

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 978-3-88579-630-5

This volume contains papers from the conference “Projektmanagement und Vorgehensmodelle 2014” (PVM 2014) held in Stuttgart October 16 to 17, 2014. 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, concept and tools are discussed at the future track papers.



M. Engstler, E. Hanser, M. Mikusz, G. Herzwurm (Hrsg.): Projektmanagement und Vorgehensmodelle 2014

GI-Edition

Lecture Notes in Informatics

**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**



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.**

**16. und 17. Oktober 2014
in Stuttgart**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-236

ISBN 978-3-88579-630-5

ISSN 1617-5468

Volume Editors

Prof. Dr. Martin Engstler

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

Prof. Dr. Eckhart Hanser

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

Dr. Martin Mikusz

Universität Stuttgart (martin.mikusz@gsame.uni-stuttgart.de)

FOM Hochschule für Oekonomie und Management Stuttgart

Prof. Dr. Georg Herzwurm

Universität Stuttgart (herzwurm@wius.bwi.uni-stuttgart.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 2014

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Liebe Leserinnen und Leser,

der vorliegende Tagungsband enthält die Beiträge der ersten Fachtagung „Projektmanagement und Vorgehensmodelle 2014“ (PVM 2014), die gemeinsam von den Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. durchgeführt wird. Beide GI-Fachgruppen haben in den vergangenen Jahren bereits erfolgreiche Veranstaltungen zur Zukunft des Projektmanagements und zu aktuellen Fragestellungen bei Vorgehensmodellen in der Softwareentwicklung durchgeführt. So hat die Fachgruppe Projektmanagement zusammen mit der Deutschen Gesellschaft für Projektmanagement (GPM) seit 2003 mit der Tagung interPM eine der größten interdisziplinären Tagungen zur Zukunft im Projektmanagement veranstaltet. Seit über 20 Jahren gilt die Fachtagung „Vorgehensmodelle“ der Fachgruppe Vorgehensmodelle als die wichtigste Veranstaltung zu diesem Thema im deutschsprachigen Raum. Beide Veranstaltungsreihen verfolgten dabei stets das Ziel, fundierte Ansätze und Erfahrungen zu deren Anwendung in der Praxis einem Fachpublikum aus Wissenschaftlern und Praktikern vorzustellen, und im intensiven Diskurs und Erfahrungsaustausch zu vertiefen. Mit der ersten gemeinsamen Tagung wird den vielfältigen thematischen Anknüpfungspunkten der Arbeiten in den beiden Fachgruppen sowie den jeweiligen Fachtagungen Rechnung getragen.

Vorgehensmodelle der Softwareentwicklung, insbesondere die agilen Ansätze, haben in allen Anwendungsfeldern des Projektmanagements und darüber hinaus Eingang gefunden. Es sind neue Arbeitsformen für die Projektbeteiligten entstanden, die klassische und agile Vorgehensmodelle aufgreifen bzw. diese in hybriden Ansätzen kombinieren. Hierbei haben sich auch neue Rollen und Kooperationsmechanismen in der Projektarbeit herausgebildet. Dabei stellt sich u. a. die Frage, welche Erfahrungen inzwischen vorliegen, wie diese ggf. in die Standards im Projektmanagement einfließen können und welche sozialen bzw. kulturellen Aspekte die agilen bzw. die zunehmend hybriden Vorgehensmodelle implizieren.

Die Tagung eröffnet mit einer Keynote, in der Marco Kuhmann und Oliver Linssen unter dem Vortragstitel „Welche Vorgehensmodelle nutzt Deutschland“ die Ergebnisse einer Studie zur Verbreitung von Vorgehensmodellen erstmals öffentlich präsentieren. In der eingeladenen Keynote am zweiten Veranstaltungstag analysiert Peter Meyerhans, CIO von Drees & Sommer, erfolgreiche Bauprojekte und leitet daraus Empfehlungen für ein erfolgreiches IT-Projektmanagement ab.

Das Hauptprogramm der Tagung umfasst acht ausgewählte Beiträge aus Praxis und Wissenschaft (inkl. der Eröffnungs-Keynote), 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 47%) erst einen objektiven Bewertungsprozess möglich machten.

Unter dem Leitthema der Tagung „Soziale Aspekte und Standardisierung“ analysieren die Experten aus Wissenschaft und Wirtschaft in ihren Fachbeiträgen Konzepte und Methoden agiler bzw. hybrider Vorgehensmodelle im Gesamtkontext des Projektmanagements. Hierzu gehören Aspekte wie die Möglichkeiten und Notwendigkeiten einer Standardisierung, aber auch Erfahrungen in spezifischen Einsatzfeldern (z. B. internationale Projekte, On-/Offshore-Projekte) sowie besonderen Rahmenbedingungen und Konfigurationen (z.B. Integration agiler Modelle in traditionellen Unternehmensstrukturen bzw. Großunternehmen, Erfüllung von Government-Anforderungen, Interessensausgleich durch gezieltes Stakeholder-Management).

Ergänzend dazu liefern die sechs ausgewählten „Future Track-Vorträge“ weitere Impulse in Form von (teilweise provokanten) Thesen und Vorstellung innovativer Konzepte und Tools, die mit dem Auditorium direkt oder in ergänzenden Open Spaces diskutiert und vertieft werden.

Ein großer Dank gilt dem Hauptsponsor *KOMTEAMER GmbH* sowie dem Medienpartner *Projekt Magazin*, die die Finanzierung der Tagung und des Tagungsbandes erheblich erleichterten.

Unser Dank gilt auch der Universität Stuttgart, die uns das Internationale Begegnungszentrum für unsere Tagung zur Verfügung stellt und einen inspirierenden Rahmen für die Vorträge und den interaktiven Diskurs bietet.

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 Fachgruppen *Projektmanagement WI-PM* (<http://fg-wi-pm.gi.de/>) und *Vorgehensmodelle WI-VM* (<http://www.vorgehensmodelle.de/>).

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

Stuttgart und Lörrach im Oktober 2014

Martin Engstler, Eckhart Hanser, Martin Mikusz, Georg Herzwurm

Programmkomitee

Vorsitz

Prof. Dr. Martin Engstler, Hochschule der Medien Stuttgart

Prof. Dr. Eckhart Hanser, Duale Hochschule Lörrach

Mitglieder

Dr. Volker Arendt, Bergische Universität Wuppertal

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

Dr. Martin Bertram, Commerzbank AG

Dr. Marco Kuhrmann, University of Southern Denmark

Hubert Biskup, IBM Rational Software

Dr. Oliver Linsen

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

Dr. Martin Mikusz, Universität Stuttgart

Prof. Dr. Martin Engstler, Hochschule der Medien Stuttgart

Günther Müller-Luschnat, iteratec GmbH

Dr. Thomas Greb, Thomas Greb Consulting

Prof. Dr. Andreas Oberweis, Universität Karlsruhe

Dr. Alexander Grimme, KOMTEAMER GmbH

Prof. Dr. Wolfram Pietsch, Fachhochschule Aachen

Prof. Dr. Eckhart Hanser, Duale Hochschule Lörrach

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

Prof. Dr. Georg Herzwurm, Universität Stuttgart

Alexander Volland, Union IT-Services GmbH

Gerrit Kerber, aragon interactive GmbH

Prof. Dr. Doris Weßels, Fachhochschule Kiel

Organisatorische Leitung

Prof. Dr. Martin Engstler, Hochschule der Medien Stuttgart

Dr. Martin Mikusz, Universität Stuttgart

Prof. Dr. Eckhart Hanser, Duale Hochschule Lörrach

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

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, followed by "Magazin" in a regular weight of the same font. Below this, the tagline "Das Fachportal für Projektmanagement" is written in a smaller, white, sans-serif font.

