Spalten mit den bewerteten Attributen, die dem Netzwerk 1:1 übergeben wird (s. Abb. 1).

### 3 Das Self-Enforcing Network (SEN)

Beim SEN handelt es sich um ein künstliches neuronales Netzwerk, das durch selbstorganisiertes (bzw. nicht überwachtes) Lernen charakterisiert ist. Ein SEN besteht aus drei wesentlichen Komponenten: einer semantischen Matrix, dem eigentlichen Netzwerk und der Visualisierungskomponente (für detailliertere Beschreibungen sowie verschiedene Anwendungsbeispiele vgl. z.B. [KKS12] sowie [KK13]).

In der *semantischen Matrix* sind die „semantischen“ Beziehungen zwischen Objekten und Attributen enthalten, die vom SEN geordnet werden sollen; hier geht es um die Beziehung zwischen Vorgehensmodellen und deren Charakteristika. Die semantische Matrix kann zum einen binär codiert werden, wenn ein Attribut zu einem Objekt vorliegt (1) oder nicht vorliegt (0), oder zum anderen reell codiert, wenn man die relative Zugehörigkeit des Attributs zu einem Objekt ausdrücken möchte (stark, mittel, schwach usw.). In Abb. 1 wird ein Ausschnitt gezeigt:



Objekt Name	Monetäres Budget	Anzahl Teilprojekte	Schulungsaufwand	Implementierungsaufwa...
XP	0,50	0,10	0,20	0,10
SCRUM	0,30	0,40	0,30	0,20
RUP	0,80	0,90	0,80	0,80
V-Modell 97	1,00	0,70	0,50	0,50
V-Modell XT	1,00	1,00	0,80	0,70
Wasserfall Modell	1,00	0,70	0,70	0,40
Spiral Modell	1,00	0,80	0,70	0,60
Chrysal	0,50	0,60	0,50	0,50
ASD	0,50	0,50	0,50	0,20
FDD	0,50	0,50	0,50	0,60
DSM	0,70	0,30	1,00	0,60
LSD	0,70	0,50	0,60	0,30
MDD	0,50	0,50	0,70	0,60

Abb. 1: Ausschnitt aus der semantischen Matrix

Aus der semantischen Matrix wird das eigentliche *Netzwerk* des SEN konstruiert. Dies geschieht durch die Generierung einer Gewichtsmatrix, in der den Verbindungen zwischen den Objekten und den Attributen Gewichte zugeordnet werden. Die Objekte und die Attribute stellen Neuronen dar, die miteinander voll verbunden sein können. Bei der Initialisierung der Gewichtsmatrix sind alle Gewichte  $w = 0$ . Aus der semantischen Matrix werden mit folgender Lernregel die Gewichtswerte generiert:

$$(t + 1) = w(t) + \Delta w \quad (1)$$

$$\Delta w = c * v_{sm}$$

Hier ist  $c$  eine Lernrate; diese wird gewöhnlich als  $c = 0.1$  gesetzt. Bei  $v_{sm}$  handelt es sich um den entsprechenden Wert aus der semantischen Matrix, also dem Beziehungswert zwischen einem Objekt und einem Attribut.

Die Dynamik eines SEN wird wie üblich bei neuronalen Netzen durch eine spezifische Aktivierungsfunktion generiert; das SEN bietet hier sechs Möglichkeiten zur Auswahl an. Bei praktischen Anwendungen reicht es häufig, die einfache *lineare Aktivierungsfunktion* ( $a_j$ ) zu verwenden:

$$a_j = \sum w_{ij} * a_i \quad (2)$$

Bei gewünschten Dämpfungen der Ergebnisse steht die von uns entwickelte *logarithmisch-lineare Aktivierungsfunktion* zur Verfügung:

$$a_j = \sum \begin{cases} \lg_3(a_i + 1) * w_{ij}, \\ \lg_3(|a_i + 1|) * -w_{ij} \end{cases} \quad (3)$$

Diese Funktion wurde vor allem deshalb entwickelt, da bei großen Netzen wegen der Linearität der einfachen linearen Funktion die Aktivierungswerte der Neuronen rasch sehr groß werden können. Die zusätzliche Einführung des Logarithmus bewirkt eine entsprechende Dämpfung dieser Werte.

Zusätzlich verfügbar sind weitere Varianten der linearen Aktivierungsfunktion, nämlich die sog. *lineare Aktivierungsfunktion mit Mittelwert* und die sog. *Enforcing Activation Function*, die ebenfalls von uns entwickelt wurden. Außerdem gibt es die Möglichkeiten, die *Tangens Hyperbolicus Funktion* sowie die bekannte logistische Aktivierungsfunktion zu verwenden [KKS12].

Die im nächsten Kapitel gezeigten Anwendungsbeispiele verwenden nur die einfache lineare Aktivierungsfunktion.

Für die Klassifikation eines oder mehrerer Objekte werden die entsprechenden Attributs-Neuronen extern aktiviert, die die Objekte charakterisieren. Die Endaktivierungswerte der Objektsneuronen entscheiden einerseits über die Ähnlichkeit zwischen verschiedenen Objekten (Vorgehensmodellen) und andererseits über die Eignung der verschiedenen Vorgehensmodelle für ein bestimmtes Projekt. „Externe Aktivierung“ bedeutet die Belegung eines Neurons mit einem bestimmten numerischen Wert – im folgenden Beispiel 1.0 ; anschließend berechnet das Netz über die Aktivierungsfunktion die Werte der Endaktivierungen der Objektneuronen. Die folgende Graphik demonstriert dies Verfahren; die externe Aktivierung wird sozusagen von den Attributsneuronen an die Objektneuronen in modifizierter Form durch die lineare Aktivierungsfunktion weiter gegeben:

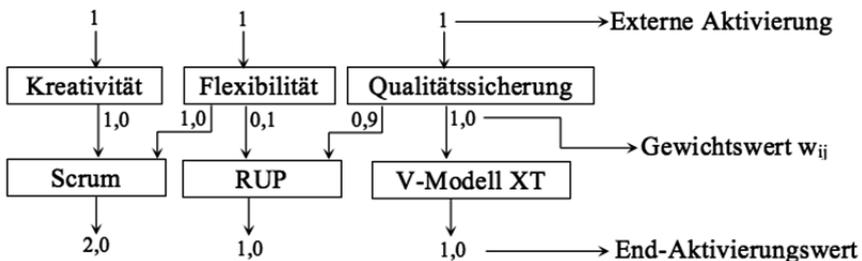


Abb. 2: Beispiel für die Aktivierung einzelner Neuronen. Dargestellt wird nur ein Teil der Verbindungen, um die Übersichtlichkeit zu gewährleisten

Aufgrund der Stärke und Anzahl der Verbindungen wird offenbar „Scrum“ am stärksten aktiviert.

Für die *Visualisierung* wird, wie bemerkt, die Stärke der Endaktivierung zwischen den Objektneuronen verglichen. Dabei sind Objekte mit annähernd gleichen Endaktivierungen ähnlich bzw. Objekte mit stark unterschiedlichen Endaktivierungen unähnlich. SEN visualisiert dies durch die Darstellung der Distanz zwischen den Objekten. Eine nahe Distanz entspricht einer hohen Ähnlichkeit und umgekehrt. Die Distanz zwischen den jeweiligen Objekten ist die Differenz ihrer jeweiligen Endaktivierungswerte.

SEN bietet drei verschiedene Möglichkeiten an, die Ergebnisse für einen Benutzer anschaulich darzustellen. Die erste Möglichkeit ist die von uns so genannte *Zentrumvisualisierung*. Diese dient vor allem dazu, dem Benutzer graphisch darzustellen, welche Vorgehensmodelle für sein Projekt am besten geeignet sind. Dazu wird das Projekt als Vektor mit den entsprechenden Anforderungen ins Zentrum eines zweidimensionalen Gitters platziert und durch die externe Aktivierung der Attributsneuronen werden dann die Symbole für die verschiedenen Vorgehensmodelle ins Zentrum des Gitters gezogen. Je näher ein Modellsymbol dem Zentrum ist, desto geeigneter ist das entsprechende Vorgehensmodell für das Projekt des Benutzers.

Die zweite Möglichkeit ist die sog. *Kartenvisualisierung*. Diese demonstriert die relative Ähnlichkeit der Objekte zueinander, indem die Objekte, hier die Vorgehensmodelle, zu Clustern geordnet werden. Daraus lässt sich entnehmen, welche Vorgehensmodelle ggf. füreinander substituierbar sind. Technisch wird dies, sehr einfach ausgedrückt, durch die Berechnung der Euklidischen Distanz realisiert: Je größer die Distanz zwischen zwei Vektoren der VM ist, desto weiter voneinander werden diese im Koordinatensystem platziert.

Die dritte Möglichkeit ist die sog. *Expertenansicht*. Bei dieser werden die Vorgehensmodelle tabellarisch danach geordnet, welche für ein Projekt am besten geeignet sind. Die Expertenansicht ist demnach eine andere Darstellungsform der Zentrumsvisualisierung. In den folgenden Beispielen werden alle drei Visualisierungsmöglichkeiten gezeigt.

## 4 Ergebnisse

Um die Potentiale der Auswahl mit einem SEN zu veranschaulichen werden unterschiedliche Szenarien aufgeführt.

### 4.1 Ordnung von Basismodellen in einschlägigen Clustern

Für eine Clusterordnung von Basismodellen gemäß der Option Kartenvisualisierung wurden lediglich einige der bekanntesten Modelle, nämlich Wasserfall-, Spiral-, V-Modell 97, RUP, V-Modell XT, Scrum und XP gewählt. Für alle Vorgehensmodelle (VM) wurden die Werte in der semantischen Matrix gemäß ihrer Ausprägung für das jeweilige VM definiert.

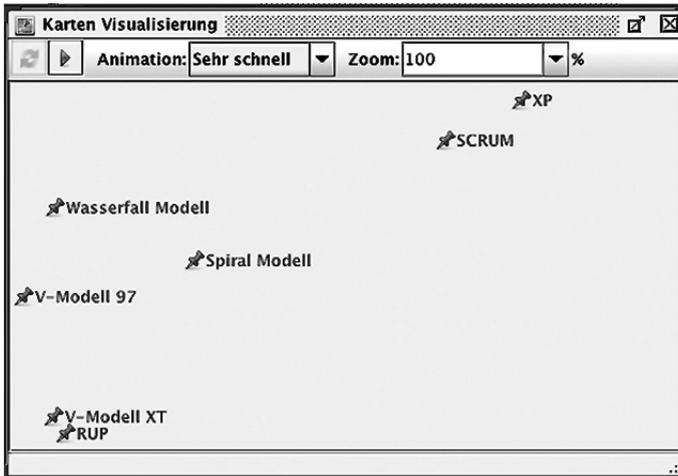


Abb. 3: Clustern der Basismodelle

Anhand der Platzierung der einzelnen VM auf der Visualisierungskarte sind drei Cluster erkennbar. Scrum und XP bilden gemeinsam einen Cluster; das Ergebnis ist nachvollziehbar, da beide für kleinere Projekte geeignet sind und insbesondere zu den bekanntesten agilen VM gehören. In der Softwareentwicklung wird entsprechend eine Kombination von Scrum mit XP als vielversprechend empfohlen [Hi09] und diese wurde in der Praxis bereits durchgeführt [DD08; DD14].

RUP und V-Modell XT bilden ebenfalls einen Cluster. Beide können als generische Metamodelle betrachtet werden, Risikomanagement sowie Qualitätssicherung werden berücksichtigt und die Vorgehensweise wird ganz genau geplant.

Die Clusterung am linken Rand ist insbesondere bei dem Wasserfall- und V-Modell 97 nicht überraschend, da letzteres eine Erweiterung des bekannten Wasserfallmodells ist.

Da mit dem Spiralmodell versucht wurde, andere Vorgehensmodelle als "Sonderfälle" zu integrieren [Ga08, 72] ist es ebenfalls plausibel, dass dieses VM eher zentriert platziert wird. Die Ergebnisse des SEN sind demnach durchaus kompatibel mit den bekannten Eigenschaften der verschiedenen VM und deren Ähnlichkeiten untereinander.

## 4.2 Erweiterung eines SEN durch weitere Vorgehensmodelle

Mittlerweile existiert eine große Anzahl an Vorgehensmodellen, die in der Praxis unterschiedlich eingesetzt werden [KL14]. Wenn man neue Vorgehensmodelle durch SEN einordnen lassen will, ist es ohne große Probleme möglich, die bisherige semantische Matrix durch die Attribute dieser neuen VM zu erweitern und die neuen VM entsprechend einer der Optionen ordnen bzw. sie auf ihre Eignung für ein bestimmtes Projekt überprüfen zu lassen.

Für sieben neue VM, die im bisher verwendeten SEN noch nicht enthalten waren, ergibt eine Clustering in der Kartenvisualisierung folgendes Bild:

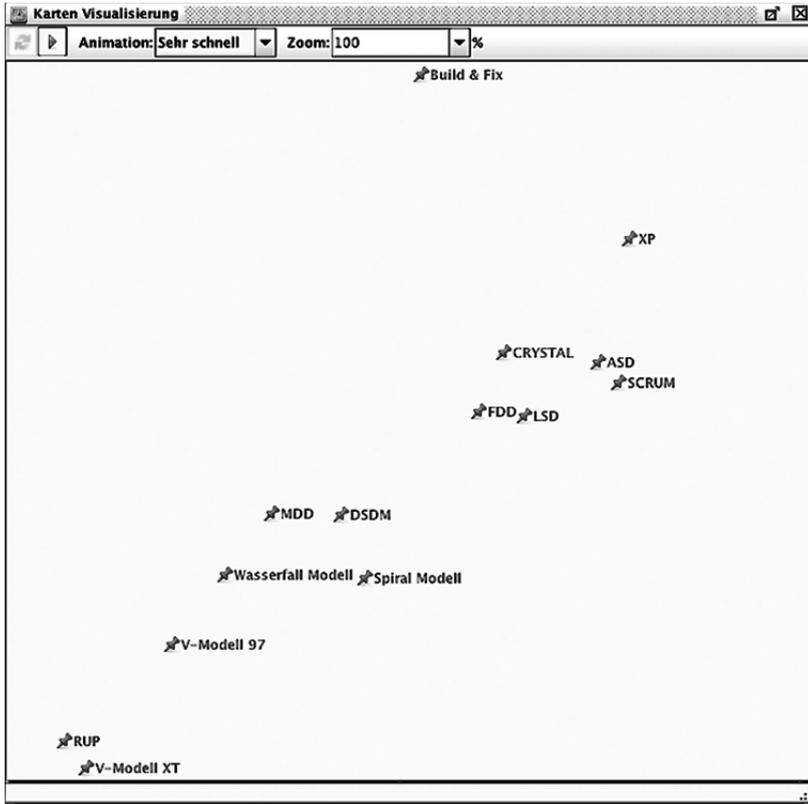


Abb. 4: Einordnung neuer VM

Die Erweiterung der semantischen Matrix durch die neuen Vorgehensmodelle führte durch das Lernverfahren zu einer leicht veränderten Ordnung der bereits implementierten Vorgehensmodelle; die grundsätzliche Ordnung ist jedoch erhalten geblieben.

In den letzten Jahren wurden insbesondere agile Vorgehensmodelle miteinander verglichen, um Vor- wie Nachteile, Gemeinsamkeiten und Unterschiede herauszuarbeiten [CH11; DD14; FC14]. Die Ergebnisse von SEN scheinen mit diesen Untersuchungen kompatibel zu sein.

### 4.3 Auswahl eines geeigneten Vorgehensmodells für ein Projekt

Damit eine konkrete Empfehlung durch das SEN erfolgen kann, muss ein Benutzer einen Vektor erstellen, einen sog. Eingabevektor, bestehend aus den Attributen, die das Projekt charakterisieren.

Als Beispiel wird unsere eigene Forschungsgruppe herangezogen und zwar in der Situation, wenn ein neuer Algorithmus implementiert werden soll. Erfahrungsgemäß ist in solchen Phasen der Code häufig zu verändern, da sich erst in der Entwicklung zeigt, ob der Algorithmus verfeinert werden muss. In dieser Phase wird selten dokumentiert, auch wenn die Entwicklung anders verlaufen sollte als geplant, die Implementierung und Funktionalität stehen dagegen im Vordergrund.

Folgende gewichtete Kriterien wurden demnach als Eingabevektor eingegeben:

	Kreativität	Code- Änderungen	Qualifikation MA	Teamgröße	Erfahrung der MA
Projekt Start	1.0	0.9	0.9	0.1	0.8

Das Ergebnis mit SEN sieht folgendermaßen aus:

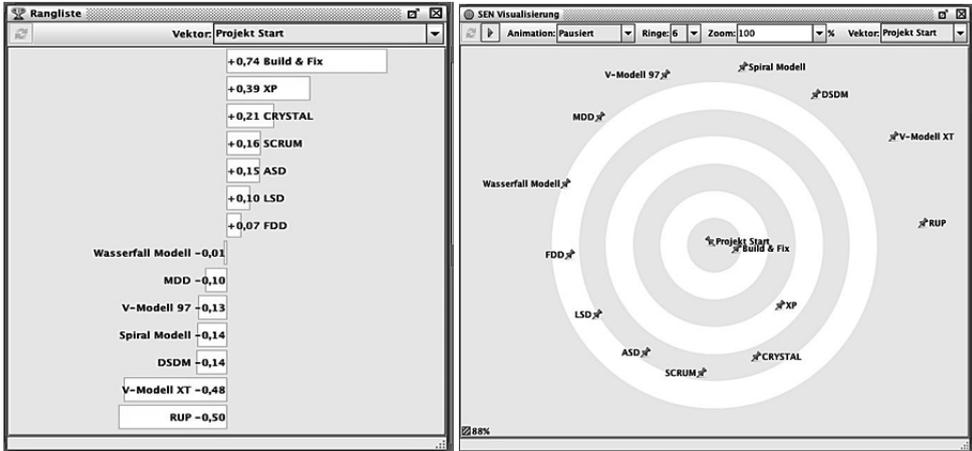


Abb. 5: Geeignete Vorgehensweise in der eigenen Forschungsgruppe, links ist die erwähnte Expertenansicht

Das Ergebnis zeigt, dass das Vorgehen eher Build & Fix entspricht, was zu Beginn eines Projektes auch zutrifft. Erst wenn der Algorithmus funktioniert und hinreichend getestet wurde, erfolgt eine Dokumentation.