ProjektMagazin
Das Fachportal für Projektmanagement

Projekt Magazin ist das Online-Fachportal zum Thema Projektmanagement für Projektleiter, Projektmitarbeiter und Berater. Unter www.projektmagazin.de finden Sie praxisnahe Artikel und konkrete Unterstützung für Ihre Projekt-Aufgaben. Im Projekt Magazin schreiben Experten für Experten: In über 1600 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 – Keynote

Marco Kuhrmann, Oliver Linssen

Welche Vorgehensmodelle nutzt Deutschland?17

Teil II – Hauptprogramm

Bodo Kraft, Axel Zöll

Von der Langstrecke zum Sprint – Agile Methoden in traditionellen Unternehmen35

Stefan Hilmer, Alexander Krieg

Standardisierung vs. Kultur: Klassisches und agiles Projektmanagement im Vergleich47

Masud Fazal-Baqai, Stefan Sauer, Torsten Heuft

Agile Entwicklung mit On- und Offshore-Partnern – Methodenverbesserung in der Praxis59

Daniel Burgwinkel

Erweiterung von Vorgehensmodellen um Aspekte der Information Governance.....71

Gunnar Auth

Zur Rolle des Stakeholder-Managements in IT-Projekten an Hochschulen – Erfahrungen aus der Einführung eines integrierten Campus-Management-Systems...83

Patrick Daut

Agil in großen Organisationen: Eine neue Rolle im Scrum-Framework99

Teil III – Eingeladene Beiträge der Session „Future Track“

Jan Friedrich

Formale Systemmodelle der Informatik als Basis für eine bessere Beherrschung von KPM-Szenarien113

Christa Weßel

Elche fangen - Von der Kunst in IT-Projekten Beobachtungen und Interviews durchzuführen121

Hans Christian Liebig

Praktische Umsetzung eines internationalen Web-Relaunchs mit Hilfe einer hybriden Projektmethodik129

Klaus Schopka

Dimensionen von Projekten - Differenzierung von Standards und Methoden?135

Roman Roelofsen, Stephan Wilczek

*Markup-basiertes Spezifikations- und Anforderungsmanagement in agilen
Softwareprojekten*143

Teil I
Keynote

Welche Vorgehensmodelle nutzt Deutschland?

Marco Kuhrmann^{1,2}, Oliver Linssen³

¹University of Southern Denmark, The Mærsk Mc-Kinney Møller Institute
kuhrmann@mmmi.sdu.dk

²Technische Universität München, Fakultät für Informatik
kuhrmann@mytum.de

³Bergische Universität Wuppertal, Schumpeter School of Business and Economics
Lehrstuhl für Wirtschaftsinformatik und Operations Research
linssen@wiwi.uni-wuppertal.de

Abstract: In der Praxis finden viele unterschiedliche Vorgehensmodelle Anwendung, was oft in Problemen resultiert, da sich z.B. Philosophie, Projektstruktur, Terminologie oder Rollenmodelle und Aufgabenzuordnung unterscheiden. Ziel dieses Beitrags ist es, eine Karte zu zeichnen, in welcher die in Deutschland aktuell verwendeten Vorgehensmodelle enthalten sind. Dieser Beitrag präsentiert die Ergebnisse einer Studie aus dem Jahr 2006 und stellt diesen die Ergebnisse gegenüber, welche als Teil des *3ProcSurvey* im Jahr 2013 ermittelt wurden. Wir stellen dar, welche Vorgehensmodelle aktuell im Einsatz sind und wie sich die Anwendung von Vorgehensmodellen über die Zeit entwickelt hat. Die Studie hat gezeigt, dass eine Vielzahl von Vorgehensmodellen im Einsatz ist. Es werden sowohl agile als auch „klassische“ Ansätze angewendet, obwohl ein Trend weg von großen Standards zu beobachten ist. Insbesondere zeigt die Studie aber auch, dass ob der großen Anzahl von Vorgehensmodellen, die Auswahl und das Tailoring in der Regel wenig systematisch und individuell durch Projektleiter erfolgen.

1 Einleitung

Vorgehensmodelle für die Softwareentwicklung haben vielfältige Aufgaben: Sie legen den organisatorischen Rahmen für Projekte fest, sie geben Projekten Struktur und sie bieten den Nutzern von Vorgehensmodellen Hilfestellung bei der Planung und der Umsetzung der Entwicklungsaufgaben [BK13]. Dazu beschreiben Vorgehensmodelle i.d.R., welche Ergebnisse zu erstellen sind (Artefaktmodell), welche Rollen für die Erstellung der Ergebnisse verantwortlich sind (Rollenmodell), wie die Ergebnisse erstellt werden sollen (Methoden), wann und in welcher zeitlichen Ordnung und mit welchen (zeitlichen und logischen) Abhängigkeiten die Ergebnisse fertigzustellen sind (Prozessmodell). Grundsätzlich lassen sich Vorgehensmodelle also auf wenige Kernkonzepte zurückführen; und dennoch gibt es eine Vielzahl von Vorgehensmodellen, die sich (oftmals nur in Nuancen) unterscheiden. In der Regel wird das Ziel verfolgt, ein für das jeweilige Unternehmen bestmöglich angepasstes Vorgehensmodell zu erhalten, sei es durch Adaption eines Standardmodells oder durch Festlegen eines individuellen Vorgehens. Dies kann

bei der Zusammenarbeit zwischen unterschiedlichen Organisationen zu Reibungsverlusten führen, da sich z.B. Projektorganisation, die jeweils existierenden Rollen, eingesetzte Methoden oder die Terminologie unterscheiden. Ein Ansatz, dieser Situation zu begegnen, ist die umfassende Gegenüberstellung und gegenseitige Abbildung der Konzepte. Hierzu ist es zunächst jedoch erforderlich zu verstehen, welche Vorgehensmodelle überhaupt eingesetzt werden.