Wie in Abb. 5 zu erkennen ist, werden auch negative Aktivierungswerte gezeigt. Das liegt darin, dass die Rohdaten im Wertebereich [-1,1] abgebildet werden. Dieses Vorgehen bietet sich an, wenn die semantische Matrix ebenso viele Nullen wie Einsen enthält

(s. Abb. 1). Dadurch wird vermieden, dass nur Neuronen – in diesem Fall VM – am stärksten aktiviert werden, deren Attributswerte überwiegend hoch oder eins sind.

Gerade bei Forschungsprojekten ist es natürlich denkbar, dass nach Überprüfung des Projektstandes über Variationen des benutzen VM nachgedacht werden muss. Das ist durch Eingabe neuer Kriterien in SEN problemlos möglich.

## 4.4 Hybride Modelle

Unter hybriden Modellen verstehen wir hier, wie in der Einleitung erwähnt, die Kombination von klassischen mit agilen VM [BKS14]. Eine Unterstützung der Entscheidung, welche Kombinationen für bestimmte Projekte geeignet sind, kann durch SEN beispielsweise folgendermaßen geschehen: Wenn in einem Projekt bereits ein klassisches VM wie V-Modell XT eingesetzt wird, kann SEN zeigen, welche agilen VM am besten mit V-Modell XT kombiniert werden können. Ein entsprechendes Resultat zeigt Abb. 6:

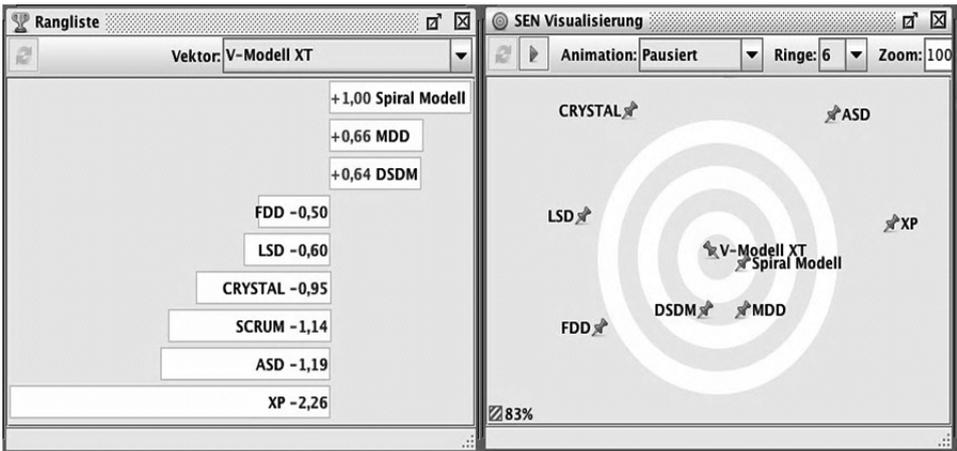


Abb. 6: Vereinbarkeit von V-Modell XT mit agilen VM

Offenbar sind für eine Kombination mit V-Modell XT das Spiralmodell und dann MDD sowie DSDM am besten geeignet.

## 5 Ausblick

SEN ist seit seiner Entwicklung mehrfach bei unterschiedlichen praktischen Problemen erfolgreich eingesetzt worden wie z.B. bei der Auswahl geeigneter Standorte für Offshore Windanlagen [KKS11], als Orientierungssystem für Studienanfänger und als Entscheidungsunterstützung für die Auswahl von Betriebsrichtungen am Flughafen Frank-

furt. Offenbar ist SEN auch sehr gut dafür geeignet, Entscheidungshilfe für Projektmanager bei der Auswahl von VM zu liefern. Ein Manager, der die Aufgabe hatte, ein geeignetes VM für eine IT-Firma und für bestimmte Projekte einzuführen, brauchte laut eigener Auskunft fünf Manntage, um eine Entscheidung zu treffen. Durch SEN wurde das ermittelte VM bestätigt und darüber hinaus wurden die nächst Platzierten als eine sinnvolle Alternative betrachtet. In dieser ersten Evaluierungsphase wurde das entwickelte SEN-Modell ebenfalls für die Meilensteinanalyse untersucht, um festzustellen, ob durch Projektänderungen andere VM geeigneter sind.

Einer der praktischen Vorteile von SEN ist die problemlose Erweiterung durch neue Vorgehensmodelle wie zum Beispiel mit Kanban, sowie die beliebig feine Differenzierung der Kriterien, nach denen die VM beurteilt werden. Darüber hinaus können Kriterien für das Tailoring [u.A. KaK13] geprüft und durch die Projektbeteiligten, je nach Projektart, bestimmt werden.

In naher Zukunft soll die Erstellung eines evaluierten semantischen Netzes durch Kooperation mit Experten und durch empirische Studien erfolgen.

## Literaturverzeichnis

- [BKS14] Berg, B.; Knott, P.; Sandhaus, G.: *Hybride Softwareentwicklung: Das Beste aus klassischen und agilen Methoden in einem Modell vereint*. Berlin Heidelberg: Springer-Verlag, 2014.
- [CH11] Chowdhury, A.F.; Huda, M.N.: *Comparison between Adaptive Software Development and Feature Driven Development*. International Conference on Computer Science and Network Technology (ICCSNT), vol.1, S. 363-367, IEEE *Xplore*, 2011.
- [DD08] Dybå, T.; Dingsøy, T.: *Empirical studies of agile software development: a systematic review*. Information and Software Technology 50 (August (9/10)), S. 833-859, 2008.
- [DD14] Diebold, P.; Dahlem, M.: *Agile Practices in practice: a mapping studies*. In: EASE '14 Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. New York: ACM, 2014.
- [FC14] Flora, H.K.; Chande, S.V.: *A Systematic Study on Agile Software Development Methodologies and Practices*. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (3), S. 3626-3637, 2014.
- [Ga08] Gadatsch, A.: *Grundkurs IT-Projektcontrolling*. Wiesbaden: Vieweg+Teubner, 2008.
- [Hi09] Hindel, B. et al.: *Basiswissen Softwaremanagement*. Heidelberg: dpunkt Verlag, 2009.
- [KaK13] Kalus, G.; Kuhrmann, M.: *Criteria for software process tailoring: a systematic review*. In: ICSSP 2013 Proceedings of the 2013 International Conference on Software and System Process. New York: ACM, S. 171-180, 2013
- [KK11] Klüver, C.; Klüver, J.: *IT Management durch KI-Methoden und andere naturanaloge Verfahren*. Wiesbaden: Vieweg Teubner, 2011.

- [KKS12] Klüver, C.; Klüver, J., Schmidt, J.: Modellierung komplexer Prozesse durch naturanaloge Verfahren. Soft Computing und verwandte Techniken. 2. erweiterte Auflage. Wiesbaden: Springer Vieweg, 2012.
- [KK13] Klüver, C.; Klüver, J.: Self-Organized Learning by Self Enforcing Networks. In (Rojas, I., Joya, G., Cabestany J. Hrsg.): Advances in Computational Intelligence. Heidelberg Dordrecht London New York: Springer, 2013.
- [KL14] Kuhrmann, M.; Linszen, O.: Welche Vorgehensmodelle nutzt Deutschland? In (Engstler, M.; Hanser, E.; Mikusz, M.; Herzwurm, G. Hrsg.): Projektmanagement und Vorgehensmodelle 2014. Soziale Aspekte und Standardisierung Lecture Notes in Informatics (LNI) - Proceedings Series of the Gesellschaft für Informatik (GI) Volume P-236 Bonne Köllen Verlag, S. 17-33, 2014.
- [WM10] Wiczorrek, H.W.; Mertens, P.: Management von IT-Projekten: Von Der Planung Zur Realisierung. Springer, London, Dordrecht, Heidelberg, New York, 2010.

## Das Business setzt die Prioritäten?!

Sixten Schockert<sup>1</sup>, Georg Herzwurm<sup>2</sup>

**Abstract:** Um zu entscheiden, welche Anforderungen im nächsten Entwicklungszyklus einer agilen Softwareentwicklung berücksichtigt werden, ist deren Priorisierung unabdingbar. Diese Priorisierung sollte gemäß dem agilen Paradigma aus Sicht des Business erfolgen. Doch allzu oft geschieht eben diese, für das weitere Vorgehen entscheidende Bewertung wenig systematisch, adhoc und vor allem ohne dezidierte Berücksichtigung der Nutzeffekte für den Kunden. Dieser Beitrag will einen Weg aufzeigen, dies zu ändern.

**Keywords:** Priorisierung von Anforderungen, Agile Softwareentwicklung, Requirements Engineering

### 1 Motivation

Innerhalb der agilen Softwareentwicklung wird in jedem Entwicklungszyklus eine abgestimmte Menge von Anforderungen in ein auslieferbares Produktinkrement umgewandelt. Um diese Anforderungen festzulegen, ist im Allgemeinen deren Auswahl aus einer größeren Menge an möglichen Anforderungen zwingend notwendig. Inhärenter Bestandteil der agilen Softwareentwicklung in Inkrementen ist demzufolge die Bewertung und Priorisierung von Anforderungen. Auf dieser Basis werden dann die wichtigsten dieser Anforderungen als konkrete, handlungsleitende Vorgaben für die Entwicklung ausgewählt. Um Unsicherheiten in Anforderungen zu reduzieren bzw. um keine unsicheren Anforderungen umzusetzen, müssen wir entscheiden, welche Anforderungen im nächsten Inkrement umgesetzt werden und welche nicht. Aber nicht nur das: von dieser Priorisierung von Anforderungen hängen essentiell die zukünftigen Entwicklungsaktivitäten ab. Fehlentscheidungen an dieser Stelle führen zu Fehlentwicklungen, die oft nur schwer zu korrigieren sind.

Dieser wichtige Schritt bei jeder agilen Entwicklung in Inkrementen – in Scrum z. B. die Priorisierung der Anforderungen aus dem Product Backlog für den nächsten Sprint [Pi08] – wird allerdings allzu oft wenig systematisch und wenig strukturiert durchgeführt. Sicherlich, durch die vermehrte Kommunikation und regelmäßige Diskussion von Anforderungen – z. B. in den Sprint-Planungs- und Reviewmeetings in Scrum – werden diese von allen Stakeholdern von Kunden- und Entwicklungsseite besser verstanden.

---

<sup>1</sup> Universität Stuttgart, BWI, Abt. VIII, LS für ABWL und WI II, Keplerstraße 17, 70174 Stuttgart und Fachhochschule für Ökonomie und Management (FOM), Rotebühlstraße 121, 70178 Stuttgart, schockert@wius.bwi.uni-stuttgart.de

<sup>2</sup> Universität Stuttgart, BWI, Abt. VIII, LS für ABWL und WI II, Keplerstraße 17, 70174 Stuttgart, herzwurm@wius.bwi.uni-stuttgart.de

Doch die Priorisierung ist schon aufgrund der unterschiedlichen Perspektiven der Beteiligten aus Technik und Business kein einfaches Unterfangen. Das agile Paradigma setzt dabei die höchste Priorität auf die Zufriedenheit der Kunden durch die frühe und kontinuierliche Auslieferung von für die Kunden „wertvoller“ Software [Be01]. Doch gerade dieser „business value“ d. h. die konkreten Nutzeffekte für den Kunden werden oft höchstens adhoc und wenig nachvollziehbar in die Priorisierung einbezogen. In vielen Fällen ist es dann doch die „große Erfahrung“ einzelner Personen, die den Bewertungsprozess bestimmen. Dieser Beitrag will einen Weg aufzeigen, gerade die Bewertung aus Kundensicht systematisch und nachvollziehbar in den agilen Entwicklungsprozess einzubeziehen.

## 2 Priorisierung bei agiler Entwicklung

Das agile Paradigma fokussiert auf die Auslieferung der wichtigsten Geschäftsanforderungen innerhalb kurzer Zeiträume von zwei Wochen bis zu einem Monat. Schnell und in regelmäßigen Abschnitten soll tatsächlich lauffähige und für die Kunden produktiv einsetzbare Software verfügbar sein. Aus diesen Gründen gibt es auch die ideale Zweiteilung im Sinne der Konzentration auf Kernkompetenzen:

- „Das Business setzt die Prioritäten“ [Co15]: Auftraggeber, Fachabteilung, Produktmanager o.ä. orientieren sich am maximalen business value für die Kunden und haben damit die finale Entscheidung über die im nächsten Entwicklungszyklus zu berücksichtigenden Anforderungen.
- Die Entwicklung entscheidet über die Art und Weise der Umsetzung der Anforderungen: Selbstorganisierende Entwicklungsteams legen das beste Vorgehen zur Auslieferung der höchstpriorisierten Anforderungen fest [Co15].

Diese Zweiteilung ist nicht gleichbedeutend damit, dass diese Aktivitäten strikt getrennt werden sollten. Es werden für jeden Zyklus konkrete Entwicklungsvorgaben benötigt und diese sollten in gemeinsamen Sitzungen der Business- und Entwicklungsseite abgeleitet und abgestimmt werden. In der Terminologie des Requirements Engineering werden diese Entwicklungsvorgaben auch als Produkthanforderungen (engl. product requirements) bezeichnet [Eb14]. Produkthanforderungen repräsentieren konkrete Eigenschaften und Fähigkeiten der Softwarelösung, die z. B. für den Mitarbeiter auf der Fachseite oder den Endbenutzer deren Probleme löst bzw. einfach Nutzen stiftet [ISO08].

**Doch wie kann das Business z. B. der Projektleiter auf der Fachseite die Priorisierung von solchen Lösungseigenschaften und –fähigkeiten leisten?** Oder konkret für Scrum formuliert: wie kann der Product Owner als Verantwortlicher für den Projekterfolg [Co15, Pi08] die Priorisierung von Lösungsfeatures aus Marktsicht leisten? Die Produkthanforderungen entstammen ja gerade nicht dem Problemraum des Business, sondern bereits dem Lösungsraum der Entwickler. Müssen sie ja auch, sonst würden sie nicht als Entwicklungsvorgaben taugen.

### 3 Dreiteilung aus Problemen, Lösungen und Anforderungen

Der Projektleiter auf der Fachseite oder auch der Product Owner bei Scrum müssen dabei unterstützt werden, die Auswahl der Entwicklungsvorgaben unter Berücksichtigung des business value, d. h. auf Basis der Nutzeffekte aus Kundensicht, nachvollziehbar für alle Stakeholder zu leisten. Wenn man sich in dieser Situation nicht auf den Erfahrungsschatz einzelner Personen verlassen will, ist es zuerst nötig sich bewusst zu machen, dass es eine **Dreiteilung aus Problemen, Lösungen und Anforderungen** gibt. Lauenroth hat dies folgendermaßen für das Requirements Engineering zusammengefasst:

„Das Requirements Engineering als gestaltende Disziplin ist dafür verantwortlich, dass mit allen relevanten Stakeholdern die Problemstellungen, die eine Software lösen soll, definiert und abgestimmt werden, Lösungen für die identifizierten Problemstellungen definiert und abgestimmt werden (sowie) Anforderungen an die Lösungen definiert und abgestimmt werden.“ [La10]

Nötig für die Priorisierung von konkreten Anforderungen aus Business-Sicht sind demzufolge vor allem das Verständnis von Problemen und möglichen Lösungen zu den Problemen. Oder anders formuliert: Produkthanforderungen können nur dann aus Business-Sicht zu Entwicklungsvorgaben priorisiert werden, wenn zuvor Probleme bzw. Kundenbedürfnisse und Lösungen definiert wurden.

Aber dazu muss zuerst einmal die explizite Trennung von Kundenbedürfnissen, Lösungen und konkreten Produkthanforderungen erfolgen, die beschriebene Dreiteilung muss nachvollzogen werden. Das ist sowohl im Requirements Engineering (RE) als auch bei agiler Softwareentwicklung nicht selbstverständlich. Beide Domänen lassen sich als Teilgebiete des umfassenderen Software Engineering auffassen in dem traditionell der Schwerpunkt auf den Produkthanforderungen im Sinne konkreter Vorgaben für die weitere Entwicklung liegt. Die Trennung von „was“ wird gefordert und „wie“ wird das „was“ umgesetzt ist zwar legendär [Da90] und viele Ansätze bemühen sich um eine Trennung wie z. B. Ebert mit der Unterscheidung von Markt-, Produkt- und Komponentenanforderungen [Eb14].

Doch die (sprich-)wörtliche Trennung reicht alleine nicht aus für die Priorisierung von Anforderungen aus Business-Sicht. Sie erlaubt zwar den Blick auf die Anforderungen aus unterschiedlichen Perspektiven wie der Entwickler- und der Marktsicht. Sie führt aber auch dazu, dass Entwickler und Kunden sich noch verstärkt mit ihren Beschreibungsmitteln für Anforderungen voneinander entfernen: Entwickler bevorzugen präzise, möglichst formale Modellbeschreibungen von Anforderungen als exakte Entwicklungsvorgaben, Kunden bevorzugen die natürlich-sprachliche Beschreibung von Anforderungen. Nicht umsonst sind textbasierte Anforderungsbeschreibungen in natürlicher Sprache mit und ohne Vorlagen weiterhin weit verbreitet [AWK13]. Entscheidend ist deswegen nicht die Trennung von Kundenbedürfnissen, Lösungen und konkreten Produkthanforderungen, sie ist nur die notwendige Bedingung. Vor allem müssen Bedürfnisse und Lösungen für die Priorisierung von Produkthanforderungen als Vorgabe für die Entwicklung wieder begründet zusammengeführt werden.

## 4 Zusammenführung von Bedürfnissen und Lösungen

Neben textbasierter Anforderungen haben sich insbesondere bei agiler Entwicklung sog. user stories als Beschreibungen für Eigenschaften und Fähigkeiten zukünftiger Softwarelösungen durchgesetzt [AWK13, Co10]. So werden user stories z. B. bei Scrum zur Formulierung der Einträge im Product Backlog eingesetzt, die der Product Owner dann für die Auswahl für den nächsten Sprint priorisiert. User stories sind dabei explizit aus der Perspektive der Benutzer bzw. Stakeholder formuliert. Und bei konsequenter Handhabung der Vorlage „als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen/Vorteil> zu erreichen“ [Co10] lässt sich mit ihnen auch eine Beziehung zwischen Bedürfnissen (= Nutzen/Vorteil) und Lösungen (= Ziel/Wunsch) abbilden. Solche „begründeten“ user stories berücksichtigen explizit den Grund für einen Lösungswunsch in der Beschreibung einer Anforderung. Sie repräsentieren damit einen Schritt in die Richtung der Zusammenführung von Bedürfnissen und Lösungen.