Dieses Papier leistet einen Beitrag zur Analyse der aktuell im Einsatz befindlichen Vorgehensmodelle. Wir greifen die Ergebnisse der IOSE-W² [FK07] Studie aus dem Jahr 2006 auf und verwenden diese als initiale Messung. Anschließend stellen wir diese Studie den Ergebnissen des im Jahr 2012/2013 durchgeführten *3ProcSurvey* [SK+13] gegenüber. Wir präsentieren die Liste der im Einsatz befindlichen Vorgehensmodelle und diskutieren die Entwicklung in der Nutzung über die letzten Jahre. Gleichzeitig ist dieser Beitrag eine Motivation: Der *3ProcSurvey* hat das Ziel, die Systematik und Professionalität der Softwareentwicklung zu analysieren. Vorgehensmodelle sind, neben weiteren Kerndisziplinen wie z.B. Projekt- und Qualitätsmanagement, ein Teil einer systematischen Softwareentwicklung. Dieser Beitrag soll die Teilnahme an folgenden Instanzen des *3ProcSurvey* motivieren, der ebenfalls kurz vorgestellt wird.

Der Rest dieses Papiers ist wie folgt strukturiert: In Abschnitt 2 setzen wir den Kontext und geben eine kurze Übersicht zu verwandten Arbeiten. In Abschnitt 3 stellen wir die Ergebnisse von IOSE-W² und *3ProcSurvey* vor und vergleichen diese. Das Papier wird im Abschnitt 4 zusammengefasst.

2 Kontext und Verwandte Arbeiten

Die Untersuchung hinsichtlich der Nutzung von Vorgehensmodellen und ihres Einflusses auf den Projekterfolg wird an vielen Stellen schon über Jahre hinweg betrieben. Speziell in Deutschland sind hier die Umfragen im Kontext der Studien IOSE-W² [FK07] und SUCCESS [BEJ06] zu nennen. Im internationalen Umfeld ist der vielzitierte (und kritisierte) CHAOS Report der Standish Group [Sta06] zu nennen. Diese Umfragen lassen sich zwei Gruppen zuordnen: Während die ersten beiden einmalige Analysen im Rahmen von Forschungsprojekten waren (methodisch rigoros und präzise), ist der CHAOS Report eine kontinuierliche Untersuchung, bei der viele Aspekte der Methode nicht nachvollziehbar dokumentiert sind. Unter anderem sind die Methoden der Auswertung ein Betriebsgeheimnis, weshalb er oft in Frage gestellt wird. Die gerade aufgezählten Umfragen verfolgen eher grundsätzliche und dabei themenübergreifende Analysen, in denen die unterschiedlichen Aspekte von Softwareprojekten abgefragt werden, z.B. Art und Weise des Projektmanagements, Grad der Verteilung von Projektteams, Einsatz von Vorgehensmodellen, grundsätzliche Erfolgsfaktoren, und so weiter.

Darüber hinaus gibt es noch eine Vielzahl an spezialisierten Umfragen (auf die wir aus Platzgründen nicht weiter im Detail eingehen), etwa im Bereich V-Modell XT [KLS11], der Agilen Methoden [Mah08] oder im Requirements Engineering [MW13].

Der 3ProcSurvey [SK+13] ist den themenübergreifenden Studien zuzuordnen und setzt dort, wo Studien wie IOSE-W² [FK07] oder SUCCESS [BEJ06] aufhörten. Ziel ist es, ein Instrument für eine kontinuierliche Ermittlung des aktuellen Status der Softwareentwicklung in Deutschland (und auch international) zu schaffen. Gegenstand der Untersuchung sind die Kernkomponenten, die eine professionelle und systematische Softwareentwicklung dem Ideal eines ingenieurmäßigen Vorgehens folgend ausmachen. Diese Kernkomponenten sind Vorgehensmodelle zum Setzen eines organisatorischen Rahmens (in dem Softwareentwicklung systematisch und reproduzierbar ablaufen kann), das Projektmanagement als primäres Steuerungsinstrument für Softwareprojekte, und die Qualitätssicherung zur kontinuierlichen Überprüfung der Qualität.

Als Grundlage für den 3ProcSurvey dienten einmal die BITKOM-Veröffentlichung zum Agilen Software Engineering [DS+13], welche den Referenzpunkt für Werte und Ziele setzt (etwa Modularisierung und Wiederverwendung), sowie insbesondere die Umfrage zu den Themen Vorgehensmodelle, verteilte Entwicklung, Wandlung und Prozessverbesserung aus dem Projekt IOSE-W². Unternehmen wurden hinsichtlich ihrer Erwartungshaltung zu den o.g. Prozessen befragt und zu deren tatsächlicher Umsetzung. Erwartungsgemäß zeigen solche Befragungen Lücken¹ auf. Daher floss in die Umfrage auch das Thema Verbesserungsprozesse mit ein, um zu ermitteln, ob und wie die befragten Unternehmen in der Lage sind, Optimierungspotenzial zu erkennen, zu nutzen, und auch tatsächlich zu implementieren.

In enger Zusammenarbeit von Technischer Universität München, SQS AG, BITKOM und der Gesellschaft für Informatik (GI) e.V. ist ein Fragebogen entstanden, welcher 33 Top-Level-Fragen enthält und in Summe etwa 150 Variablen abfragt. Gegenstand des Fragebogens sind, neben einigen demografischen Fragen, insbesondere:

- Wie wichtig schätzen die Teilnehmer Vorgehensmodelle, Projekt- und Qualitätsmanagement für den Projekterfolg ein und wo (bei diesen Prozessen) wird das meiste Verbesserungspotenzial gesehen?
- Welche Erwartungshaltung haben die Teilnehmer hinsichtlich Vorgehensmodellen, Projekt- und Qualitätsmanagement und hinsichtlich ihrer Verzahnung?
- Wie werden Vorgehensmodelle, Projekt- und Qualitätsmanagement tatsächlich in Softwareprojekten implementiert?
- Wie werden kontinuierliche Verbesserungsprozesse in den befragten Unternehmen implementiert?

Inhalt der Studie ist somit die Feststellung der generellen Wahrnehmung der einzelnen Prozesse, die Erwartungshaltung und die tatsächliche Umsetzung, und die Implementierung von Verbesserungsprozessen, um mit erkannten Schwachstellen bzw. Verbesserungspotenzialen umzugehen.

Gegenstand des vorliegenden Beitrags ist der Teil des 3ProcSurvey, welcher sich mit der Nutzung von Vorgehensmodellen auseinandersetzt. Dieser Teil wird mit den Ergebnissen der zuvor durchgeführten IOSE-W² Studie gegenübergestellt.

¹ Hier ist der Unterschied zwischen gewünschter und tatsächlich durchgeführter Praxis gemeint.

3 Welche Vorgehensmodelle sind im Einsatz?

Vorgehensmodelle folgen wenigen allgemeinen Grundsätzen (in [BK13] als grundsätzliche Vorgehensmodelle bezeichnet). Die konkreten Vorgehensmodelle, die sich in der praktischen Anwendung befinden, greifen diese Grundsätze auf, verfeinern sie, und bieten Kombinationen davon an, etwa das phasenbasierte Vorgehen im Rational Unified Process (RUP), die hierarchische Dekomposition und die stufenweise Integration zum Gesamtsystem im V-Modell XT, oder das inkrementell/iterative Vorgehen in agilen Methoden wie Scrum. Im Folgenden widmen wir uns diesen konkreten Vorgehensmodellen und untersuchen, welche sich davon im praktischen Einsatz befinden.

3.1 Studienpopulationen

Als Datenquellen kommen die IOSE-W² Studie (2006) sowie die Teile des 3ProcSurvey (2012/2013) zum Einsatz, die sich mit dem Themenkomplex Vorgehensmodelle befassen. Beide Studien wurden mithilfe des Survey-Instruments [WR+12] als Fragebogen entworfen und kombinieren Single- und Multiple Choice Fragen, Freitext sowie Wertungsfragen, welche auf Likert-Skalen abgebildet wurden.

Die IOSE-W² Studie wurde mit etwa 65 Teilnehmern durchgeführt, der Teil des 3ProcSurveys, welcher sich auf die Vorgehensmodelle konzentriert, wurde von ca. 40-50 Teilnehmern² bearbeitet. Die Teilnehmer an beiden Studien deckten alle Unternehmensgrößen (0-20 bis hin zu über 2000 Mitarbeitern), sowie ein breites Tätigkeitsfeld (Softwareentwicklung, Dienstleistung, Beratung, usw. in unterschiedlichen Domänen, z.B. Informationssysteme, eingebettete Systeme, Automotive, Avionik usw.) ab. Aufgrund der vergleichsweise kleinen Datenbasis, können wir jedoch nicht nachweisen, dass die Daten repräsentativ für die deutsche Software-Branche sind.

3.2 Einzelergebnisse

Wir stellen nun die Ergebnisse der Umfragen vor. Zunächst greifen wir die Ergebnisse der IOSE-W² Studie aus dem Jahr 2006 auf. Im Anschluss stellen wir die Ergebnisse des 3ProcSurvey gegenüber und zeigen abschließend einige Trends auf.

3.2.1 Die IOSE-W2 Studie

Die IOSE-W² Studie adressierte unterschiedliche Aspekte, wie Vorgehensmodelle, verteilte Entwicklung, etc. Im Folgenden konzentrieren wir uns auf den Aspekt der Vorgehensmodelle. In der IOSE-W² Studie ging es u.a. darum, herauszufinden, welche Vorgehensmodelle, bzw. Vorgehensweisen zur Softwareentwicklung angewendet werden. Hierzu wurde in der Studie die folgende Frage gestellt:

² Im 3ProcSurvey war es möglich, einzelne Fragen auszulassen bzw. zu überspringen.

FI₁: Welche(s) der folgenden Entwicklungsmodelle/-konzepte findet/n bei Ihnen Anwendung?

In Abbildung 1 ist die Nennung (links) und die entsprechende prozentuale Verteilung (rechts; bezogen auf die Nennungen) zu sehen.

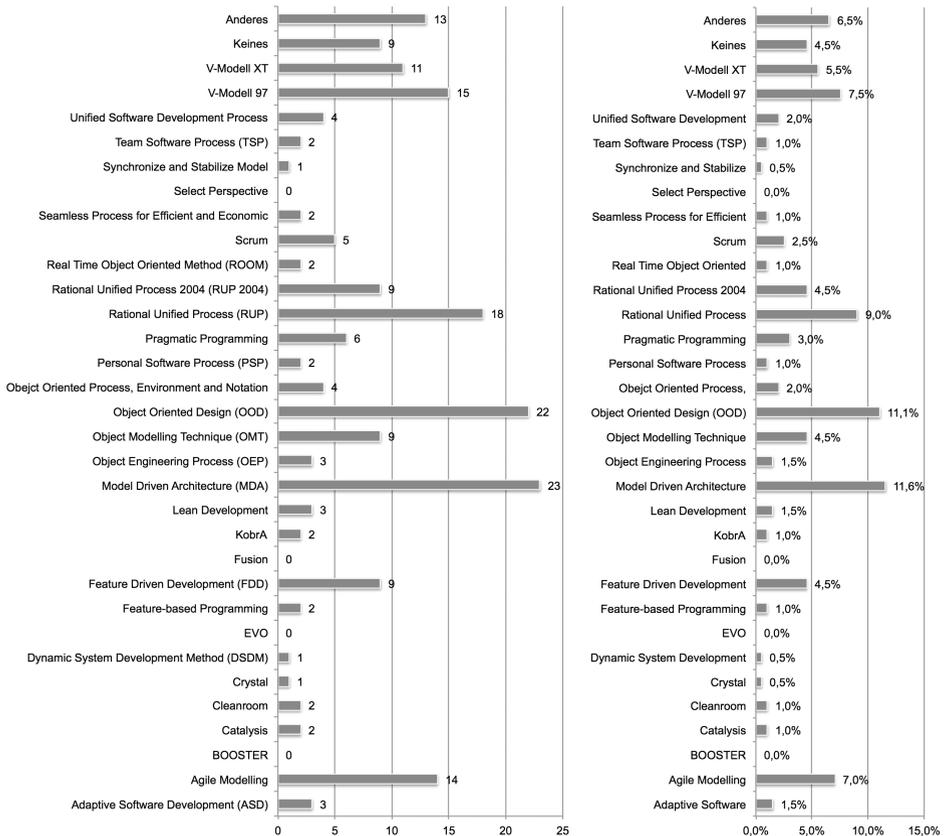


Abbildung 1: Nennung und prozentuale Verteilung von Vorgehensmodellen 2006

Die Teilnehmer hatten die Möglichkeit, mehrere der angebotenen Optionen zu wählen (Multiple Choice). Insgesamt wurde 13 Mal angegeben, dass „andere“ Vorgehensmodelle zum Einsatz kämen. Diese sind im Detail:

- Prototyping auf der Basis eines SAP Systems
- SAP-spez. Modelle (ASAP, PIL)
- Im Unternehmen selbst definierter Software-Entwicklungsprozess (2 Nennungen)
- HERMES
- Eigenes Vorgehensmodell auf Basis von internationalen Standards (RUP, PMI, UML) oder Kundenvorgaben/-vorgehen (3 Nennungen)
- Management der Softwareentwicklung, Carl Steinweg, Vieweg (2 Nennungen)

- eXtreme Programming (2 Nennungen)
- Catalan III

Auffällig ist die große Zahl an Modellen, die angegeben werden. Es zeigt sich somit, auch unter Berücksichtigung der „anderen“ Modelle, dass es für jeden Anwendungszweck ein passendes Vorgehensmodell zu geben scheint. Nur 9 Unternehmen (4,5%) gaben an, kein Vorgehensmodell einzusetzen.

Auf Basis der Umfragedaten ist auch zu sehen, dass oft keine „richtigen“ Vorgehensmodelle (i.S. eines konsolidierten und integrierten Vorgehensmodells nach [BK13]) zum Einsatz kommen. Vielmehr werden einzelne Methoden angewendet. Beispielsweise zählen OOD, OMT oder MDA eigentlich nicht zu den Vorgehensmodellen, sondern zu den Methoden.