Aber user stories besitzen gleichzeitig auch einige Schwächen. So sind sie oft zu unbestimmt um als Entwicklungsvorgabe im Sinne einer konkreten Produkthanforderung zu dienen. Zudem bilden sie nur einfache 1:1-Beziehungen zwischen Bedürfnissen und Lösungen ab, ggf. vorhandene 1:n bzw. n:m-Beziehungen sind nur schwer bzw. kompliziert darstellbar. So werden sowohl Lösungen, die mehrere Kundenbedürfnisse befriedigen als auch mögliche alternative Lösungen für dasselbe Kundenbedürfnis nicht systematisch berücksichtigt.

Darüber hinaus geben user stories nur wenig Handlungsunterstützung bei der expliziten Priorisierung vieler Anforderungen untereinander, denn dazu müssen hier gleichzeitig viele Bedürfnisse und Lösungen miteinander verglichen werden. Die vorhandene Abgrenzung von Bedürfnissen und Lösungen wird eben gerade nicht (!) zu einer getrennten Priorisierung von Bedürfnissen und Lösungen genutzt. Eine getrennte Bewertung würde den Fokus in Richtung des Business verschieben, denn dann können die Lösungen auf Basis der wichtigsten Nutzeffekte für den Kunden ausgewählt werden. Nur dann ist die für die Formulierung von konkreten Produkthanforderungen als Entwicklungsvorgaben entscheidende Zusammenführung von priorisierten Kundenbedürfnissen mit den besten Lösungen zur Erfüllung dieser Bedürfnisse nachvollziehbar leistbar.

Ansätze basierend auf der Produktplanungsmethode Quality Function Deployment (QFD) [HSM00] können hier einen Weg weisen, denn QFD ist im Kern eine Priorisierungsmethode von Lösungen auf Basis von Kundenbedürfnissen und passt demzufolge exakt auf die beschriebene Problemstellung. QFD trennt konsequent zwischen Bedürfnissen und Lösungen, priorisiert die Bedürfnisse separiert, beurteilt die Lösungen im Hinblick auf ihre Nutzeffekte für die Kunden, ermittelt auf diese Weise die wichtigsten Lösungen aus Kundensicht, die dann weiter bewertet werden und zu konkreten Entwicklungsvorgaben verdichtet werden. Das alles geschieht mit dem einfachen Mittel einer Matrixdarstellung (Abb. 1), die im Gegensatz zu User Stories n:m-Beziehungen übersichtlich darstellt und eben auch die für die Priorisierung aus Business-Sicht notwendige Zusammenführung von Bedürfnissen und Lösungen zu Produkthanforderungen leistet.

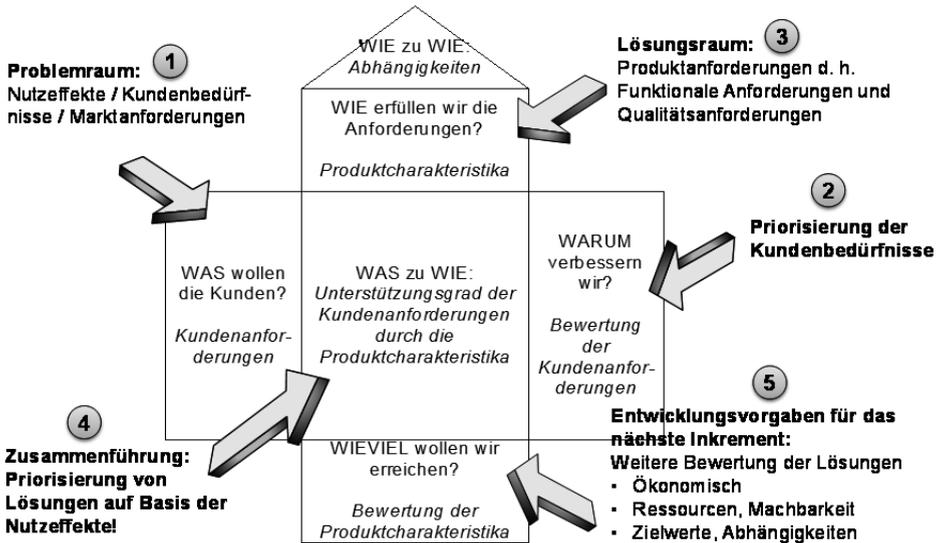


Abb. 1: Schematische Matrixdarstellung in QFD

Allerdings ist QFD im Sinne der Kundenorientierung zwar als effektiv aber vor allem auch als sehr komplex bekannt [z. B. CE94, HSM00]. Gerade die Zeit und der Aufwand zur Erstellung der Priorisierungsmatrix können bei vielen Bedürfnissen und Lösungen unverhältnismäßig im Verhältnis zum Nutzen erscheinen. Neuere methodische Ansätze in der Durchführung von QFD sind allerdings vielversprechend und fordern eine Integration mit dem Vorgehen der agilen Softwareentwicklung.

So beruht das sog. Continuous QFD (CQFD) auf inkrementellen Planungs- und Implementierungszyklen, die durch den Einsatz von ergänzenden Methoden wie dem design thinking [z. B. PML11] und Informationstechnologie zum Prototyping die möglichen Produkteigenschaften visualisieren und den Kunden damit helfen, mögliche Umsetzungen ihrer Anforderungen zu ersten Lösungen besser zu verstehen [HS14]. In CQFD werden die Priorisierungsmatrizen inkrementell bei jeder Iteration d. h. für jeden Sprint weiterentwickelt. Dabei werden in die Planungs- und Reviewmeetings die QFD-Aktivitäten der Ideensammlung/Brainstorming, des Verstehens und Sortieren/Klassifizieren von Artefakten sowie des Bewertens/Prüfens und schließlich des Entscheidens integriert. Die Priorisierung erfolgt dabei zweigeteilt: auf der einen Seite die Bedürfnisse z. B. mittels des Analytic Hierarchy Process [Sa99] aber auch einfacherer Priorisierungstechniken [z. B. HSM00, Po10]; auf der anderen Seite die Lösungen durch die Beurteilung ihrer Wirkung auf die Bedürfniserfüllung in der Priorisierungsmatrix.

Die Matrix d. h. die Zusammenführung aus Bedürfnissen und Lösungen ist dabei zu Anfang eher klein: ausgehend von den wichtigsten Bedürfnissen aus Kundensicht werden zuerst nur die Lösungen identifiziert und zu Entwicklungsvorgaben konkretisiert, die diese am besten abdecken. Mit jedem Entwicklungszyklus vergrößert sich die Matrix, es

kommen sowohl neue Bedürfnisse als auch neue Lösungen hinzu, die in die bestehende Matrixdarstellung integriert werden. Die Priorisierung der Bedürfnisse und Lösungen zu Entwicklungsvorgaben erfolgt transparent und dynamisch mit der inkrementell wachsenden Priorisierungsmatrix. Diese kann dabei auch als Kontrollinstrument in den Reviewmeetings eingesetzt werden um z. B. festzustellen, ob eine umgesetzte Lösung den gewünschten Effekt auf den Kundennutzen hatte oder nicht. Die Projektverantwortlichen wie z. B. der Product Owner bei Scrum erhalten auf diese Weise ein einfaches, für alle nachvollziehbares Planungs- und Steuerungsinstrument aus Business-Sicht in die Hand.

In CQFD aber auch in anderen QFD-Ansätzen wie in Watanabes Twin Peaks Model [WYS13] fließen überdies auch Architekturüberlegungen und mögliche Technologiekonzepte (d. h. Betriebssysteme, Datenbanken, Anwendungsserver, Softwareplattformen, wiederverwendbare Webkomponenten etc.) in die Priorisierung ein. Damit ist auch sichergestellt, dass die Bewertung nicht nur die Kundensicht sondern eben auch Realisierbarkeit, Risiko und grundsätzliche Wirtschaftlichkeitsüberlegungen berücksichtigt. QFD und insb. die Priorisierungsmatrix leisten dabei die rückverfolgbare Verbindung zwischen Problem- und Lösungsraum. Neben der iterativen Erstellung einer immer weiter wachsenden Matrix gibt es zudem die Möglichkeit gerade zu Entwicklungsbeginn oder nur für bestimmte Nutzeffekte eine Variante eines Ursache-Wirkungsdiagramms, die sog. Maximum Value Table (Abb. 2), einzusetzen [Ma12]. Dabei können allerdings n:m-Beziehungen zwischen Bedürfnissen und Lösungen nur eingeschränkt berücksichtigt werden, denn die Übersichtlichkeit würde bei zu vielen Pfeilen leiden. Trotzdem kann die Maximum Value Table als eine Art Mittelweg zwischen der einfachen user story und der umfassenderen Matrix fungieren.

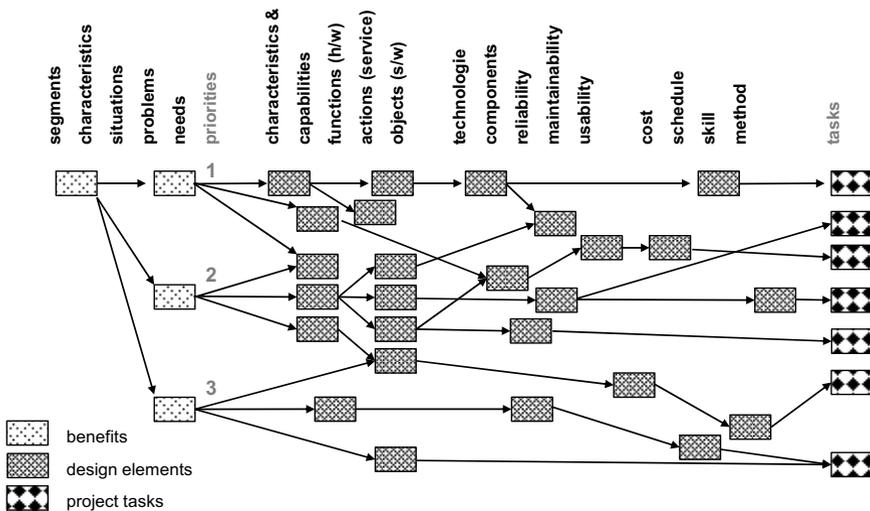


Abb. 2: Konzept der Maximum Value Table [Ma12, GDM14]

## 5 Fazit

Ziel muss es sein die Dreiteilung aus Problemen, Lösungen und Anforderungen einfach und nachvollziehbar bei allen Stakeholdern zu verankern. Dazu gehört die Trennung dieser Artefakte, aber noch viel mehr ist ihre begründete Zusammenführung für die wirkliche Bewertung aus Business-Sicht entscheidend. Nur dadurch kann die Priorisierung von Lösungen nachvollziehbar auf Basis der Bedürfnisse/Nutzeffekte erfolgen, um die wichtigsten Lösungen zu Produktanforderungen im Sinne von Entwicklungsvorgaben für das nächste Inkrement zu konkretisieren. Überdies muss dies systematisch und transparent geschehen, denn die korrekte Bewertung darf nicht von einzelnen Personen und deren Erfahrung abhängen. Die verbreitet in agiler Softwareentwicklung genutzten user stories gehen hier allerdings nicht weit genug. In Quality Function Deployment hingegen ist sowohl die Trennung als auch die Zusammenführung von Bedürfnissen und Lösungen inhärent vorhanden. Insbesondere mit Continuous QFD, der inkrementellen Matrixerstellung und der Maximum Value Table kann QFD als Priorisierungsmethode einen Weg zur Bewertung von Anforderungen aus Business-Sicht und ihrer Auswahl für den nächsten Entwicklungszyklus ebnet.

## Literaturverzeichnis

- [AWK13] Adam, S.; Wunsch, C.; Koch, M.: Ergebnisbericht RE-Kompass 2013. Fraunhofer-Institut IESE.  
[www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche\\_studien/Ergebnisbericht\\_RE-Kompass\\_2013.pdf](http://www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche_studien/Ergebnisbericht_RE-Kompass_2013.pdf), Stand: 8.9.2015.
- [Be01] Beck, K. et.al.: Prinzipien hinter dem Agilen Manifest. [www.agilemanifesto.org/iso/de/principles.html](http://www.agilemanifesto.org/iso/de/principles.html), Stand: 7.9.2015.
- [CE94] Curtius, B.; Ertürk, Ü.: QFD-Einsatz in Deutschland. In: QZ – Qualität und Zuverlässigkeit, Nr. 4, 1994, S. 394-402.
- [Co10] Cohn, M.: User Stories: für die agile Software-Entwicklung mit Scrum, XP u.a. mitp, Bonn 2010.
- [Co15] Cohn, M.: Einführung in Scrum. [www.mountaingoatsoftware.com/agile/scrum/a-reusable-scrum-presentation](http://www.mountaingoatsoftware.com/agile/scrum/a-reusable-scrum-presentation), Stand: 7.9.2015.
- [Da90] Davis, A.M.: Software Requirements. Analysis & Specification. Englewood Cliffs, New Jersey 1990.
- [Eb14] Ebert, C.: Systematisches Requirements Engineering. Anforderungen ermitteln, dokumentieren, analysieren und verwalten. dpunkt, Heidelberg 2014.
- [GDM14] Grimm, J.; Denavs, D.; Mazur, G.: Using QFD to Design a Multi-Disciplinary Clinic. In: Proceedings of North American Symposium on QFD, San Diego, USA, 2011.
- [HS14] Herzwurm, G.; Schockert, S.: QFD for Cloud Computing. In: Transactions of the 26th North American Symposium on Quality Function Deployment. Charleston, 2014, S. 54-64.

- [HSM00] Herzwurm, G.; Schockert, S.; Mellis, W.: Joint Requirements Engineering. QFD for Rapid Customer-Focused Software and Internet Development, Vieweg – Gabler, Braunschweig – Wiesbaden 2000.
- [ISO08] ISO/IEC 12207:2008: Systems and software engineering -- Software life cycle processes. 2008.
- [La15] Lauenroth, K.: Die Legende von der lösungsneutralen Anforderung oder warum das Requirements Engineering eine Gestaltungsdisziplin ist! In: Object Spectrum – Online Themenspecial Requirements Engineering 2015.
- [Ma12] Mazur, G.: Blitz QFD – The Lean Approach to Product Development. In: Proceedings of the World Conference on Quality and Improvement. Milwaukee WI, ASQ, Mai 2012.
- [Pi08] Pichler, R.: Scrum: Agiles Projektmanagement erfolgreich einsetzen. dpunkt, Heidelberg 2008.
- [Po10] Pohl, K.: Requirements Engineering. Springer, Berlin 2010.
- [PML11] Plattner, H.; Meinel, C.; Leifer, L.: Design Thinking – Understand, Improve, Apply, Springer, Heidelberg 2011.
- [Sa99] Saaty, T. L.: Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World. 3. Aufl., Pittsburgh 1999.
- [WYS13] Watanabe, Y.; Yoshikawa, M.; Shindo, H.: Software development method based on twin peaks model with QFD. In: Proceedings of the 19th International Symposium on QFD & 22nd North American Symposium on QFD 2013, Santa Fe, USA, S. 117-126.

# agile@scale: Do more with LeSS or be SAFe?

## Ansätze zur Skalierung – ein Überblick

Patrick Daut<sup>1</sup>

**Abstract:** Als Framework für agile Entwicklung hat vor allem Scrum in großer Verbreitung Anwendung gefunden. Gerade kleine und mittelständische Unternehmen in technologie-nahen Märkten profitieren von verkürzter Time-to-Market und hoher Flexibilität gegenüber Kundenanforderungen und der Variabilität des Marktes. Etablierte Konzerne versuchen, es ihnen gleichzutun, Flexibilität und Innovationskraft zu gewinnen und ihre Marktanteile zu sichern. Nun beruht Scrum aber auf der Arbeit eines kleinen Teams. Es liefert zunächst keine Lösungen für den Einsatz in einer großen Organisation mit einer Vielzahl von Teams. Aus diesem Bedarf heraus sind in den letzten Jahren verschiedene Ansätze zur Skalierung agile Vorgehens entstanden: Frameworks wie das Scaled Agile Framework (SAFe) oder Large-Scale Scrum (LeSS), Ansätze wie der Scaling Cycle und populäre Umsetzungen in der Praxis wie bei Spotify. Worauf beruhen diese Ansätze, was ist ihnen gemein und was sind die grundlegenden Unterschiede? Woran kann ich mich bei der Auswahl eines Vorgehens orientieren? Im Rahmen dieses Beitrags wird ein Überblick über die wachsende Landschaft der Skalierungsansätze gegeben.

**Keywords:** Scrum, agile, lean, scaled agile, agile@scale, skalierung

### 1 Aus dem praktischen Bedarf heraus sind verschiedene agile Skalierungsframeworks entstanden

Agile Produktentwicklung verspricht Schnelligkeit, Flexibilität und Qualität. In großen Organisationen beschränkt sich die Implementierung agiler Methoden jedoch oftmals noch auf Softwareentwicklung und -test. Oft sind schon Produktmanagement und das obere Management kaum eingebunden; klassisches Anforderungsmanagement ist der Entwicklung vorgelagert und beschränkt die Möglichkeit, Nutzen aus agilen Vorgehensweisen in der Entwicklung zu ziehen. Analog bestehen Schranken auch in Richtung klassischer Unternehmensbereiche wie Finance, Budgeting Controlling etc. und auch interner wie externer Stakeholder. Des Weiteren ist die Koordination mehrerer paralleler Entwicklungsteams, die ab einer gewissen Organisationsgröße nötig wird – nur unzureichend gelöst.

In den vergangenen Jahren haben sich verschiedene methodische Ansätze und Frameworks entwickelt, die die Skalierung agilen Vorgehens in großen Organisationen zu lösen versuchen. Wie unterscheiden sich diese und welches ist das Richtige für mich?