Inwiefern diese Methoden mit anderen integriert sind, wurde im Rahmen der IOSE-W² Studie jedoch nicht explizit ermittelt. Stattdessen wurde erfragt, ob sich Unternehmen bewusst für den parallelen Einsatz mehrerer Vorgehensmodelle entschieden haben:

FI₂: *Gibt es unterschiedliche Vorgehensmodelle im Unternehmen?*

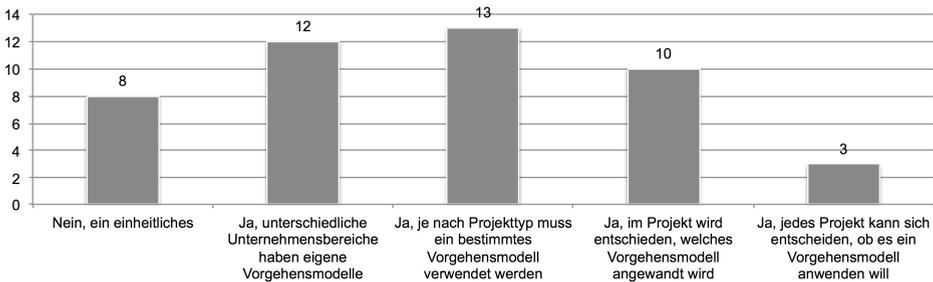


Abbildung 2: Werden unterschiedliche Vorgehensmodelle im Unternehmen angewendet? (2006)

In Abbildung 2 ist das Ergebnis der Frage FI₂ in Form von Nennungen dargestellt. Es ist zu sehen, dass lediglich 8 Unternehmen angeben, nur ein einheitliches Vorgehensmodell zu verwenden. Die verbleibenden Befragten gaben an, unterschiedliche Modelle zu verwenden, wobei die Gültigkeit des Vorgehensmodells sowohl Unternehmensbereiche wie auch einzelne Projekte umfassen kann.

Es stellt sich damit unmittelbar die Frage, wer die entsprechende Auswahl eines Vorgehensmodells trifft und nach welchen Kriterien:

FI₃: Wie erfolgt das Tailoring/Zuschneiden des Vorgehensmodells?

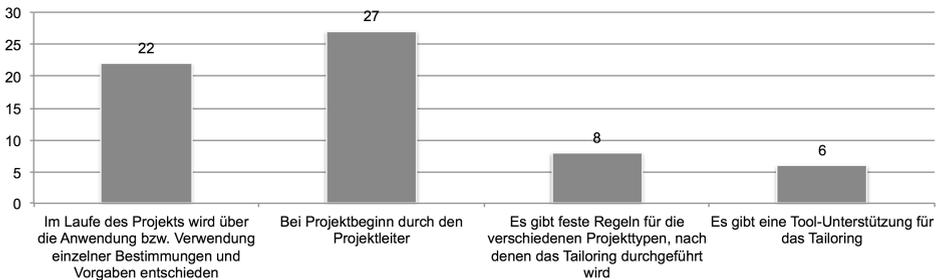


Abbildung 3: Implementierung des Tailorings in Unternehmen (2006)

Abbildung 3 zeigt die unterschiedlichen Ansätze beim Tailoring eines Vorgehensmodells. Hier geben 8 Unternehmen an, dass es für den Tailoring-Prozess feste Regeln gibt. 27 Befragte geben an, dass die Projektleiter das Tailoring zum Projektbeginn durchführen und 22 geben an, dass über die Verwendung einzelner Methoden erst während der Projektlaufzeit entschieden wird. Sechs Unternehmen geben an, dass für die Auswahl/Anpassung des Vorgehensmodells eine Werkzeugunterstützung vorliegt.

Interpretation. Die IOSE-W² Studie hat gezeigt, dass mehr als $\frac{3}{4}$ der befragten Unternehmen unterschiedliche Vorgehensmodelle einsetzen und diese ggf. kombinieren. Obwohl wenig überraschend, ist dennoch die Vielzahl der genannten Methoden und Vorgehensweisen erstaunlich. Ebenso fällt auf, dass in 2006 eine Vielzahl an Methoden genannt wurde, die im heutigen Verständnis gar nicht als selbständiges Vorgehensmodell gesehen werden. Ebenso überraschend ist die vergleichsweise geringe Präsenz agiler Methoden (5 Jahre nach dem Agilen Manifest kann man hier in der deutschen Wirtschaft noch eine deutliche Zurückhaltung vermuten).

Eine Vielzahl von Methoden und Vorgehensmodellen macht ein (systematisches) Tailoring erforderlich. Hier zeigt die IOSE-W² Studie jedoch, dass in der überwiegenden Mehrzahl keine definierten Regeln oder Werkzeuge zum Einsatz kommen. Üblicherweise wird die Auswahl des Vorgehensmodells dem Projektleiter überlassen. Bedenklich ist auch die hohe Anzahl an Befragten, welche die Vorgehensweise situationsabhängig und zur Projektlaufzeit auswählen. Den Prinzipien des *Situational Method Engineering* (SME; [HR+14]) folgend kann dies als gute Praxis angesehen werden. Ohne soliden Unterbau und ausreichend qualifizierte Projektteams kann dies jedoch auch schnell in ein ungesteuertes ad-hoc Vorgehen umschlagen; mit entsprechenden Auswirkungen auf die Kritikalität eines Projekts.

3.2.2 Der 3ProcSurvey

Im Folgenden stellen wir die Ergebnisse des 3ProcSurvey hinsichtlich der Vorgehensmodelle dar. Dieser Teil des 3ProcSurvey basiert auf der IOSE-W² Studie, weshalb die Fragen zwar nicht identisch sind, jedoch ausreichend Information für eine spätere Gegenüberstellung liefern.

F3₁: Welche(r) der folgenden Entwicklungsprozesse findet/n bei Ihnen Anwendung?

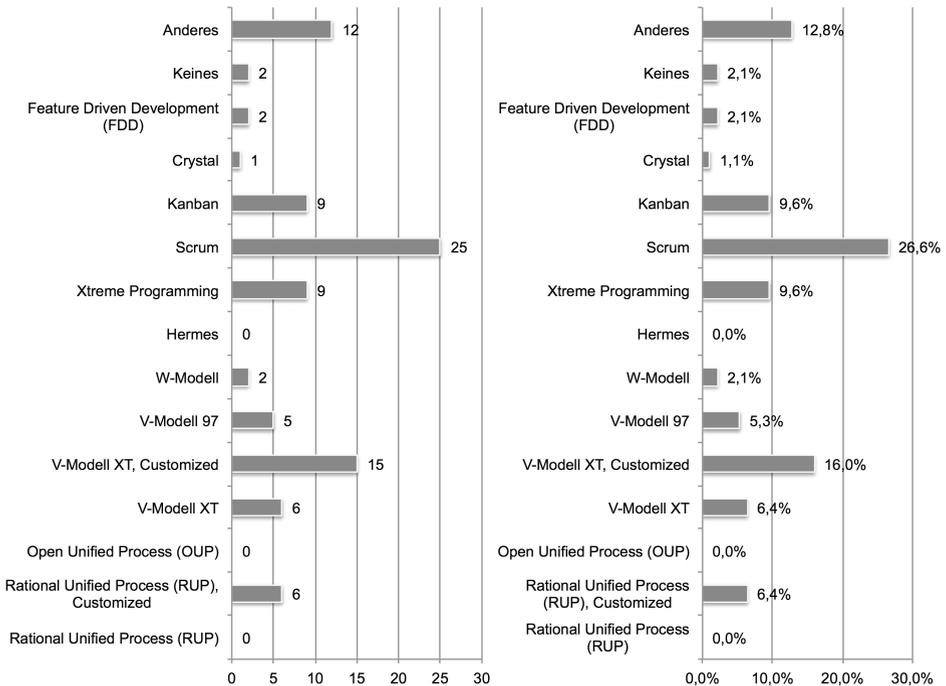


Abbildung 4: Nennung und prozentuale Verteilung von Vorgehensmodellen (2012/13)

Abbildung 4 zeigt die Nennungen (links) und die entsprechende prozentuale Verteilung (rechts; bezogen auf die Nennungen, eine Mehrfachauswahl war möglich) hinsichtlich der eingesetzten Vorgehensmodelle. Im Gegensatz zur IOSE-W² Studie wurde die Menge an Optionen für die Befragten eingeschränkt. Es konnten jedoch weiterhin „andere“ Vorgehensmodelle genannt werden. Hierbei ergaben sich folgenden Nennungen:

- eigenes V-Modell, angelehnt an DO-178
- Inkrementelles V-Modell
- Test Driven Development
- Selbst definiert/entwickelt, Mixtur (4 Nennungen)
- KOMPASS
- Evo
- Auftraggeber-spezifischer/-abhängiger Prozess (2 Nennungen)

Auch hier fällt auf, dass die befragten Unternehmen unterschiedliche Methoden bzw. Vorgehensmodelle einsetzen und auch zu einem erheblichen Anteil (12,8%) nicht auf standardisierte Vorgehensmodelle zurückgreifen. Jedoch zeigen die Daten des 3ProcSurvey nicht mehr die hohe Breite an Methoden. Auffällig ist hierbei insbesondere, dass Scrum mit 26,6% die am häufigsten eingesetzte Methode ist.

F3₂: Haben Sie einen unternehmensweiten Standardprozess für die SW-Entwicklung?

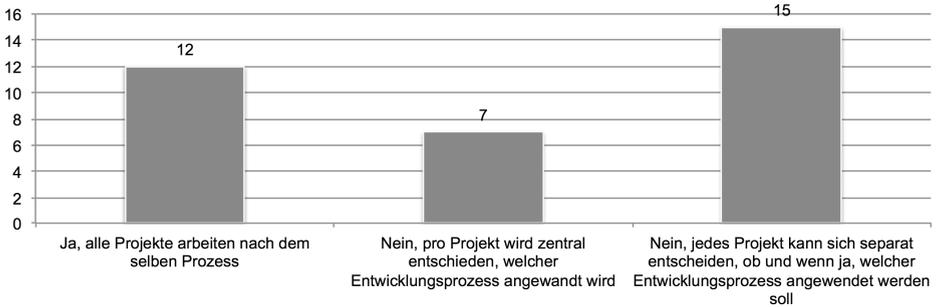


Abbildung 5: Werden unterschiedliche Vorgehensmodelle im Unternehmen angewendet? (2012/13)

Analog zur IOSE-W² Studie wurden die Unternehmen auch befragt, ob sie mehrere Vorgehensmodelle, bzw. einen Standard einsetzen, und ob ein eventuell eingesetzter Standard verpflichtend für alle Projekte ist? Abbildung 5 zeigt, dass nur 12 der befragten Unternehmen einen Standard einsetzen, welcher für alle Projekte des Unternehmens verbindlich ist. Abschließend wurde die Frage gestellt, wie die Auswahl/Anpassung des Vorgehensmodells an die jeweiligen Projekte erfolgt.

F3₂: Wie erfolgt das Tailoring (Zuschneiden) des Entwicklungsprozesses?

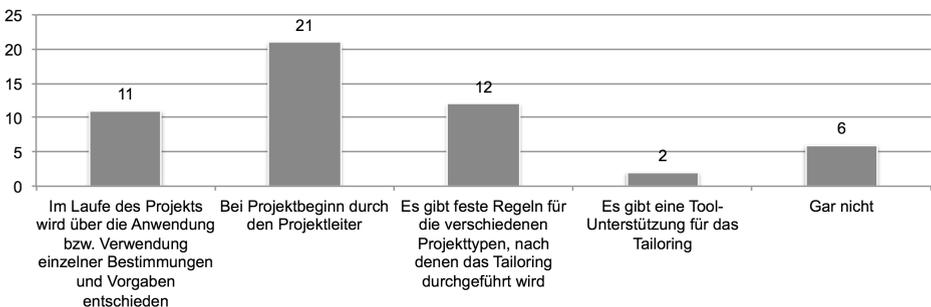


Abbildung 6: Implementierung des Tailorings in Unternehmen (2012/13)

Zwölf der Befragten geben an, dass es feste Regeln für das Tailoring gibt, und 2 Befragte geben an, dass es für das Tailoring eine Werkzeugunterstützung gibt (Abbildung 6). Überwiegend wird angegeben (11 und 21 Nennungen), dass das Tailoring entweder durch den Projektleiter zu Beginn des Projekts durchgeführt wird, oder situationspezifisch während der Projektlaufzeit. Sechs der Befragten geben sogar an, dass ein Tailoring gar nicht durchgeführt wird.

Interpretation. Die Ergebnisse des 3ProcSurvey (Bereich: Vorgehensmodelle) zeigen, dass auch heute noch eine Vielzahl von Vorgehensmodellen Anwendung findet, wenn auch eine gewisse Konsolidierung festzustellen ist. Es zeigt sich jedoch ein recht klarer

Trend zu Scrum- und V-Modell-XT-basierten Prozessen, wobei der Rational Unified Process scheinbar keine Relevanz mehr hat. Bei dieser Vielfalt stellt sich auch wieder die Frage nach Auswahl und Anpassung. Hier zeigt sich, dass überwiegend projektspezifisch durch die Projektleiter entschieden wird, welches Vorgehensmodell zum Einsatz kommt. Eine Vergleichbarkeit einzelner Projekte auf Prozessbasis wird somit jedoch beeinträchtigt. Ebenfalls fällt auf, dass das Tailoring Großteils ohne zentrale Vorgaben erfolgt, oder, wie auch zu sehen war, gar nicht. Dies legt auch nahe, dass die projektspezifische Anpassung unter Umständen willkürlich erfolgen kann, was das Risiko birgt, dass die „freihändige“ Anpassung die innere Logik des Vorgehensmodells zerstört und Vorteile des Vorgehensmodells gar nicht mehr genutzt werden können.