---

<sup>1</sup> Cassini Consulting, Cassini Consulting Nord GmbH, Johannissbollwerk 16, 20459 Hamburg patrick.daut@cassini.de

## 2 Bei der Wahl eines Frameworks sind zwei Dimensionen ausschlaggebend

Die hier betrachteten Frameworks unterscheiden sich zum Teil erheblich in Ansatz, Ausgestaltung und im Grad der spezifischen Beschreibung. Bemerkenswert ist aber, dass sie alle auf ähnlichen Ideen und Prinzipien – letztlich aus dem Bereich des Lean Thinking – beruhen. Lean Thinking wird hier verstanden als konzeptionelle Idee hinter Lean Production und dem Toyota Production System, wie Womack und Jones<sup>2</sup> sie herausgearbeitet haben in einer generellen und auch auf Wissensarbeit anwendbaren Weise: Fokussierung auf Werthaltigkeit, Wertströme, Flussorientierung, Pull-Prinzip, Qualität – basierend auf einem hohen Stellenwert des Menschen.

Daneben weisen auf den ersten Blick ganz verschiedene Ansätze große Ähnlichkeiten in den durch sie definierten grundlegenden Organisationsstrukturen auf. Steht man vor der Aufgabe der Auswahl eines der Ansätze für den Einsatz in einer Organisation, sind die folgenden beiden Dimensionen ausschlaggebend:

- Größe der Entwicklungsorganisation
- Komplexität des Produktes und Marktes sowie die Größe der Gesamtorganisation

Die These ist, dass es sich bei diesen beiden Aspekten um die wesentlichen Einflussfaktoren auf den Bedarf an Steuerungs- und Kontrollmechanismen in Aufbau- und Ablauforganisation handelt.

## 3 Was bietet der Markt?

Bevor wir uns dem Vergleich der Frameworks und einer Hilfestellung für eine Auswahl widmen, soll im Folgenden ein Kurzüberblick über die betrachteten Ansätze gegeben werden:

**Das Scaled Agile Framework (SAFe)**<sup>3</sup> ist ein komplexes Framework zur Einbettung agiler Entwicklung in einer großen Organisation mit Einbindung der klassischen Unternehmensbereiche. Es ist komplex in dem Sinne, dass es die folgenden Bestandteile definiert:

- Eine Aufbauorganisation in Form von Strukturen, Rollen und deren Abhängigkeiten
- Ein Prozess mit Artefakten, die zu definierten Zeitpunkten erstellt werden
- Methoden und Praktiken

---

<sup>2</sup> S. [WJ03] und [WJR07]. Vgl. auch das Konzept des *Lean Managements*.

<sup>3</sup> S. [Le15] sowie [Le07], [Le10].

Kern ist eine über die Zeit stabile Organisationsstruktur, die die Arbeit mehrerer an einem (Teil-)Produkt beteiligten Teams zusammenhält und koordinative Rollen definiert. Die Teams selbst folgen einer erweiterten, angepassten Form von Scrum

**Large-Scale Scrum (LeSS)**<sup>4</sup> definiert – vereinfacht ausgedrückt – zwei Aspekte:

- Eine einfache Struktur, die zum größten Teil auf Scrum Teams sowie einer Hierarchie aus Product Ownern besteht, wobei der Verantwortungsbereich eines Product Owners auf tieferen Hierarchieebenen kleiner wird
- Einen einfachen – ebenfalls kaum verändert auf Scrum basierenden Ablauf pro Team mit einzelnen Koordinationsmechanismen zwischen den Teams wie Scrum-of-Scrum oder teamübergreifenden Retrospektiven

Der **Agile Scaling Cycle**<sup>5</sup> definiert im Gegensatz zu SAFe und LeSS weniger ein Framework für Aufbau- und Ablauforganisation als ein Vorgehen zur Entwicklung einer eben solchen Organisationsform für die jeweilige Umgebung. Er schlägt vor, zunächst mit der Minimierung von Abhängigkeiten zu beginnen, im zweiten Schritt mit den verbleibenden Abhängigkeiten in Form von Koordinationsmechanismen zwischen den Teams umzugehen um letztlich die Organisation weiterzuentwickeln, die die aufgetretenen Probleme dauerhaft löst. Anschließend beginnt der Zyklus von vorn. Hier kann – was den Ansatz angeht – ein Vergleich zu Kanban gezogen werden: Kanban beschreibt ebenfalls Schritte zur Definition eines situationsgerechten, individuellen Prozesses vorschlägt – nicht aber den Prozess selbst.

Bei der **Spotify Culture**<sup>6</sup> handelt es sich um eine Beschreibung der Organisationsstruktur der Produktentwicklung bei Spotify mit einem Fokus auf der Aufbauorganisation. Es kann also weniger von einem Framework als von einem Praxisbeispiel oder einer Best Practice gesprochen werden: Ein gelebtes Beispiel einer leichtgewichtigen Organisationsform mit Fokus auf Strukturierung und Koordination von Teams. Auffälliges Merkmal der bei Spotify gelebten Struktur ist eine „Tribe“ genannte stabile Struktur, die mehrere agile Teams<sup>7</sup> zusammenfasst und als koordinierende Organisationsform dient.

SAFe ist durch mehrere Bücher sowie eine inhaltsreiche Website beschrieben. Eine inzwischen starke Community wird aktiv betreut. Daneben existiert ein Trainings- und Zertifizierungsprogramm. Zusammen ergibt sich der Eindruck eines aktiv vermarkteten Produkts. LeSS basiert auf der Arbeit der beiden Autoren, zunächst in Form zweier Bücher. Auch für LeSS ist inzwischen eine inhaltgetriebene Website sowie ein Trainings- und Zertifizierungsprogramm verfügbar. Der Scaling Cycle hingegen wird in einzelnen Blogposts und Vorträgen beschrieben. Ähnliches gilt für die Spotify Culture von Kniberg – dessen Beiträge allerdings sehr populär geworden sind.

---

<sup>4</sup> S. [LV15] sowie [LV08], [LV10].

<sup>5</sup> S. [Ro14a], [Ro14b], [RW14].

<sup>6</sup> S. [Kn14a], [Kn14b].

<sup>7</sup> In der von Kniberg vorgestellten Struktur werden die agilen Teams „Squads“ genannt.

Von SAFe und LeSS sind jeweils Sonderformen in der Entstehung oder veröffentlicht: LeSS Hufe sowie SAFe for Systems Engineering. Beides fließt in die vorliegende Betrachtung nicht dediziert ein.

#### 4 Wie unterscheiden sich die Frameworks?

Bereits die Kurzvorstellung hat gezeigt, dass sich die Ansätze teils stark unterscheiden. Für die Auswahl eines Ansatzes für den Einsatz in einer konkreten Organisation ist eine detailliertere Gegenüberstellung vonnöten. Tab. 1 zeigt die Ergebnisse hinsichtlich ausgewählter Aspekte.

Aspekt	SAFe	LeSS	Scaling Cycle	Spotify Culture
<b>Ansatz</b>	Komplexe Struktur und Vorgehensweise über 3 Ebenen	Erweiterung von Scrum um einzelne Elemente und Koordinationsmechanismen Sammlung einzelner Praktiken	Vorgehen zu Entwicklung einer individuellen Organisationsform	Vorstellung einer in der Praxis gelebten Organisationsstruktur und -kultur
<b>Basis<sup>8</sup></b>	Feature-Teams, die im Wesentlichen nach Scrum arbeiten	Feature-Teams	Unabhängige Teams	Squads (Unabhängige, cross-funktionale Teams)
<b>Hierarchie/ Linie<sup>9</sup></b>	Definierte Rollen mit jeweiligem Verantwortungsbereich	Hierarchie aus Entwicklung, Chief Product Owner, Area Product Owner und Product Owner	Für die jeweilige Organisation auszugestalten	Chapters

<sup>8</sup> Welches sind die Organisationseinheiten an der Basis der Struktur?

<sup>9</sup> Wird eine klassische Organisationsstruktur bzw. Organisationshierarchie durch das Framework mit abgebildet?

<b>Aggregati- onsebene<sup>10</sup></b> (Teil- produkt,...)	<i>Agile Release Train</i> : stabile, langfristige angelegte Aggregation von Teams (100-120 Personen, Dunbar-Number)	Area; fachliche und/oder architekturelle Zerlegung		Tribes (100-120 Personen, Dunbar-Number)
<b>Product Owner</b>	Ein Product Owner pro Team zzgl. Koordinieren der Rollen auf Programm- und Portfolioebene	Team von Product Ownern, ggf. Area Product Owner	Ein Product Owner; Teams besitzen selbst die Fähigkeit, grobe Features zu zerlegen	Ein Product Owner pro Squad
<b>Quer- schnitts- funktionen</b>	Definierte Rollen für Funktionen wie Architektur, UX, etc, die koordinative und steuernde Funktion haben aber nicht die Verantwortung aus den Teams nehmen	Communities of Practice	Vorschlag: Communities of Practice	Guilds (und Chapters)
<b>Komplexitäts- bewältigung</b>	Rollen und Koordinationsmechanismen zum Umgang mit/Management von Komple-	Reduzierung von Abhängigkeiten, um unabhängige Teams mit der Möglichkeit zur Selbstorga-	Reduzierung von Abhängigkeiten, um unabhängige Teams mit der Möglichkeit zur Selbstorganisation zu schaffen	Keine Beschreibung der Auflösung/des Managements von Abhängigkeiten innerhalb des Tribes. Ziel ist Minimierung der

<sup>10</sup> Welche Struktur oder welches Mechanismus sieht das Framework vor, um mehrere Teams hinsichtlich einen aggregierten gemeinsamen Ergebnisses zu koordinieren?

xität sowie auch zu des- sen Verringe- rung (Bsp.: Identifikation von Abhän- gigkeiten zwischen Teams im gemeinsamen Planning)	nisation zu schaffen; aber auch Einsatz einzelner Me- chanismen zum Umgang mit bestehenden Abhängigkei- ten	Abhängigkeiten zwischen Tribes. Mechanismen zu deren Identifikati- on und Ma- nagement
--	---	---

Tab. 1: Gegenüberstellung der Frameworks

Zum einen sind grundlegende Unterschiede festzustellen. Beispielsweise wird eine Aggregationsebene zur Koordination mehrerer Teams teils sehr verschieden gebildet. Auch wird die Einführung einer hierarchischen Organisationsstruktur – letztlich zum Zweck der Delegation - unterschiedlich gelöst. Andererseits finden sich Ideen wie Communities of Practice zur fachlichen Koordination über Teamgrenzen hinweg in fast allen der Lösungsansätze wieder. Ähnliches gilt für die Betonung der Bedeutung von unabhängigen Feature-Teams für die Skalierung. Hintergrund hier ist die Minimierung von Abhängigkeiten und somit eines minimalen Bedarfs an Koordinationsmechanismen über Teams hinweg.

Bemerkenswert ist, dass ein für seine komplexe Spezifikation teils in der Literatur kritisiertes Framework wie SAFe gerade hinsichtlich der definierten Organisationsstrukturen Ähnlichkeit zu denen des Spotify Modells aufweist.

## 5 Welcher Ansatz eignet sich in welcher Umgebung?

Kommen wir nach diesem ersten Vergleich der Ansätze zur eingangs gestellten Fragestellung zurück: Was kann mir bei der Auswahl eines Frameworks als Orientierung dienen?

Der Vergleich der Ansätze zeigt Unterschiede vor allem in Komplexität der Frameworks, Detailgrad der definierten Strukturen, Umgang mit Komplexität sowie teamübergreifende Koordinationsmechanismen. Im Rahmen dieses Beitrags wird vorgeschlagen, dass eine Wahl von den folgenden Dimensionen und Kriterien abhängen sollte:

### Produkt, Markt und Umgebung

- Welche Kontrollinstanzen habe ich? Bewege ich mich in einem regulierten Bereich oder liegen besondere regulatorische Anforderungen vor?
- Ist mein Geschäft eher produkt- oder projektorientiert?

- Wie komplex sind Produkte, Projekte oder Aufgaben? Wie groß sind Unsicherheiten?
- Wie groß ist die Gesamtunternehmung?

### Organisation

- Wie groß ist die Entwicklungsorganisation?
- Wie ist das Unternehmen aufgebaut? Besteht Bedarf der Einbettung einer agilen Entwicklung in das Gesamtunternehmen und in klassische Unternehmensfunktionen? Welche Möglichkeit der Anpassung besteht dort?
- In welchem Maß besteht bereits Erfahrung mit agilen Methoden? Ist dies für das ganze Unternehmen neu? Gab es fehlgeschlagene Versuche? Oder ist die Entwicklung agil erfolgreich und es fehlt noch an den Schnittstellen?

Abb. 1 zeigt eine mögliche Einordnung der untersuchten Frameworks hinsichtlich der beiden beschriebenen Dimensionen.

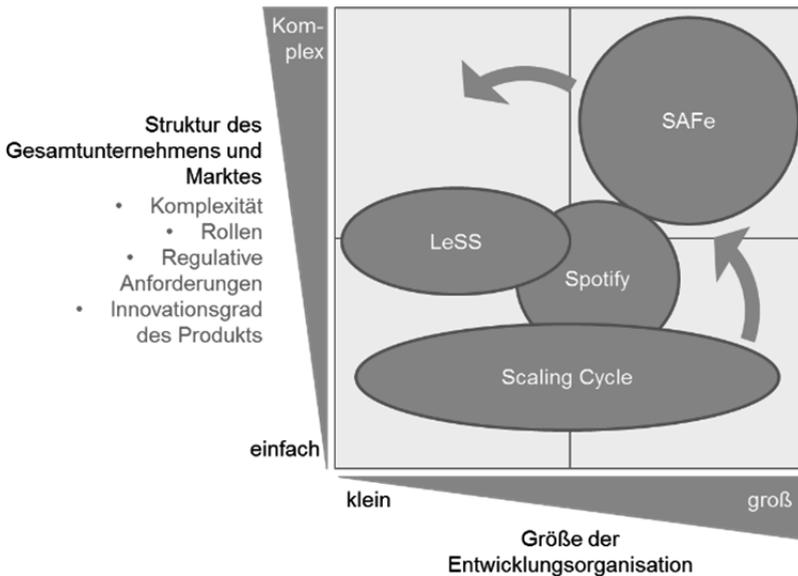


Abb. 1: Systematik der Eignung der untersuchten Frameworks

Ein Ansatz wie der Scaling Cycle kann unabhängig von der Größe der Entwicklungsorganisation eingesetzt werden – er selbst definiert noch keine feste Struktur und keinen Prozess. Mit zunehmender Komplexität der Gesamtunternehmung, des Produktes und des Marktes bringen diejenigen Ansätze Vorteile mit sich, die bereits konkrete Lösungen liefern. Diese Vorteile kommen bei wenig Erfahrung mit agilem Vorgehen in der Organisation noch mehr zum Tragen. Die durch eine solche Komplexität induzierte Vielzahl

an Rollen im Unternehmen, an internen und externen Stakeholdern sowie etwaigen Reglementierungen erfordern konkrete Lösungen. Während der am Beispiel von Spotify beschriebene Ansatz eine für mittelgroße Unternehmen durchaus geeignete Aufbauorganisation definiert, bringt er nur wenig prozessuale Unterstützung mit sich. LeSS wiederum definiert ein einfaches, kaum von Scrum abweichendes Vorgehen sowie eine skalierbare Struktur. Der Fokus auf eine Hierarchie aus Chief Product Owner, Area Product Owner sowie Product Owner auf Teamebene wird jedoch ab einer gewissen Größenordnung auf Grenzen stoßen: Der Kommunikations- und Koordinationsbedarf wird in dieser Struktur stark ansteigen. LeSS betont, dass der Schnitt der Anforderungen wie auch der Teams entlang kundenrelevanter Features und nicht technische Module oder Schichten entscheidend ist. Die dadurch angestrebte Unabhängigkeit der Anforderungen untereinander und Teams voneinander ist ab einer gewissen Größe aber nicht mehr ausreichend, um den Koordinationsbedarf zu begrenzen. Product Owner auf den unteren Hierarchieebenen tragen kaum noch Produktverantwortung. SAFe hingegen ist in seiner definierten Form für kleinere Unternehmen eher zu mächtig, bringt aber eine Vielzahl konkreterer Lösungen auf prozessualer Ebene sowie eine skalierungsfähige Aufbauorganisation mit, die die Koordination einer hohen Zahl an Teams erlaubt sowie verschiedene Stakeholder und klassische Unternehmensbereiche einzubinden vermag.

Während der Scaling Cycle durch seine Anwendung durchaus erlaubt, funktionierende Lösungen auch für größere Unternehmen zu entwickeln, so zeigt die Projekterfahrung, dass auf Basis von SAFe einfachere Strukturen auch für mittelgroße Organisationen entwickelt werden können.

## Literaturverzeichnis

- [Kn14a] Kniberg, H.: Spotify Engineering Culture, Blogeintrag auf [blog.crisp.se](http://blog.crisp.se), <http://blog.crisp.se/2014/03/27/henrikkniberg/spotify-engineering-culture-part-1>, 5.8.2015.
- [Kn14b] Kniberg, H.: Spotify Engineering Culture, Blogeintrag auf [labs.spotify.com](https://labs.spotify.com), <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1>, 5.8.2015.
- [LV15] Large-Scale Scrum Website, <https://less.works/de/less/framework/index.html>, 5.8.2015
- [Le07] Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Addison Wesley, 2007.
- [Le10] Leffingwell, D.: Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison Wesley, 2010.
- [LV07] Larman, C.; Vodde, B.: Practices for Scaling Lean and Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum. Addison Wesley, 2010.
- [LV10] Larman, C.; Vodde, B.: Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum: Successful Large, Multisite and Offshore Products with Large-scale Scrum. Addison Wesley, 2010.

- [Le15] Scaled Agile Framework Website, <http://www.scaledagileframework.com>, 5.8.2015
- [Ro14a] Roock, S.: Folien eines Vortrags auf dem Agile Leadership Day 2014, Slideshare, <http://de.slideshare.net/roock/agile-scaling-cycle-lightning-talk-at-agile-leadership-day-2014-zurich>, 5.8.2015.
- [Ro14b] Roock, S.: Agile Scaling Cycle: from SCaLeD principles to practices, Blogbeitrag, <https://stefanroock.wordpress.com/2014/07/13/agile-scaling-cycle-from-scaled-principles-to-practices>, 5.8.2015.
- [RW14] Roock, S.; Wolf, H.: Agile Skalierung - Prinzipien statt Blueprint, Folien eines Vortrags auf der OOP 2014, Slideshare, 5.8.2015.
- [WJ03] Womack, J.; Jones, D.: Lean Thinking: Banish Waste And Create Wealth In Your Corporation. Simon & Schuster UK, New Edition, 2003.
- [WJR07] Womack, J.; Jones, D., Roos, D.: The Machine That Changed the World: The Story of Lean Production - Toyota's Secret Weapon in the Global Car Wars That Is Now Revolutionizing World Industry. Free Press, Reprint, 2007.



# Softwareentwicklung im Maschinenbau – ein kooperativer Ansatz

Martin Jud<sup>1</sup>, Jörg Hofstetter<sup>2</sup>

**Abstract:** Kleine- und mittelständische Maschinen- und Anlagenbauer haben oft keine eigene Software-Entwicklungsabteilung. Bei Neu- oder Weiterentwicklungen beauftragen sie externe Softwaredienstleister. Dabei wird regelmäßig die Komplexität solcher hybride Projektstrukturen unterschätzt. Mangelnde Controlling-Möglichkeiten insbesondere auf Softwareseite erschweren die Steuerbarkeit der gesamten Entwicklung und können zu hohen Kosten führen. In einem Forschungsprojekt konnten wir zeigen, wie ein innovatives interdisziplinäres Kooperationsmodell im Projektmanagement Abhilfe schafft. Als Basis dient das hybride Projekt- und Vorgehensmodell SoDa, welches an der Hochschule Luzern in Lehre und Forschung eingesetzt wird. Eine Gliederung in Entwicklungsabschnitte, an deren Ende Soft- und Hardware integriert werden, ermöglicht ein effektives Controlling.

**Keywords:** Hardware-/Software Co-Entwicklung, Projekt-Controlling in hybriden Projektstrukturen, Vorgehensmodelle im interdisziplinären Umfeld, Softwareentwicklungs-Verträge

## 1 Einleitung

An der 20. Tagung in Lörrach haben wir unter dem Titel „Was fehlt Scrum?“ unser Vorgehensmodell für Forschung und Lehre an der Hochschule Luzern HSLU vorgestellt [Hof13]. Die Integration eines Scrum-artigen Vorgehens in einen Projektmanagement-Gesamtrahmen war schon damals einer der Gründe ein eigenes Vorgehensmodell für Forschung und Lehre zu entwickeln.

SoDa (kurz für 'Software Development agile') ist ein einfaches Projekt- und Vorgehensmodell für Lehre, Forschung und Industrie in der Informatik [Jud14]. Im Fokus sind Software-Entwicklungsvorhaben mit kleineren Teams über einige Monate. SoDa basiert auf dem Vorgehensmodell Scrum und bindet dieses in einen Projektmanagement-Rahmen ein. Damit wird dem Bedürfnis von Auftraggebern und übergeordneter Organisation nach geordneter Projektdurchführung mit standardisierten Phasen und Meilensteinen Rechnung getragen und die Anschlussfähigkeit von Informatik-Teams in interdisziplinären Projekten wie Software- / Hardware-Co-Entwicklungen unterstützt.

---

<sup>1</sup> Hochschule Luzern, Informatik, Technikumstr. 21, CH-6048 Horw, martin.jud@hslu.ch

<sup>2</sup> Hochschule Luzern, Informatik, Technikumstr. 21, CH-6048 Horw, joerg.hofstetter@hslu.ch

In der Zwischenzeit können wir sagen, dass sich unser hybrides Vorgehensmodell im Hochschulalltag bewährt. Ein von der Kommission für Technologie und Innovation KTI, der Schweizerischen Förderagentur für Innovation unterstütztes Forschungsprojekt gab uns Gelegenheit anspruchsvolle Entwicklungsprojekte in der Industrie zu begleiten und so unser Modell weiter zu entwickeln.

In diesem Beitrag legen wir den Fokus auf die Co-Entwicklung, d.h. wir untersuchen, wie eine erfolgreiche Zusammenarbeit zwischen Maschinenbauern und Softwareengineering Unternehmen gestaltet werden kann auch wenn zum Auftragszeitpunkt noch gar nicht vollständig feststeht, wie die neue Maschine gesteuert werden muss. (Was bei der Neuentwicklung einer Maschine zwangsläufig der Fall sein wird, denn auch die mechanischen Komponenten sind ja noch nicht fertig konstruiert und hergestellt.)

## 2 Ausgangslage und Herausforderungen

Viele Maschinen- und Anlagebauer sind KMU und haben keine eigene Software-Entwicklungsabteilung. Für die Entwicklungen neuer Produkte, aber auch für die Weiterentwicklung bestehender Produkte werden fallweise externe Software-Entwickler oder auf Softwareentwicklung spezialisierte Engineering-Firmen beigezogen.

Ausschlaggebend für das KTI Projekt war eine Anfrage der Firma PROFIN Progressive Finish AG, die seit über 40 Jahren in der Oberflächen- und Entgrattechnologie tätig ist. Mit Maschinen, Prozesstechnologien und Dienstleistungen bedient das Unternehmen als technologischer Marktleader eine vielseitige Kundschaft. Unter anderem ist man Alleinanbieter des neu entwickelten Prozesses «Flakkotieren».

PROFIN hat bei der Entwicklung der letzten Maschinengenerationen bei Vergabe, Entwicklung und Pflege der Software schlechte Erfahrungen gemacht: Es mussten massive Verspätungen und daher ein Mehraufwand von über 1 Mio. Franken in Kauf genommen werden.

Wir hatten hier die Gelegenheit unsere theoretischen Überlegungen laufend bei einem Maschinenbauunternehmen zu überprüfen und zu sehen, was praxistauglich ist. PROFIN hat das interne Verständnis für Softwareentwicklungsprozesse breiter abgestützt und führt heute Co-Entwicklungen effizienter zum Erfolg.

Einerseits auf der Basis dieses Forschungsprojektes, welches auch eine projektinterne Validierung mit weiteren Industriepartnern umfasste und andererseits aufgrund unserer eigenen langjährigen Praxiserfahrung sind wir überzeugt, dass der hier vorgestellte Ansatz in der Branche allgemein dazu beitragen kann die Innovation mit beherrschbarem Risiko erfolgreich voran zu treiben.

## 2.1 Wissensdefizite und Komplexität

Der Versuch einfach Softwareentwicklungskapazität zuzukaufen führt bei vielen Entwicklungen zu ernststen Problemen. Massive Zeit- und Kostenüberschreitungen sind die Folge. Auftraggeber und Auftragnehmer fühlen sich gegenseitig über den Tisch gezogen, denn in dieser Konstellation ergeben sich in der Regel zwei nicht triviale Herausforderungen für die Beteiligten: Wissensdefizite und Komplexität. Wer sich diesen Herausforderungen erfolgreich stellen will, muss sein Projektmanagement und Vorgehen entsprechend anpassen.

### Wissensdefizite

Der Maschinen- bzw. Anlagebauer hat in der Regel ein gutes Domänenwissen nicht nur im Maschinenbau selbst, sondern insbesondere auch im Anwendungsgebiet seiner Anlagen, aber er hat kaum eigenes Know-How, was die Methoden und Vorgehensweisen der Software-Entwicklung angeht.

Software-Engineering Firmen haben in der Regel ihre Fachkompetenz primär im Bereich des Software-Engineering. Das Domänenwissen in den oft verschiedenen Anwendungsgebieten ist aber eher oberflächlich. Insbesondere im spezifischen Anwendungsgebiet der Maschinen fehlt das Wissen meist.

### Komplexität

Die Entwicklung von Maschinen und Anlagen ist nicht einfach nur anspruchsvoll und kompliziert, sie ist wegen der Anzahl der Einflussfaktoren und dem Ausmaß ihrer gegenseitigen Abhängigkeiten komplex.

Nach Funke zeichnet sich Komplexität durch große Anzahl und Vernetzung von Variablen, sich teilweise widersprechende Ziele und Intransparenz der Abhängigkeiten aus [Fun06]. Diese Punkte treffen alle auf den interdisziplinären Systementwurf zu, wie er z.B. bei der Entwicklung einer neuen Maschine oder Anlage gemacht werden muss. Parallele Entwicklung von Maschine und Software ist also komplex.

Das CYNEFIN Strategiemodell [Sno00] liefert Anhaltspunkte, welche Typen von Problemstellungen in welcher Art beherrscht werden können: Nach diesem Modell ist das klassische ingenieurmäßige Vorgehen (erkennen > analysieren > reagieren) bei komplexen Aufgaben nicht zielführend. Es eignet sich für komplizierte Probleme, aber eben nicht für die bei der Entwicklung von Maschinen und Anlagen typischen komplexen Problemstellungen. Hier ist die erfolgversprechendste Lösungsstrategie gemäß dem CYNEFIN Modell probieren > erkennen > reagieren.

Der Architekt Richard MacCormac hat schon in den 70er Jahren erkannt, dass komplexe Probleme sich nicht linear durch Sammeln und Zusammenführen von Informationen lösen lassen [McC76]. Er sagt:

“I don’t think you can design anything just by absorbing information and then hoping to synthesize it into a solution. *What you need to know about the problem only becomes apparent as you’re trying to solve it.*”

Zu Beginn einer Maschinen-Entwicklung ist nicht klar, welches die relevanten Informationen zur Lösung der komplexen Aufgabe sind. Der Lösungsweg wird wesentlich durch die auf diesem Weg selbst gewonnenen Einsichten mit bestimmt.

## 2.2 Annahmen und Erwartungen

Die Voraussetzungen für einen konventionellen Softwareentwicklungsauftrag sind nach obigen Erkenntnissen nicht gegeben: Beide Seiten begeben sich in eine Rolle, die sie nicht auszufüllen vermögen und haben an die jeweils andere Seite Erwartungen, denen diese wiederum nicht gerecht werden kann. So sind Probleme und Missverständnisse vorprogrammiert.

Da im Vorfeld weder die konkrete Lösung noch der dazu nötige Entwicklungsweg im Detail bekannt sind und beides natürlich auch von den Partnern und der Qualität ihrer Zusammenarbeit wesentlich abhängt, ist eine detaillierte Planung und Offerstellung gar nicht möglich. Die benötigte Funktionalität kann erst grob skizziert werden, entsprechend lässt sich der Realisierungsaufwand nur grob abschätzen. Lastenheft und Planung müssen entsprechend dem Wissenszuwachs im Projektverlauf nachgeführt werden.

In einem solchen Entwicklungsprojekt können weder Maschinenbauer noch Softwareentwickler alleine wissen, ob ihre Entwurfsentscheidungen Konsequenzen für die jeweils andere Domäne haben. Die Intransparenz bezüglich der beteiligten Variablen und deren gegenseitige Abhängigkeiten verbietet eine zu hohe Arbeitsteilung in der Lösungsfindung und im Entwurfsprozess.

## 3 Co-Entwicklung

Wer die Ausgangslage bei Vergabe eines Softwareentwicklungs-Auftrages genauer untersucht und die oben gemachten Überlegungen mit einbezieht, versteht, dass die Entwicklung der Software nicht einfach als Auftrag „abgeschoben“ werden kann. Vielmehr wird offensichtlich, dass ein erfolgversprechendes und zielführendes Vorgehen eine enge Kooperation zwischen Maschinenbau- und Software-Domäne erfordert. Projektmanagement und Entwicklungsvorgehen müssen entsprechend angepasst werden.

Erfolgversprechend ist daher eher eine enge Zusammenarbeit zwischen Maschinenbauer und Software-Engineering Unternehmen, denn wer was macht hängt auch stark vom Kenntnisstand der Maschinenbauer und Softwareentwickler in den verschiedenen Wissensdomänen ab. Wir sprechen deshalb von Co-Entwicklung, für die es einen geeigneten Partner, ein angemessenes Projektmanagement und Vorgehen sowie der spezifischen Situation angepasste Verträge braucht.

### 3.1 **Gemeinsames Verständnis**

Eine Co-Entwicklung, das heißt ein gemeinsames Erarbeiten der konkreten Lösung ist der angemessene Umgang mit der vorliegenden Komplexität. Eine Co-Entwicklung macht aus Auftraggeber und Auftragnehmer auf inhaltlicher Ebene Partner auf Augenhöhe. Dabei werden sich die gegenseitigen Wissensdefizite wie oben aufgezeigt bemerkbar machen. Nach unserer Erfahrung und unseren Erkenntnissen braucht es

- Ein minimales Know-How auf Seite des Maschinen- bzw. Anlagebauers, was die Methoden und Vorgehensweisen der Software-Entwicklung angeht.
- Ein minimales Domänenwissen auf Seiten des Softwareentwicklers im spezifischen Anwendungsgebiet ihres Auftraggebers.
- Ein gemeinsames Projekt-Controlling um die Zusammenarbeit und Kommunikation der Projektpartner zu unterstützen und zu verbessern.

### 3.2 **Kommunikation und Vertrauen**

Bei einer Co-Entwicklung kommt dem Vertrauen zwischen den Projektpartnern eine zentrale Bedeutung zu. Aber gerade in der Situation zwischen Co-Entwicklungspartnern ist Vertrauen nicht selbstverständlich, denn die Partner kommen aus unterschiedlichen Kulturen (z.B. Maschinen- / Anlagenbau und Informatik). Die darin begründete potentielle Diskordanz von Werten, Abläufen und Begriffen ist vielfach der Ausgangspunkt einer negativen Vertrauensentwicklung.

Betriebliche Organisationen als Gesamtheit können nicht vertrauen, Vertrauen kann nur von einzelnen Individuen entwickelt werden [Plö95]. D.h. Vertrauen in einer Co-Entwicklung ist Vertrauen zwischen den involvierten Personen. Für dieses haben experimentalpsychologische Untersuchungen [Zan77] wirtschaftlich relevante positive Korrelate des Vertrauens belegt.

Vertrauen verbessert vor allem die Qualität der Kommunikation in einer Organisation. Vertrauen entsteht durch Partizipation und Reduktion der Kontrolle. Probleme werden effektiver gelöst. Umgekehrt leidet in asymmetrischen, nicht reziproken Verhältnissen, z.B. wenn ein Partner über eine höhere punitive Macht verfügt, als erstes die Qualität der Kommunikation: Informationen werden nicht mehr weiter gegeben. [DeD98]

#### **Aufbau von Vertrauen**

Wie oben ausgeführt, profitiert die Kommunikation von Vertrauen, respektive leidet sie unter mangelndem Vertrauen. Umgekehrt trägt vor allem das Kommunikationsverhalten zum Aufbau von Vertrauen bei [Gri67]. Wesentliche Aspekte der Vertrauensbildung via Kommunikation sind die Kompetenz, Verlässlichkeit und Klarheit des Gesprächspartners.

Wichtig ist das persönliche Engagement und der persönliche Kontakt: „Erlebtes Vertrauen geht mit dem persönlichen Engagement einher, das man für eine Institution investiert. Vertrauen findet seinen Niederschlag immer auch im Handeln“ [ST03]. Schließlich ist Transparenz ein wesentlicher Baustein zum Aufbau von Vertrauen: Wer genug über den Partner weiß, um dessen Verhalten vorhersagen zu können, muss sich nicht mehr durch Kontrolle und Bestrafungsandrohung absichern.

### **Aufrechterhalten des Vertrauens**

Vertrauen und Kommunikation stehen in einem wechselseitigen sich selbst stabilisierenden Verhältnis zueinander. Eine Gefahr stellt aufkeimende Unsicherheit dar [Bec98]: diese führt zur Erosion von Vertrauen, was wiederum die Unsicherheit verstärkt. Im persönlichen Kontakt regelmäßig gemeinsam die Zukunft planen gibt Sicherheit und stabilisiert so das Vertrauen. Kommunikation setzt eine gemeinsame Sprache voraus, auch aus diesem Grund ist es wichtig, dass ein gegenseitiges Domänenwissen aufgebaut wird:

- Maschinen- bzw. Anlagebauer kommen nicht umhin, sich ein Stück weit mit der Problematik und den Eigenheiten der Software-Entwicklung auseinander zu setzen.
- Umgekehrt müssen sich die Software-Entwickler in die Funktionsweise der zu steuernden Maschine bzw. Anlage hineindenken und darüber hinaus auch ein Verständnis für die damit durchgeführten Prozesse beim Endkunden aufbauen.

## **4 Ein Vorgehensmodell für Co-Entwicklungen**

Das typische Entwicklungsvorgehen im Maschinen- und Anlagebau unterscheidet sich heute stark vom üblichen Entwicklungsvorgehen in der Informatik [Eig14]. Während in der Informatik heute mehrheitlich iterativ-inkrementelle Vorgehensweisen angewendet werden, herrscht im Maschinenbau das klassische sequentielle Vorgehen bei der Produktentwicklung vor. Das hat auch gute Gründe:

Im Gegensatz zu den klassischen Ingenieur-Disziplinen sind in der Informatik für Konzipieren (Entwerfen, Konstruieren) und Produzieren (Herstellen, Umsetzen) die gleichen Skills gefordert und entsprechend werden diese Aufgaben vom gleichen Team ausgeführt. Während in den klassischen Ingenieur-Disziplinen die Produktion kostenintensiv ist (Material und Arbeit) ist das bei der Software im Wesentlichen ein Kopieren und damit kostengünstig und schnell.

### **4.1 Das Controlling-Problem**

Bei der gleichzeitigen Entwicklung von Hard- und Software stellen Projektsteuerung und Controlling eine besondere Herausforderung dar.

Typischerweise liegt die Führung einer Co-Entwicklung beim Maschinenbauer. Entsprechend baut die Projektstruktur in aller Regel auf dem klassischen Vorgehensmodell auf. Die Integration von Hard- und Software erfolgt so ganz am Schluss. Die Entwicklungspartner arbeiten oft räumlich getrennt voneinander. Designentscheide der jeweils anderen Domäne bleiben gegenseitig verborgen. Das technische Risiko des Zusammenspiels von Hard- und Software bleibt bis zuletzt hoch.

Das Controlling mag unter diesen Voraussetzungen auf der Maschinenbau-Seite funktionieren. Viele Arbeitsergebnisse sind sicht- und prüfbar, bis zu einem gewissen Grad durchaus auch von fachlichen Laien. Auf der Software-Seite sind wichtige Deliverables nicht sichtbar und sie sind erst sinnvoll prüfbar, wenn lauffähige Software vorliegt und diese – möglichst auf der Zielhardware – ausgeführt werden kann:

### **Software-Controlling ist Blindflug bis zum ersten lauffähigen Prototyp**

Ein übergeordnetes Controlling und eine regelmäßige Abstimmung unter den Projektpartnern sind in einem Co-Entwicklungsprojekt sicher nötig.