3.3 Vergleich und Trends

Da der 3ProcSurvey (2012/2013) auf der IOSE-W² (2006) Studie aufbaut, bietet sich eine erste Sichtung von möglichen Trends an³. Hierzu betrachten wir zunächst die Entwicklung der Anwendung von einigen der ermittelten Vorgehensmodelle. Da die Menge an Vorgehensmodellen vergleichsweise groß war, wurden für diese Analyse diejenigen Vorgehensmodelle gruppiert und untersucht, welche in beiden Umfragen vorkamen (siehe Anhang).

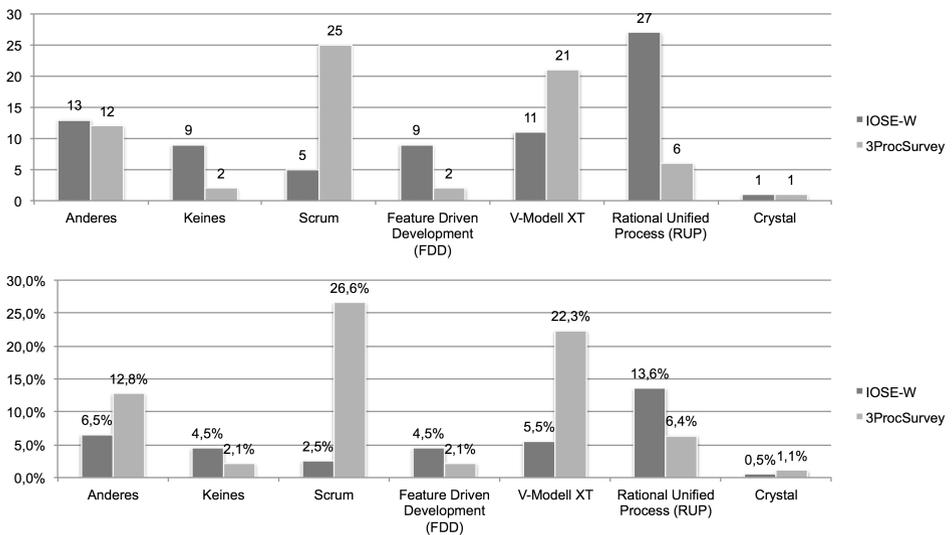


Abbildung 7: Entwicklung der Anwendung zwischen 2006 und 2012/2013

Abbildung 7 zeigt für Scrum, Crystal, V-Modell XT (inkl. Anpassungen), sowie RUP (inkl. Anpassungen) die Entwicklung der Anwendung von 2006 nach 2012/2013. Auffallend sind insbesondere der deutliche Rückgang des RUP auf der einen, sowie die ebenso deutliche Zunahme von Scrum und V-Modell XT auf der anderen Seite. Diese beiden

³ **Hinweis:** Aufgrund der noch kleinen Datenbasis, kann an dieser Stelle noch keine vollständige Trendanalyse erfolgen. An dieser Stelle werden nur erste Tendenzen betrachtet.

Prozesse/Prozessfamilien stellen zusammen fast die Hälfte aller Antworten. Ebenso deutlich ist auch der Anstieg der „anderen“ Vorgehensmodelle bei gleichzeitigem Rückgang von „kein Vorgehensmodell“. Hier zeigt sich ein erster Trend hin zu mehr Prozessorientierung, wobei durch den Anstieg der „anderen“ Vorgehensmodelle deutlich wird, dass Firmen eigene Entwicklungen anstreben.

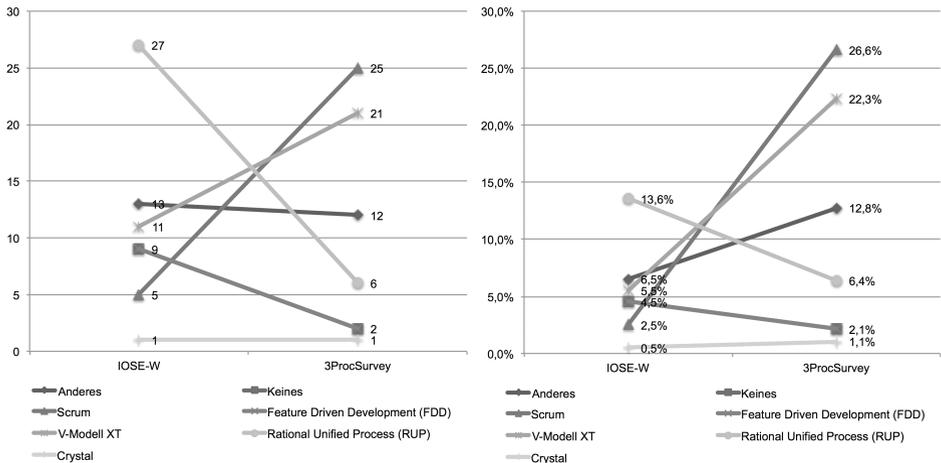


Abbildung 8: Trendentwicklung absolut (links) und prozentual (rechts)

In Abbildung 8 ist diese Entwicklung noch einmal in der direkten Gegenüberstellung zu sehen. Hier wird noch einmal der deutliche Anstieg von Scrum und V-Modell XT deutlich, sowie der starke Einbruch des Rational Unified Process.

Der in Abbildung 8 zu erkennende starke Anstieg der Anwendung von Scrum führt zu der Frage, ob in Deutschland der Trend weg von den reichhaltigen, strukturierten Prozessen (vgl. [BK13]) hin zu den agilen Ansätzen geht. Dazu wurden alle genannten Vorgehensmodelle (sofern eindeutig zuzuordnen) in die Kategorien „Agile“ (z.B., Scrum, Crystal, XP) und „Rich“ (z.B. RUP, V-Modell XT, HERMES) eingeordnet und analysiert (siehe Abbildung 9). Erstaunlich hierbei ist, dass hinsichtlich der absoluten Nennungen ein klarer Trend weg von den reichhaltigen Prozessen zu sehen. Prozentual hingegen ist jedoch festzustellen, dass auch diese Art von Prozessen immer noch auf Interesse stößt. Deutlich hingegen fällt die Entwicklung hin zu den agilen Ansätzen auf; hier ist ein Wachstum von (relativ) über 30% zu sehen.

Dies zu erklären ist schwer, jedoch könnte eine Ursache in der Wahrnehmung „professioneller Softwareentwicklung“ liegen: Wer heutzutage kein Vorgehensmodell verwendet, setzt sich schnell dem Vorwurf aus, nicht professionell zu arbeiten. Insofern ist ein gewisser Druck vorhanden, einen Entwicklungsprozess einzuführen. Hierbei ist z.B. die Einführung eines V-Modell-XT-basierten Vorgehens inhärent mit Aufwand verbunden, wohingegen die Einführung eines agilen Vorgehens *scheinbar* einfacher und günstiger ist. Führt man einen agilen Prozess ein, liegt es natürlich näher, z.B. das mit großen Freiheitsgraden versehene Scrum-Framework einzuführen, als das rigide eXtreme Pro-

gramming. Aber auch hier besteht die Möglichkeit, dass viele Fälle existieren, in denen Scrum nur unvollständig oder halbherzig umgesetzt wird (auch als „ScrumBut“ bezeichnet). Da Scrum keine Vorgaben bezüglich der eigentlichen Entwicklung macht, könnten sich hier auch Situationen ergeben, in denen die eigentliche Entwicklung mehr oder weniger chaotisch verläuft (vgl. [PC86]).