## **4.2 Eine gemeinsame Projektstruktur für CoEntwicklungen**

Eine gemeinsame Projektsteuerung ist nur möglich, wenn ein gemeinsames Projektverständnis aufgebaut wird, welches den fundamentalen Unterschied zwischen dem Controlling von Hardware- und Software-Projekten überbrückt. Aussagekräftiges Controlling setzt Kooperation und gegenseitige Mitwirkung der Co-Entwicklungspartner voraus.

Der Projektfortschritt insbesondere auf der Softwareseite lässt sich oft nur schwer ermitteln, da die Software schon für sich wenig greifbar ist und im Fall eine Co-Entwicklung auch noch gegenseitige Abhängigkeiten bestehen. So kann zum Beispiel der Entwicklungsstand der Software nur begrenzt überprüft werden, solange die Hardware noch nicht zur Verfügung steht.

Software-Projekte haben meist iterativ-inkrementelle Vorgehensweisen (weil Konzeption und Realisierung von den gleichen Fachleuten gemacht werden und oft erst die Realisierung zeigt, welche konzeptionellen Fragen überhaupt geklärt werden müssen).

### **Synchronisationspunkte**

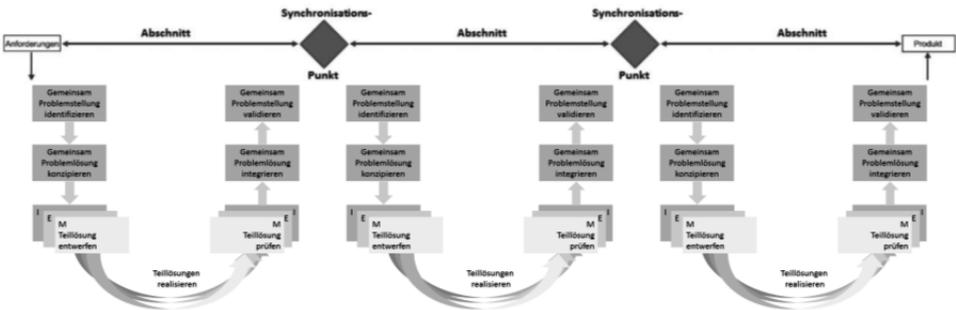
Weil Software-Controlling bis zum ersten lauffähigen Prototyp Blindflug ist, ist es aus Controlling-Sicht von höchster Bedeutung, dass so früh wie irgend möglich lauffähige Software entsteht und diese auch möglichst realitätsnah (Funktionsmuster der Maschine) demonstriert und getestet werden kann. Erst ab diesem Moment ist der Blindflug beendet und ein effektiver Controlling-Prozess kann einsetzen.

Daraus ergeben sich aus übergeordneter Controlling-Sicht zwei Ansprüche an die gemeinsame Projektleitung und -steuerung:

- sicherzustellen, dass die Maschinenentwicklung möglichst früh geeignete Funktionsmuster bereit stellt und diese entsprechend dem Projektfortschritt laufend weiter ausbaut.
- eine Rahmenplanung mit Kosten- und Ressourcenrahmen aufzusetzen, die möglichst gleichverteilt über die erwartete Projektdauer „gute“ Meilensteine aufweist, an denen der Projektfortschritt beider Projektpartner effektiv ermittelt werden kann.

Bei einer Hard- und Software Co-Entwicklung sind die zu erwartenden (Zwischen-) Ergebnisse beider Projektpartner naturgemäß unterschiedlich und insbesondere bei der Software nur unzuverlässig messbar, solange diese nicht auf einem Funktionsmuster der Maschine demonstriert und getestet werden kann.

Ein „guter“ Meilenstein ist deshalb in einem Co-Entwicklungsprojekt wenn immer möglich ein Hardware/Software-Rendez-vous. Er ist damit auch ein Kontrollpunkt, der Auskunft darüber gibt, ob die Projektpartner „aufeinander zu“ oder „nebeneinander her“ entwickeln. Diese Bedingung erfüllen die Synchronisationspunkte am Ende der Entwicklungsabschnitte.



[Fig. 1] Entwicklungsabschnitte und Synchronisationspunkte

### Entwicklungsabschnitte

Hauptteil des Co-Entwicklungsprojektes, die Konzeptions- und Realisierungsphase, muss also in Entwicklungsabschnitte gegliedert werden. Die Länge der Abschnitte ist gegeben durch die Entwicklung der Hardware, sollte jedoch ca. 3-5 Monate nicht überschreiten. Innerhalb eines Entwicklungsabschnitts wird – im Rahmen der technischen Möglichkeiten – ein Teilsystem, eine Komponente eine Erweiterung sowohl maschinenbauseitig als auch softwareseitig realisiert.

Auf den Synchronisationspunkt am Ende eines Entwicklungsabschnittes hin werden Soft- und Hardware integriert. Am gemeinsamen Meilenstein beider Entwicklungspartner werden dann die Ergebnisse des Entwicklungsabschnittes abgenommen. Je mehr maschinenseitig (allenfalls auch prototypisch) an Hardware zur Verfügung steht, desto weniger muss softwareseitig mit Mocks und Stubs simuliert werden und desto aussagekräftiger ist das Controlling der Software.

Innerhalb der Abschnitte kann die Software in Sprints von 2-4 Wochen iterativ entwickelt werden. Am Ende jedes Sprints steht ein Inkrement der Software zur Verfügung, das potentiell lauffähig wäre, jedoch vom Kunden – mangels Hardware – noch nicht abgenommen werden kann. Trotzdem sollte am Ende jeder Iteration ein Treffen mit dem Kunden stattfinden. Eine Art Sprint-Review, wo der aktuelle Stand, sowie allfällige Änderungen an der Projektplanung besprochen werden.

Für die maschinenbauseitige Entwicklung ist es wohl nicht der effizienteste Weg ebenfalls inkrementell vorzugehen (mit Zykluszeiten von Monaten, damit hier sinnvolle Einheiten entstehen können). Der Zusatzaufwand rechtfertigt sich aber allemal. Denn heute ist der Entwicklungsaufwand für eine neue Maschine auf Software Seite mindestens ebenso groß wie auf Hardware Seite. Und Software-Controlling ist Blindflug bis zum ersten lauffähigen Prototyp.

## **5 Aufsetzen einer Co-Entwicklungen**

Wir haben gesehen, dass die Entwicklung der Software nicht einfach als Auftrag an einen Dritten abgeschoben werden kann. Es gilt vielmehr einen Partner für eine Co-Entwicklung zu finden und mit diesem das gemeinsame Projekt aufzugleisen. Solche Co-Entwicklungen sind äußerst anspruchsvoll, weil vieles bei Auftragserteilung noch unbekannt ist. Ein Vorprojekt mit fixem Aufwand und fixer Dauer soll das Risiko bei der Partnerwahl begrenzen und die Voraussetzungen schaffen für eine erfolgreiche Projektdurchführung. Die für das gemeinsame Vorhaben zu erbringenden Leistungen auch müssen kommerziell und rechtlich adäquat abgebildet werden.

### **5.1 Gemeinsames Vorprojekt**

Das Vorprojekt dient dazu, dass sich die Co-Entwicklungspartner kennen lernen. Auf der Seite des Software-Partners ist die nötige Fachkompetenz eine doppelte: Softwarekompetenz und Kompetenz in der Anwendungsdomäne. Deshalb ist es eine erste wichtige gemeinsame Aufgabe der Projektpartner von HW/SW-Co-Entwicklungen diese Abgrenzung gemeinsam vorzunehmen und den Auftrag gemeinsam zu formulieren.

Im Vorprojekt ist der Scope abzuklären, die Machbarkeit zu prüfen und das Hauptprojekt zu planen. Es ist nicht realistisch alle Anforderungen an die Maschine oder Anlage von Beginn weg im Detail festzulegen. Zum einen, weil der erforderliche Detaillierungsgrad von den Projektbeteiligten abhängig ist (wenn ein Entwicklungsteam vergleichbare Projekte schon gemacht hat, ist vieles selbstverständlich was einem Branchenfremden erläutert werden müsste). Zum andern, weil sich Problemfelder und Klärungsbedarf zum Teil erst im Verlauf der Entwicklung ergeben.

Um dennoch die Kosten und benötigten Ressourcen abschätzen zu können, müssen die Co-Entwicklungspartner die Anforderungen soweit klären, dass der Umfang des Hauptpro-

jekt es abgeschätzt und eine inhaltlich sinnvolle Aufteilung in Abschnitte geplant werden kann. Im Hauptprojekt werden dann vor den jeweiligen Entwicklungsabschnitten die Anforderungen soweit detailliert, dass die Co-Entwicklungspartner ihren Beitrag planen und umsetzen können

Insbesondere muss sorgfältig geprüft werden, wie sich das System in Teilsysteme und Komponenten so gliedern lässt, dass in vernünftigen Zeitabschnitten von ca. 3-5 Monaten jeweils Hard- und Software integriert und getestet werden kann. Das entwickeln in Zyklen bzw. Entwicklungsabschnitten ist Voraussetzung für ein realistisches Controlling. Es erlaubt die Projektrisiken zu kontrollieren und die Projektziele dem Erkenntnisfortschritt anzupassen.

Weitere wichtige Themen bzw. Quellen von Missverständnissen und falschen Annahmen, die im Rahmen des Vorprojektes geklärt werden sollten sind: Konfiguration, Inbetriebnahme, Wartung und Weiterentwicklung. Auch hier macht sich der Kulturunterschied zwischen Maschinenbau und Softwareentwicklung bemerkbar: Oft haben Maschinenbauer die Erwartung, dass die Softwareentwickler ohne weitere Abmachung dafür zuständig seien. Es ist darum wichtig, die benötigte Art und den zu erwartenden Umfang dieser Unterstützung mit dem Co-Entwicklungspartner abzusprechen und geeignete Vereinbarungen rechtzeitig zu treffen.

## 5.2 Ein Vertragsmodell für Co-Entwicklungen

Herkömmliche Vertragsmodelle vermögen der parallelen Entwicklung von Maschine und Software nicht gebührend Rechnung zu tragen ([Frö04], [HM04], [Heu05], [Mor07], [Slo91], [Str03], [Str07] und [Str08]). Im Rahmen unseres KTI Forschungsprojekts sind deshalb auch Vertragsvorlagen für eine Co-Entwicklung erarbeitet worden. Dabei haben wir uns auch auf die Vorschläge von Dejaeger [Dej11], Oesterreich [Oes06], Auf der Maur/Steiner [AM11] sowie Gruhn [Gru14] gestützt.

Eine vertragliche Struktur, die den oben vorgestellten Überlegungen gerecht wird, umfasst die Ebenen Vorprojekt (Vorprojektvertrag), Softwareentwicklung (Rahmenvertrag) und die dazugehörigen Entwicklungsabschnitte (Einzelverträge).

Ein Vorprojektvertrag regelt das gemeinsame Vorprojekt. Basierend auf den Erkenntnissen des Vorprojekts wird zwischen den Parteien ein Rahmenvertrag zur Regelung der grundsätzlichen, für das Projekt geltenden Klauseln abgeschlossen. Dieser Rahmenvertrag Softwareentwicklung wird – analog den Vorschlägen des Verbundprojekts Salomo [VPS12] – um werkvertragliche Einzelverträge ergänzt, welche die spezifischen Entwicklungsabschnitte abbilden.

Nebst der Dokumentation des KTI Forschungsprojektes sind auch die Vertragsvorlagen für diese drei Vertragsarten auf der Website der HSLU frei verfügbar:

<https://www.hslu.ch/de-ch/hochschule-luzern/forschung/projekte/detail/?pid=1016>

## 6 Unterschied zu bestehenden Modellen

Das hier vorgestellte Co-Entwicklungsvorgehen hat ähnliche Ansprüche wie die VDI Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ [VDI06]. Diese will ebenfalls das domänenübergreifende Entwickeln mechatronischer Systeme methodisch unterstützen und stellt dazu Vorgehensweisen, Methoden und Werkzeuge zur Verfügung.

Unser Co-Entwicklungsvorgehen stellt sich aber explizit der Herausforderung, hybride Projektstrukturen erfolgreich umzusetzen. Dazu führt unser Modell mit den Entwicklungsabschnitten ein iteratives Element in die Maschinenentwicklung ein, das aber im Gegensatz zu den Makrozyklen des VDI Modells nicht für alle Domänen synchron ist: wir lassen beliebig schnellere Iterationen auf der Software-Seite zu.

Mit den Synchronisationspunkten schaffen wir ein Werkzeug zum zuverlässigen Controlling und zur effektiven Risiko-Beherrschung. Außerdem stellen wir dem Vorgehen auch ein angepasstes Vertragsmodell zur Seite.

### Literaturverzeichnis

- [AM11] Auf der Maur, Rolf; Steiner, Thomas: Die Quadratur des Kreises. In: Swiss Made Software - Das Buch Vol. 1, S. 102 ff, Verlag 9.6 Konzeptionelle Welten, Basel, 2011.
- [Bec98] Beckert, J. et.al.: Vertrauenserosion als organisatorische Gefahr und wie ihr zu begegnen ist. In: Organisationsentwicklung, Jg.3, Fachverlag, Würzburg, 1998.
- [DeD98] DeDreu, C. et.al.: Social Motives and Trust in Integrative Negotiation. In: Journal of Applied Psychology, Vol.83, American Psychological Association, Washington, 1998.
- [Dej11] Dejaeger, Glenn, Scrum and fixed price; impossible?: software development, 30. Januar 2011.
- [Eig14] Eigner, Martin; Roubanov, Daniil: Modellbasierte virtuelle Produktentwicklung, Springer-Verlag, Berlin 2014
- [Frö04] Fröhlich-Bleuler, Gianni: Softwareverträge, Stämpfli, Bern 2004.
- [Fun06] Funke, Joachim: Komplexes Problemlösen. In J. Funke (Ed.), Denken und Problemlösen, Hogrefe ,Göttingen 2006.
- [Gri67] Griffin, K.: The Contribution of Studies of Source Credibility to a Theory of Interpersonal trust in the Communication Process. In: Psychological Bulletin, Vol.68, American Psychological Association, Washington, 1967.
- [Gru14] Gruhn, Volker: Unberechenbares Berechnen, in [www.dotnetpro.de](http://www.dotnetpro.de), 11/2014, S. 9 ff, Stand: 20.08.2015.
- [HM04] Heusler, Bernhard; Mathys, Roland: IT-Vertragsrecht, Zürich 2004.
- [Hei05] Heinzlmann Elsbeth: Der kürzere Weg zum mechatronischen System. In: Mechatronik

F&M, S. 53ff, Carl Hanser Verlag, München, 2005.

- [Heu05] Heusler, Bernhard: Der Software-Entwicklungsvertrag, Tagungsbeitrag Tagung zum Internet-Recht und IT-Verträge vom 11.05.2004. In: Jörg, Florian S.; Arter Oliver [Hsg.], Internet-Recht und IT-Verträge, S. 49-125, Bern 2005.
- [Hof13] Hofstetter, Jörg; Jud, Martin: Was fehlt Scrum? – ein beispielhafter Lösungsansatz aus der Hochschulpraxis. In: Hanser, Eckhart; Mikusz, Martin; Fazal-Baqaie, Masud (Hrsg.): Vorgehensmodelle 2013 - Anspruch und Wirklichkeit, Tagungsband der Gesellschaft für Informatik e.V. (GI), S. 133-145, Lörrach 2013.
- [Jud09] Jud, Martin: Kosten und Nutzen von Vorgehensmodellen. In OBJEKTSpektrum 01 2009, [http://www.sigs.de/publications/os/2009/01/jud\\_OS\\_01\\_09.pdf](http://www.sigs.de/publications/os/2009/01/jud_OS_01_09.pdf), 2009.
- [Jud14] Jud, Martin: SoDa Das integrale Projekt- und Vorgehensmodell, <http://www.hslu.ch/soda>, Stand: 20.08.2015.
- [McC76] MacCormac, Richard C.: Design is...(Interview with N. Cross), BBC/Open University TV broadcast 1976.
- [Moo06] Moore, James W.: Converging Software and Systems Engineering Standards, IEEE Computer, September 2006.
- [Mor07] Morscher, Lukas: Leistungsbeschrieb, Gewährleistung und Haftung in IT-Verträgen, Tagungsbeitrag Tagung IT-Verträge vom 17.11.2006. In: Jörg, Florian S.; Arter Oliver [Hsg.], IT-Verträge, S. 73-114, Bern 2007.
- [Oes06] Oestereich, Bernd: Der agile Festpreis und andere Preis- und Vertragsmodelle, OBJEKTSpektrum, 01/2006 p30ff, 2006.
- [Plö95] Plötner, O.: Das Vertrauen des Kunden, Verlag Gabler, Wiesbaden, 1995.
- [ST03] Schweer, M.; Thies, B.: Vertrauen als Organisationsprinzip. Perspektiven für komplexe soziale Systeme. Huber, Bern, 2003.
- [Slo91] Slongo, Doris: Der Softwareherstellungsvertrag, [Diss.] Zürich, 1991.
- [Sno00] Snowden, David J: Cynefin, A Sense of Time and Place: an Ecological Approach to Sense Making and Learning in Formal and Informal Communities, conference proceedings of KMAC at the University of Aston, July 2000.
- [Str03] Straub, Wolfgang: Informatikrecht, vdf Verlag, Zürich, 2003.
- [Str07] Straub, Wolfgang: Kostenüberschreitungen in IT-Verträgen, Tagungsbeitrag Tagung IT-Verträge vom 17.11.2006. In: Jörg, Florian S.; Arter Oliver [Hsg.], IT-Verträge, S. 115-154, Bern 2007.
- [Str08] Straub, Wolfgang: Verantwortung für Informationstechnologie, Dike Verlag, Zürich, 2008.
- [VDI06] VDI, Verein Deutscher Ingenieure: Entwicklungsmethodik für mechatronische Systeme, Beuth Verlag, Berlin, 2006.

- [VPS12] Verbundprojekt Salomo, Erleichterung der Gestaltung und Abwicklung von Softwareerstellung-Verträgen zwischen KMU (Maschinenlesbare Spezifikationen von bestimmten Anforderungen und deren automatische Überprüfbarkeit durch vereinbarte Werkzeuge), Freiburg/Saarbrücken, 28. Juni 2012.
- [Wei06] Weisshaupt Bruno: SystemInnovation, Orell Fuessli, 2006.
- [Wen03] Wendt, Siegfried: Initiative „Kommunikation in der Softwareentwicklung“ Position Statements, Dez. 2003, <http://www.fmc-modeling.org/download/projects/kommse/KommSE-Position-Statements.pdf>. Stand: 20.08.2015.
- [Zan77] Zand, D.: Vertrauen und Problemlösungsverhalten von Managern. In: Lück, H. (Hrsg): Mitleid, Vertrauen, Verantwortung. Ergebnisse der Erforschung prosozialen Verhaltens, Klett, Stuttgart, 1977.
- [Züh08] Zühlke: Prozess für Produktentwicklung, [http://www.zuehlke.com/fileadmin/pdf/flyers/fl\\_055\\_d\\_produkentwicklungsprozess.pdf](http://www.zuehlke.com/fileadmin/pdf/flyers/fl_055_d_produkentwicklungsprozess.pdf). Stand: 2008.



## Controlling von hybriden Projekten – Herausforderungen und Chancen

Klaus Schopka<sup>1</sup>

**Abstract:** Das Vorgehensmodell agiler Methoden beinhaltet keine detaillierte Gesamtaufnahme aller Anforderungen und keine darauf aufbauende Gesamtplanung. Genau auf dieser Gesamtplanung beruht der traditionelle Ansatz des Controllings von Projekten. Im Beitrag wird gezeigt, dass dies aber speziell bei Projekten mit langen Laufzeiten oder hoher Komplexität keinen Verlust an Steuerungsqualität bedeutet. Vielmehr ergeben sich aus dem agilen Vorgehensmodell klare Vorteile für ein ergebnisorientiertes Controlling. Diese resultieren vor allem aus der Aufhebung von der strikten Trennung zwischen Planungsphase und Umsetzungsphase, zugunsten ergebnisgetriebener, zeitlich überschaubarer Sprints mit sofort überprüfbaren Ergebnissen.

**Keywords:** Controlling; Projekte; Planung; Agil; Hybrid

### Einleitung

„If you don't measure it, you can't manage it!“ Getreu diesem bei Controllern beliebten Grundsatz erfolgt das traditionelle Controlling von Projekten in Form einer detaillierten Gesamtplanung und Bewertung aller Aktivitäten über die gesamte Projektdauer vor Beginn der Projektdurchführung. Hierauf baut ein regelmäßiger Soll-Ist-Vergleich auf. Der Projektfortschritt wird dabei regelmäßig anhand des Verbrauchs an Ressourcen gemessen. Agile und hybride Vorgehensmodelle durchbrechen das hier zugrundeliegende Paradigma der langfristigen, detaillierten Vorausplanung und Vorausplanbarkeit. Die Planung muss stattdessen unscharfe Kosten- und Terminpläne handhaben. Das Controlling muss diese neuen Herausforderungen annehmen, um weiterhin die Grundaufgaben der Schaffung von Transparenz zur Zielerreichung zu erfüllen. Aus Sicht des Autors liegt hierin auch eine Chance, Mängel des klassischen Controllings zu beheben: **„Agile und hybride Vorgehensmodelle fördern und erfordern ein ergebnisorientiertes Controlling!“**

### Controlling von Projekten, klassisch - hybrid

Projekte werden durchgeführt, um Änderungen an bestehenden Umgebungen oder Zuständen zu erreichen. Ziel ist die Nutzung des Projektergebnisses. Die Kernfrage lautet: „Ist das Projektergebnis geeignet, die geplante Nutzung zu ermöglichen?“

---

<sup>1</sup> Projektmanagement Schopka GmbH, Blumenstr. 28, 85774 Unterföhring, klaus.schopka@schopka.com

In den letzten Jahren kommen, speziell bei der Softwareentwicklung, agile Vorgehensmodelle wie Scrum, das in Folge als Beispiel verwendet wird, immer mehr zur Anwendung [KL14]. In der Verbindung mit klassischen Elementen im Projekt, z.B. im Bereich Infrastruktur, entstehen hybride Strukturen auf die Funktionsbereiche des Unternehmens, wie das Controlling, reagieren müssen.

Aufgabe des Controllings ist es, die notwendige Transparenz zur Erreichung dieses Zieles durch die Bereitstellung erforderlicher Informationen zu schaffen. Der traditionelle Ansatz hierzu ist eine umfassende und detaillierte Zusammenstellung aller Anforderungen an das Ergebnis und der gegebenen Rahmenbedingungen in einer Projektbeschreibung. Die Vorausplanung aller Maßnahmen und Aktivitäten zur Erfüllung dieser Anforderungen über die Projektlaufzeit erfolgt im nächsten Schritt. Dieser Plan enthält alle zu erbringenden Einzelleistungen mit Terminen, Kosten, Ressourcen und Abhängigkeiten. Der Auftraggeber des Projektes erhält eine umfangreiche Kalkulation und Planung mit dem Versprechen, dass er mit Projektabschluss sein gewünschtes Ergebnis erhält. Mit dem Projektstart beginnt der laufende Soll-Ist-Vergleich bzgl. Kosten und Terminen. Die eigentliche Projektleistung bzw. der Projektfortschritt wird oft unreflektiert anhand des Verbrauchs von Ressourcen gemessen, etwa durch Methoden wie die Earned Value Analyse. Dabei wird ein direkter Zusammenhang zwischen geleisteter Arbeit und erreichtem Ergebnis unterstellt.

Bei Projekten mit kurzer Laufzeit und geringem Leistungsumfang ist das Vorgehen meist erfolgreich. Probleme entstehen bei langen Projektlaufzeiten und hoher Komplexität. Die Planung wird aufwendig und langwierig. Müssten Angebote erstellt werden, geschieht dies oft ohne hinreichende Klärung der inhaltlichen Anforderungen, die auch dem Auftraggeber zu diesem Zeitpunkt nicht immer klar sind. Neue Anforderungen und Änderungen führen zu laufenden Planänderungen, echte oder fiktive Planabweichungen müssen erklärt werden, Ressourcen werden in der Projektdurchführung umgeschichtet, Entwicklungs- und Testphasen sind erneut zu durchlaufen und der Auftragnehmer steht unter ständigem Druck, entstehende Zusatzkosten zu rechtfertigen und einzufordern. Ein hoher Aufwand, speziell auf der Seite der Projektmitarbeiter, deren Produktivität im Projekt darunter leidet. Der tatsächliche Leistungsfortschritt wird als Nebeneffekt zunehmend intransparent. Der Auftraggeber sieht das fertige Projektergebnis teilweise erst nach 1 oder 2 Jahren. Entsprechen seine Erwartungen dann immer noch denen vor 1 oder 2 Jahren? Die Erfahrung in Projekten zeigt, dass dies nicht der Fall ist.

Eine typische Erscheinung in Projekten oder Arbeitspaketen mit langer Laufzeit oder hoher Komplexität ist die schleichende zeitliche Verschiebung von Arbeit oder deren Fertigstellung zugunsten anderer Prioritäten. Über die Zeit entsteht dadurch eine wachsende Lücke zwischen verbrauchtem Budget und fertigen Leistungen. Gegen Projektenende wird die Menge noch offener Tätigkeiten unübersehbar in Form eines Hockey-Sticks. Soll das angestrebte Ergebnis ohne Einschränkungen von Leistungs- oder Qualitätsmerkmalen erreicht werden, bleiben wenige Optionen. Terminverschiebungen führen zu einer späteren Nutzung des Projektergebnisses und damit zu geringerem Nutzen. Zusätzliche Ressourcen erfordern mehr an Budget und müssen eingearbeitet werden. Der kurz-

fristige Erfolg ist somit fraglich. Im Ergebnis kommt es dann doch zu Einschränkungen im Leistungsbereich. Die geplante Nutzung kann dann nur eingeschränkt erfolgen.

Eine Schwachstelle des Controllings von Projekten liegt darin, dass die Ermittlung des Erfolgs des durch das Projekt realisierten Ergebnisses nach Projektabschluss nicht konsequent durchgeführt wird. Eine Aussage, ob ein Projekt erfolgreich ist oder nicht, ist dann nicht möglich. Es ist dann allenfalls möglich festzustellen, dass ein Projekt erfolgreich abgeschlossen wurde, nicht aber ob es erfolgreich war. Nur die Betrachtung von Kosten und Nutzen über den Projektlebenszyklus, inkl. Nutzung und Folgekosten, kann hierzu realistische Antworten liefern.

Beliebt sind auch umfangreiche Abweichungsanalysen. Dabei werden regelmäßig erhobene Ist-Daten mit Werten aus dem ursprünglichen Projektplan verglichen. Die Qualität der Ist-Daten wird unterstellt. Qualitätsprobleme sind zwar bekannt, werden aber ignoriert, da nichts „Besseres“ vorhanden ist. Der Planwert wird als richtig vorausgesetzt. Ignoriert wird dabei die Entstehung der Planwerte, die Planbarkeit hinsichtlich Komplexität, verfügbarer Ressourcen, vorhandener Planungsvorgaben und der daraus resultierenden Plan(un)genauigkeit. Controlling und Management unterliegen einer Illusion der Genauigkeit, die in Realität nicht gegeben ist, aber dennoch als Grundlage weitreichender Wirtschaftlichkeitsrechnungen und Prioritätensetzungen zwischen Projekten dienen. Planwerte in Projekten sind Schätzwerte, die auf zahlreichen Annahmen beruhen. Das Ergebnis dieser Abweichungsanalysen ist in vielen Fällen ein hoher Verbrauch an Ressourcen, ohne dadurch die Erreichung des Projektergebnisses zu unterstützen.

Ein agiles Vorgehensmodell im Projekt wirkt sich schon im ersten Planungsschritt aus. Die Anforderungen werden nicht mehr im Detail vorausgeplant, sondern grob in einem Release Plan und Product Backlog erfasst und bewertet. Erst bei Umsetzung in einem Sprint, wird im Detail geplant [Pi08]. Die strikte Trennung von Planungs- und Umsetzungsphase entfällt dadurch. Eine Kalkulation der gesamten Projektkosten ist dadurch bedingt nicht granular wie im klassischen Vorgehen. Betrachtet man die oben genannten Probleme einer Projektkalkulation, ist dies aber lediglich ein Eingeständnis des Schätzcharakters jeder Projektkalkulation. Auf der anderen Seite steht die Einsparung an Ressourcen.

Die zeitnahe Planung der einzelnen Sprints und der aktuelle Einbezug des Auftraggebers, über die Rolle des Product Owners, führen zu einer hohen Planungsgenauigkeit in der Umsetzung. Das Controlling erhält eine neue Qualität an Informationen. Offene Themen, geänderte Anforderungen und unfertige Leistungen werden durch das Vorgehensmodell transparent und können gezielt gesteuert werden. Dieses Vorgehen stellt eine Abkehr von einem zahlenorientierten Controlling dar. Der Leistungsfortschritt ist direkt ersichtlich und muss nicht aus Hilfsgrößen hochgerechnet werden.

Bei der Umsetzungsplanung werden nur die aktuellen Anforderungen konkretisiert. Nach Umsetzung stehen diese sofort zur Verfügung und können sofort durch den Auftraggeber geprüft und abgenommen werden.

Das Vorgehen impliziert einen höheren initialen Aufwand, da bereits für die ersten Sprints z.B. geeignete Testmöglichkeiten und Schnittstellen als Voraussetzung vorhanden sein müssen, die im klassischen Modell erst im Zeitablauf bereitgestellt werden.

Schon ein Blick auf das „Agile Manifest“ zeigt Konfliktpotentiale zwischen agilen/hybriden Vorgehensmodellen und Unternehmensfunktionen, wie dem Controlling. Die Betonung der Menschen und ihrer Interaktion, die Bedeutung von funktionierender, mit dem Kunden entwickelter Software, und das Eingehen auf Veränderungen im Gegensatz zur Befolgung eines Planes [Wi15], passen nicht in das Bild vieler Unternehmen, die über die Planung Kontrolle und vordergründige Sicherheit zu erreichen suchen. Zwischen den beiden Positionen gilt es einen Kompromiss zu finden, der so viel Kontrolle wie nötig, aber vor allem so viel Freiheit wie möglich, zur Nutzung der Potentiale der neuen Vorgehensmodelle ermöglicht.

Ein Vorgehensmodell zum Einbezug des Controllings in hybride Projekte kann wie folgt aussehen:

- Die Hauptvorteile agiler Modelle, wie Scrum, werden in den einzelnen Sprints realisiert. Diese müssen daher möglichst von Anforderungen abgekapselt werden, die keinen Beitrag zu dem inhaltlichen Ergebnis des Sprints leisten. Das „Sprint-Controlling“, mit Fokus auf den inhaltlichen Fortschritt, ist in Form des Sprint Backlogs, des Sprintplans, den Daily oder Weekly Scrums und Sprint Reviews hinreichend organisiert. Notwendige Informationen müssen aus dem Team heraus erstellt und kommuniziert werden.
- Das iterative Vorgehen von Scrum hat einen evolutionären Charakter. Eine Umsetzungsebene wird dabei durch Sprints, die Steuerungsebene durch Elemente wie den Releaseplan, Product Backlog und die Scrum-Organisation repräsentiert. Auf dieser Ebene agiert das Controlling unter Verzicht der Detailkontrolle der Sprints. Die Schnittstelle zwischen Steuerungsebene und Umsetzung in den Sprints kann, abhängig von der real vorzufindenden Situation, durch den Product Owner oder den Scrum Master übernommen werden, deren Rolle dadurch eine Funktionserweiterung erhält.
- Jedes Projekt - klassisch, agil oder hybrid - erfordert eine hinreichende Vorbereitung. Auch in agilen oder hybriden Projekten hat der Auftraggeber Anspruch, vor Projektbeginn eine realistische Abschätzung von Kosten und Projektdauer zu erhalten. Hier im Projektvorfeld Klarheit zu schaffen, wird nicht immer mit dem vorgesehenen Scrum Team möglich sein. Das Idealmodell von Scrum muss hier eine Anpassung an die konkrete organisatorische Umgebung erfahren [Ba15]. Die aufgegebenen Illusion einer Scheingenauigkeit wird u.a. durch Regeln zum Umgang mit Änderungen im Projektumfang ersetzt.
- Als modellkonforme Lösung für agile und hybride Projekte bietet sich die Output-orientierte Definition von Zielen im Sinne eines „Beyond Budgeting“ an [Pf03]. Dies erfordert ein radikales Umdenken in Unternehmen und in Kundenbeziehungen. Inwieweit dies kurzfristig umsetzbar ist, bleibt fraglich.

## Fazit

Agile und hybride Projekte stellen an das Controlling einerseits neue Anforderungen. Auf der anderen Seite bieten sich für das Controlling Chancen, aus der reinen Zahlensicht eines Projektes heraus zu kommen, in Richtung eines inhalts- und ergebnisorientierten Controlling als (Projekt-) Business Partner. Nichts wäre fataler hinsichtlich Ressourcenverschwendung und Wertvernichtung, als die Forderung eines Controllings, in agilen oder hybriden Projekten, “on Top“ eine klassische Gesamtplanung einzufordern. Das Controlling muss die Grenzen der Planung und Kontrolle akzeptieren und sich aktiv an der Erreichung des angestrebten Projektergebnisses und dessen anschließender Nutzung beteiligen. Der Werkzeugkasten des Controllings ist hierfür vorbereitet. Aktuelle Ansätze eines Better Budgeting, Advanced Budgeting oder Beyond Budgeting bilden den Rahmen. Konkrete Beispiele sind die Verringerung der Detailtiefe in der Planung, Rollierende Forecasts, der Einbezug nicht-monetärer Größen in die Planung oder die Output-orientierte Festlegung von Zielen und Plänen [Ob11].

Einige dieser Maßnahmen sind schnell umzusetzen, andere erfordern ein grundlegendes Umdenken und einen Kulturwandel im Unternehmen.

Was letztendlich zählt, ist das erreichte Ergebnis. Hieran muss sich auch das Controlling messen!

## Literaturverzeichnis

- [Ba15] Barth, S.: Agile Controlling – aussagekräftiges Berichtswesen für agile Organisationen. Projekt Magazin. [https://www.projektmagazin.de/artikel/agile-controlling-aussagekraeftiges-berichtswesen-fuer-agile-organisationen\\_1097683?secret=X7vIjnnPeX](https://www.projektmagazin.de/artikel/agile-controlling-aussagekraeftiges-berichtswesen-fuer-agile-organisationen_1097683?secret=X7vIjnnPeX). 25.08.2015.
- [KL14] Kurmann, M; Linssen, O.: Welche Vorgehensmodelle nutzt Deutschland? In (Engstler, M; Hanser, E.; Mikusz, M; Herzwurm, G., Hrsg.): LNI 236, Bonn, S17 ff., 2014.
- [Ob11] Obermöller, J.: Better Budgeting. ControllingWiki ICV, [http://www.controlling-wiki.com/de/index.php/Better\\_Budgeting](http://www.controlling-wiki.com/de/index.php/Better_Budgeting), 23.08.2015.
- [Pf03] Pfläging, N.: Beyond Budgeting, Better Budgeting. Planegg, S. 276 ff, 2003.
- [Pi08] Pichler, R.: Scrum – Agiles Projektmanagement erfolgreich einsetzen. S 25 ff., Heidelberg, 2008.
- [Wi15] Wikipedia, [http://de.wikipedia.org/wiki/Agile\\_Softwareentwicklung](http://de.wikipedia.org/wiki/Agile_Softwareentwicklung), 30.05.2015.



## *GI-Edition Lecture Notes in Informatics*

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühlung, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelpath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni\_ (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Rannenber, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolfrid Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web
- P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik
- P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)
- P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)
- P-69 Robert Hirschfeld, Ryszard Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006
- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-tern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS'06
- P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006
- P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics
- P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications
- P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems
- P-86 Robert Krimmer (Ed.): Electronic Voting 2006
- P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik
- P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006
- P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur
- P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006
- P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006
- P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1
- P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2
- P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen
- P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies
- P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

- P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices
- P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik
- P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik
- P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten
- P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments
- P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)
- P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme
- P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007
- P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops
- P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications
- P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures
- P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1
- P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2
- P-111 Christian Eibl, Johannes Magenheimer, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik
- P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis
- P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1<sup>st</sup> Conference on Social Semantic Web (CSSW)
- P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3<sup>rd</sup> International Conference on IT-Incident Management & IT-Forensics
- P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walthert (Eds.) German conference on bioinformatics GCB 2007
- P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1<sup>st</sup> International Working Conference on Business Process and Services Computing BPSC 2007
- P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and management The 4<sup>th</sup> International Conference on Grid Service Engineering and Management GSEM 2007
- P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007
- P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications
- P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges
- P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik
- P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

- P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit  
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)
- P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)  
9<sup>th</sup> Workshop on Parallel Systems and Algorithms (PASA)  
Workshop of the GI/ITG Special Interest Groups PARS and PARVA
- P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)  
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde  
Referate der 28. GIL Jahrestagung
- P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)  
10<sup>th</sup> Workshop Software Reengineering (WSR 2008)
- P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)  
Modellierung 2008
- P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)  
Sicherheit 2008  
Sicherheit, Schutz und Zuverlässigkeit  
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)  
2.-4. April 2008  
Saarbrücken, Germany
- P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)  
Sigsand-Europe 2008  
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems
- P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
1. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-131 Robert Krimmer, Rüdiger Grimm (Eds.)  
3<sup>rd</sup> International Conference on Electronic Voting 2008  
Co-organized by Council of Europe, Gesellschaft für Informatik und E-Voting, CC
- P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)  
DeLFI 2008:  
Die 6. e-Learning Fachtagung Informatik
- P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)  
INFORMATIK 2008  
Beherrschbare Systeme – dank Informatik Band 1
- P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)  
INFORMATIK 2008  
Beherrschbare Systeme – dank Informatik Band 2
- P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)  
Didaktik der Informatik – Aktuelle Forschungsergebnisse
- P-136 Andreas Beyer, Michael Schroeder (Eds.)  
German Conference on Bioinformatics GCB 2008
- P-137 Arslan Brömme, Christoph Busch, Detlef Hühlein (Eds.)  
BIOSIG 2008: Biometrics and Electronic Signatures
- P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)  
Synergien durch Integration und Informationslogistik  
Proceedings zur DW2008
- P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)  
Industrialisierung des Software-Managements  
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik
- P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)  
IMF 2008 - IT Incident Management & IT Forensics
- P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)  
Modellierung betrieblicher Informationssysteme (MobIS 2008)  
Modellierung zwischen SOA und Compliance Management
- P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)  
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung
- P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)  
Software Engineering 2009  
Fachtagung des GI-Fachbereichs Softwaretechnik

- P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)  
Datenbanksysteme in Business, Technologie und Web (BTW)
- P-145 Knut Hinkelmann, Holger Wache (Eds.)  
WM2009: 5th Conference on Professional Knowledge Management
- P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung  
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)
- P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)  
Business Process, Services Computing and Intelligent Service Management  
BPSC 2009 · ISM 2009 · YRW-MBP 2009
- P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)  
9<sup>th</sup> International Conference on Innovative Internet Community Systems  
I<sup>2</sup>CS 2009
- P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
2. DFN-Forum  
Kommunikationstechnologien  
Beiträge der Fachtagung
- P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)  
Software Engineering  
2009 - Workshopband
- P-151 Armin Heinzl, Peter Dadam, Stefan Kirm, Peter Lockemann (Eds.)  
PRIMIUM  
Process Innovation for Enterprise Software
- P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)  
Enterprise Modelling and Information Systems Architectures  
Proceedings of the 3<sup>rd</sup> Int'l Workshop EMISA 2009
- P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)  
Lernen im Digitalen Zeitalter  
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik
- P-154 Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Hrsg.)  
INFORMATIK 2009  
Im Focus das Leben
- P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)  
BIOSIG 2009:  
Biometrics and Electronic Signatures  
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
- P-156 Bernhard Koerber (Hrsg.)  
Zukunft braucht Herkunft  
25 Jahre »INFOS – Informatik und Schule«
- P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)  
German Conference on Bioinformatics 2009
- P-158 W. Claudepein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)  
Precision Agriculture  
Reloaded – Informationsgestützte Landwirtschaft
- P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)  
Software Engineering 2010
- P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)  
Software Engineering 2010 –  
Workshopband  
(inkl. Doktorandensymposium)
- P-161 Gregor Engels, Dimitris Karagiannis, Heinrich C. Mayr (Hrsg.)  
Modellierung 2010
- P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)  
Vernetzte IT für einen effektiven Staat  
Gemeinsame Fachtagung  
Verwaltungsinformatik (FTVI) und  
Fachtagung Rechtsinformatik (FTRI) 2010
- P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)  
Mobile und Ubiquitäre Informationssysteme  
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler  
Kollaboration
- P-164 Arslan Brömme, Christoph Busch (Eds.)  
BIOSIG 2010: Biometrics and Electronic Signatures  
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

- P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)  
10<sup>th</sup> International Conference on Innovative Internet Community Systems (I<sup>2</sup>CS) – Jubilee Edition 2010 –
- P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
3. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-167 Robert Krimmer, Rüdiger Grimm (Eds.)  
4<sup>th</sup> International Conference on Electronic Voting 2010  
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)  
Didaktik der Informatik  
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik
- P-169 Michael Kerres, Nadine Ojstersek, Ulrik Schroeder, Ulrich Hoppe (Hrsg.)  
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.
- P-170 Felix C. Freiling (Hrsg.)  
Sicherheit 2010  
Sicherheit, Schutz und Zuverlässigkeit
- P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)  
Modellierung betrieblicher Informationssysteme (MobIS 2010)  
Modellgestütztes Management
- P-172 Stefan Klink, Agnes Koschmider, Marco Mevius, Andreas Oberweis (Hrsg.)  
EMISA 2010  
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme  
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)
- P-173 Dietmar Schomburg, Andreas Grote (Eds.)  
German Conference on Bioinformatics 2010
- P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)  
perspeGktive 2010  
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“
- P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)  
INFORMATIK 2010  
Service Science – Neue Perspektiven für die Informatik  
Band 1
- P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)  
INFORMATIK 2010  
Service Science – Neue Perspektiven für die Informatik  
Band 2
- P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)  
INFORMATIK 2010  
Business Process and Service Science – Proceedings of ISSS and BPSC
- P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)  
Vom Projekt zum Produkt  
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010
- P-179 Stefan Gruner, Bernhard Rumpe (Eds.)  
FM+AM'2010  
Second International Workshop on Formal Methods and Agile Methods
- P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)  
Datenbanksysteme für Business, Technologie und Web (BTW) 14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)
- P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)  
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft
- P-182 Ronald Maier (Hrsg.)  
6<sup>th</sup> Conference on Professional Knowledge Management  
From Knowledge to Action
- P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)  
Software Engineering 2011  
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)  
Software Engineering 2011  
Workshopband  
(inkl. Doktorandensymposium)

- P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)  
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)
- P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)  
11<sup>th</sup> International Conference on Innovative Internet Community Systems (I<sup>2</sup>CS)
- P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)  
4. DFN-Forum Kommunikationstechnologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn
- P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)  
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden
- P-189 Thomas, Marco (Hrsg.)  
Informatik in Bildung und Beruf INFOS 2011  
14. GI-Fachtagung Informatik und Schule
- P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)  
Enterprise Modelling and Information Systems Architectures (EMISA 2011)
- P-191 Arslan Brömme, Christoph Busch (Eds.)  
BIOSIG 2011  
International Conference of the Biometrics Special Interest Group
- P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)  
INFORMATIK 2011  
Informatik schafft Communities
- P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)  
IMDM 2011
- P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)  
Informationstechnologie für eine nachhaltige Landwirtschaft Fokus Forstwirtschaft
- P-195 Neeraj Suri, Michael Waidner (Hrsg.)  
Sicherheit 2012  
Sicherheit, Schutz und Zuverlässigkeit Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
- P-196 Arslan Brömme, Christoph Busch (Eds.)  
BIOSIG 2012  
Proceedings of the 11<sup>th</sup> International Conference of the Biometrics Special Interest Group
- P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)  
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012
- P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)  
Software Engineering 2012  
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)  
Software Engineering 2012  
Workshopband
- P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)  
ARCS 2012 Workshops
- P-201 Elmar J. Sinz Andy Schürr (Hrsg.)  
Modellierung 2012
- P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)  
MMS 2012: Mobile und Ubiquitäre Informationssysteme
- P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)  
5. DFN-Forum Kommunikationstechnologien  
Beiträge der Fachtagung
- P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)  
12<sup>th</sup> International Conference on Innovative Internet Community Systems (I<sup>2</sup>CS 2012)
- P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)  
5<sup>th</sup> International Conference on Electronic Voting 2012 (EVOTE2012)  
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)  
EMISA 2012  
Der Mensch im Zentrum der Modellierung
- P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)  
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.  
24.–26. September 2012

- P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)  
INFORMATIK 2012
- P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)  
Nachhaltiges Software Management
- P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)  
Automotive – Safety & Security 2012  
Sicherheit und Zuverlässigkeit für automobile Informationstechnik
- P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)  
Massendatenmanagement in der Agrar- und Ernährungswirtschaft  
Erhebung - Verarbeitung - Nutzung  
Referate der 33. GIL-Jahrestagung  
20. – 21. Februar 2013, Potsdam
- P-212 Arslan Brömme, Christoph Busch (Eds.)  
BIOSIG 2013  
Proceedings of the 12th International Conference of the Biometrics Special Interest Group  
04.–06. September 2013  
Darmstadt, Germany
- P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)  
Software Engineering 2013  
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mitschang, Theo Härder, Veit Köppen (Hrsg.)  
Datenbanksysteme für Business, Technologie und Web (BTW) 2013  
13. – 15. März 2013, Magdeburg
- P-215 Stefan Wagner, Horst Lichter (Hrsg.)  
Software Engineering 2013  
Workshopband  
(inkl. Doktorandensymposium)  
26. Februar – 1. März 2013, Aachen
- P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)  
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband  
11. – 12. März 2013, Magdeburg
- P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)  
6. DFN-Forum Kommunikationstechnologien  
Beiträge der Fachtagung  
03.–04. Juni 2013, Erlangen
- P-218 Andreas Breiter, Christoph Rensing (Hrsg.)  
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)  
8. – 11. September 2013, Bremen
- P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)  
Informatik erweitert Horizonte  
INFOS 2013  
15. GI-Fachtagung Informatik und Schule  
26. – 28. September 2013
- P-220 Matthias Horbach (Hrsg.)  
INFORMATIK 2013  
Informatik angepasst an Mensch, Organisation und Umwelt  
16. – 20. September 2013, Koblenz
- P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimos Tambouris (Eds.)  
Electronic Government and Electronic Participation  
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013  
16. – 19. September 2013, Koblenz
- P-222 Reinhard Jung, Manfred Reichert (Eds.)  
Enterprise Modelling and Information Systems Architectures (EMISA 2013)  
St. Gallen, Switzerland  
September 5. – 6. 2013
- P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)  
Open Identity Summit 2013  
10. – 11. September 2013  
Kloster Banz, Germany
- P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)  
Vorgehensmodelle 2013  
Vorgehensmodelle – Anspruch und Wirklichkeit  
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.  
Lörrach, 2013
- P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)  
Modellierung 2014  
19. – 21. März 2014, Wien
- P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)  
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement  
Referate der 34. GIL-Jahrestagung  
24. – 25. Februar 2014, Bonn

- P-227 Wilhelm Hasselbring,  
Nils Christian Ehmke (Hrsg.)  
Software Engineering 2014  
Fachtagung des GI-Fachbereichs  
Softwaretechnik  
25. – 28. Februar 2014  
Kiel, Deutschland
- P-228 Stefan Katzenbeisser, Volkmar Lotz,  
Edgar Weippl (Hrsg.)  
Sicherheit 2014  
Sicherheit, Schutz und Zuverlässigkeit  
Beiträge der 7. Jahrestagung des  
Fachbereichs Sicherheit der  
Gesellschaft für Informatik e.V. (GI)  
19. – 21. März 2014, Wien
- P-230 Arslan Brömme, Christoph Busch (Eds.)  
BIOSIG 2014  
Proceedings of the 13<sup>th</sup> International  
Conference of the Biometrics Special  
Interest Group  
10. – 12. September 2014 in  
Darmstadt, Germany
- P-231 Paul Müller, Bernhard Neumair,  
Helmut Reiser, Gabi Dreo Rodosek  
(Hrsg.)  
7. DFN-Forum  
Kommunikationstechnologien  
16. – 17. Juni 2014  
Fulda
- P-232 E. Plödereder, L. Grunske, E. Schneider,  
D. Ull (Hrsg.)  
INFORMATIK 2014  
Big Data – Komplexität meistern  
22. – 26. September 2014  
Stuttgart
- P-233 Stephan Trahasch, Rolf Plötzner, Gerhard  
Schneider, Claudia Gayer, Daniel Sassiati,  
Nicole Wöhrle (Hrsg.)  
DeLFI 2014 – Die 12. e-Learning  
Fachtagung Informatik  
der Gesellschaft für Informatik e.V.  
15. – 17. September 2014  
Freiburg
- P-234 Fernand Feltz, Bela Mutschler, Benoît  
Otjacques (Eds.)  
Enterprise Modelling and Information  
Systems Architectures  
(EMISA 2014)  
Luxembourg, September 25-26, 2014
- P-235 Robert Giegerich,  
Ralf Hofestädt,  
Tim W. Nattkemper (Eds.)  
German Conference on  
Bioinformatics 2014  
September 28 – October 1  
Bielefeld, Germany
- P-236 Martin Engstler, Eckhart Hanser,  
Martin Mikusz, Georg Herzwurm (Hrsg.)  
Projektmanagement und  
Vorgehensmodelle 2014  
Soziale Aspekte und Standardisierung  
Gemeinsame Tagung der Fachgruppen  
Projektmanagement (WI-PM) und  
Vorgehensmodelle (WI-VM) im  
Fachgebiet Wirtschaftsinformatik der  
Gesellschaft für Informatik e.V., Stuttgart  
2014
- P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)  
Open Identity Summit 2014  
4.–6. November 2014  
Stuttgart, Germany
- P-238 Arno Ruckelshausen, Hans-Peter  
Schwarz, Brigitte Theuvsen (Hrsg.)  
Informatik in der Land-, Forst- und  
Ernährungswirtschaft  
Referate der 35. GIL-Jahrestagung  
23. – 24. Februar 2015, Geisenheim
- P-239 Uwe Aßmann, Birgit Demuth, Thorsten  
Spitta, Georg Püschel, Ronny Kaiser  
(Hrsg.)  
Software Engineering & Management  
2015  
17.-20. März 2015, Dresden
- P-240 Herbert Klenk, Hubert B. Keller, Erhard  
Plödereder, Peter Dencker (Hrsg.)  
Automotive – Safety & Security 2015  
Sicherheit und Zuverlässigkeit für  
automobile Informationstechnik  
21.–22. April 2015, Stuttgart
- P-241 Thomas Seidl, Norbert Ritter,  
Harald Schöning, Kai-Uwe Sattler,  
Theo Härder, Steffen Friedrich,  
Wolfram Wingerath (Hrsg.)  
Datenbanksysteme für Business,  
Technologie und Web (BTW 2015)  
04. – 06. März 2015, Hamburg
- P-242 Norbert Ritter, Andreas Henrich,  
Wolfgang Lehner, Andreas Thor,  
Steffen Friedrich, Wolfram Wingerath  
(Hrsg.)  
Datenbanksysteme für Business,  
Technologie und Web (BTW 2015) –  
Workshopband  
02. – 03. März 2015, Hamburg
- P-243 Paul Müller, Bernhard Neumair, Helmut  
Reiser, Gabi Dreo Rodosek (Hrsg.)  
8. DFN-Forum  
Kommunikationstechnologien  
06.–09. Juni 2015, Lübeck

- P-244 Alfred Zimmermann,  
Alexander Rossmann (Eds.)  
Digital Enterprise Computing  
(DEC 2015)  
Böblingen, Germany June 25-26, 2015
- P-245 Arslan Brömme, Christoph Busch ,  
Christian Rathgeb, Andreas Uhl (Eds.)  
BIOSIG 2015  
Proceedings of the 14th International  
Conference of the Biometrics Special  
Interest Group  
09.–11. September 2015  
Darmstadt, Germany
- P-246 Douglas W. Cunningham, Petra Hofstedt,  
Klaus Meer, Ingo Schmitt (Hrsg.)  
INFORMATIK 2015  
28.9.-2.10. 2015, Cottbus
- P-247 Hans Pongratz, Reinhard Keil (Hrsg.)  
DeLFI 2015 – Die 13. E-Learning  
Fachtagung Informatik der Gesellschaft  
für Informatik e.V. (GI)  
1.–4. September 2015  
München
- P-248 Jens Kolb, Henrik Leopold, Jan Mendling  
(Eds.)  
Enterprise Modelling and Information  
Systems Architectures  
Proceedings of the 6th Int. Workshop on  
Enterprise Modelling and Information  
Systems Architectures, Innsbruck, Austria  
September 3-4, 2015
- P-249 Jens Gallenbacher (Hrsg.)  
Informatik  
allgemeinbildend begreifen  
INFOS 2015 16. GI-Fachtagung  
Informatik und Schule  
20.–23. September 2015
- P-250 Martin Engstler, Masud Fazal-Baqaie,  
Eckhart Hanser, Martin Mikusz,  
Alexander Volland (Hrsg.)  
Projektmanagement und  
Vorgehensmodelle 2015  
Hybride Projektstrukturen erfolgreich  
umsetzen  
Gemeinsame Tagung der Fachgruppen  
Projektmanagement (WI-PM) und  
Vorgehensmodelle (WI-VM) im  
Fachgebiet Wirtschaftsinformatik  
der Gesellschaft für Informatik e.V.,  
Elmshorn 2015

The titles can be purchased at:

**Köllen Druck + Verlag GmbH**

Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn

Fax: +49 (0)228/9898222

E-Mail: [druckverlag@koellen.de](mailto:druckverlag@koellen.de)