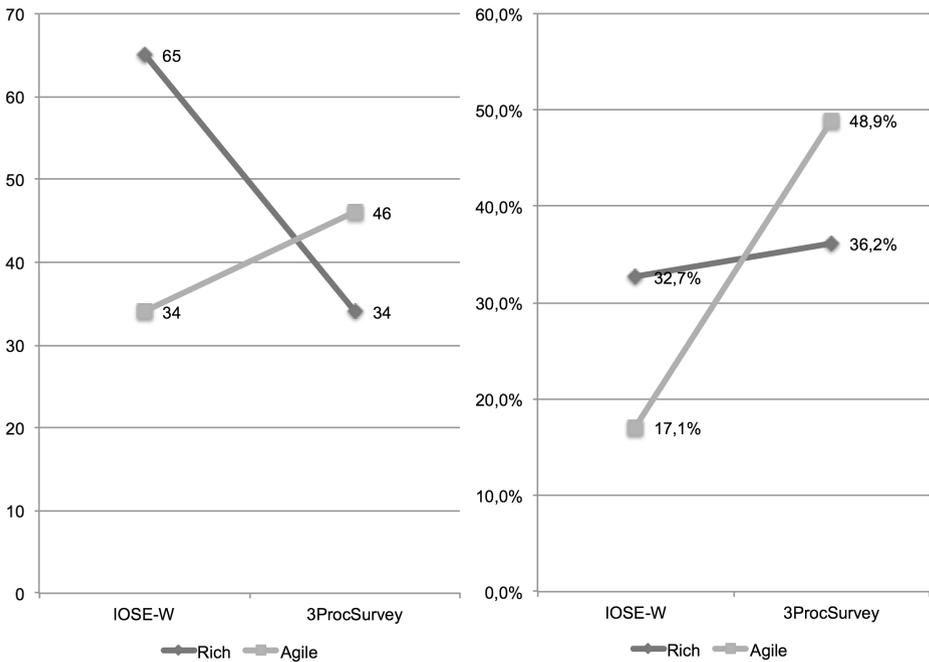


Abbildung 9: Trendentwicklung nach agilen und reichhaltigen Prozessen

Werden die Ergebnisse der IOSE-W² Studie und des 3ProcSurvey nebeneinander gelegt, fallen folgenden Entwicklungen auf:

1. Das Interesse an Prozessen nimmt zu (Rückgang bei „kein Vorgehensmodell“).
2. Agile Ansätze erfreuen sich mittlerweile großer Beliebtheit.
3. Firmen haben i.d.R. mehr als ein Vorgehensmodell im Einsatz (es gibt eine große Bandbreite angewendeter Vorgehensmodelle).
4. Auswahl und Anpassung von Vorgehensmodellen erfolgt i.d.R. individuell und ohne organisatorische Standards oder Werkzeugunterstützung.

4 Zusammenfassung und Ausblick

In diesem Papier haben wir der Frage nachgegangen, welche Vorgehensmodelle in Deutschland eingesetzt werden. Hierbei konnten wir auch erste Trends über einen Zeitraum von ca. 6 Jahren ermitteln, welche aus einem Vergleich der Ergebnisse der IOSE-

W² Studie von 2006 und Teilen des 3ProcSurvey von 2012/2013 resultieren. Hier ist aktuell zu beobachten:

- Die verwendeten Prozessmodelle weisen eine hohe Streuung auf. Es kann nicht davon gesprochen werden, dass ein bestimmter Ansatz in der Praxis dominiert.
- Agile Ansätze haben eine hohe Verbreitung in der Praxis, sind aber nicht so dominierend, wie man aufgrund der IT-Presse vermuten könnte.
- Die meisten Organisationen verwenden mehrere Prozessmodelle.
- Die Anpassung eines Prozessmodells an die konkrete Projektsituation erfolgt nicht auf der Basis definierter Regeln.

Die Studie hat gezeigt, dass die projektspezifische Anpassung der Vorgehensmodelle der Normalfall ist. In fast der Hälfte aller Fälle kann auf Projektebene entschieden werden, ob man sich an die Vorgaben eines Vorgehensmodells hält oder nicht. Dies birgt mindestens die Gefahr, dass Vorgehensmodelle als Luxusfeature verwendet werden, die man leichthin im Projektalltag ignorieren kann – wenn man dies möchte. Dabei soll niemandem böser Wille unterstellt werden, aber diese Praxis kann durchaus als problematisch betrachtet werden. Ein Beispiel einer unsachgemäßen Anpassung eines Vorgehensmodells könnte sein, dass in einer Phase durch das Tailoring ein Artefakt nicht mehr geprüft wird und es ungeprüft in einer späteren Phase als Prozess-Input verwendet wird.

Eine Überprüfung dieser und weiterer Vermutungen ist Gegenstand der weiterführenden, detaillierten Analysen und Vergleiche der Umfragen, etwa hinsichtlich der Kontrolle von Prozessen in Organisationen/Projekten, hinsichtlich der Einbindung in weitere Prozesse und Managementstandards, oder die Umsetzung von Prozessverbesserungsprogrammen.

Die 2012/13'er Instanz des 3ProcSurvey diene einerseits dem Neuaufsetzen eines Instruments zur Bestimmung des Status der Softwareentwicklung in Deutschland und darüber hinaus der ersten Datengewinnung. Es ist nun erforderlich, die Datenbasis zu verbreitern, um aus den oben beispielhaft aufgeführten Vermutungen überprüfbare Hypothesen abzuleiten. Dieser Beitrag legt somit auch die Grundlage für die nächste Runde des 3ProcSurvey, zu dem alle Leser herzlich eingeladen sind.

Danksagung

Die Ergebnisse, welche hier vorgestellt wurden, sind unter Mitwirkung vieler Personen entstanden, die wir hier nennen möchten. Zunächst möchten wir Patrick Keil und Martin Fritzsche erwähnen, die maßgeblich an der Erstellung der IOSE-W² Studie beteiligt waren. Im Kontext des 3ProcSurvey sind insbesondere Frank Simon, Annette Koßmann und Daniel Mendéz Fernández zu nennen. Wir bedanken uns insbesondere auch beim BITKOM und der Gesellschaft für Informatik (GI) e.V. für die Bereitstellung von Kontakten und Kommunikationskanälen, ohne die diese Studie nicht durchführbar gewesen wäre.

Anhang

An dieser Stelle listen wir die Datentabellen auf, welche die Grundlage für diesen Beitrag sind.

Tabelle 1: Nennungen von Vorgehensmodellen/Methoden der IOSE-W-Studie

Modell	Nennung
Adaptive Software Development (ASD)	3
Agile Modelling	14
BOOSTER	0
Catalysis	2
Cleanroom	2
Crystal	1
Dynamic System Development Method (DSDM)	1
EVO	0
Feature-based Programming	2
Feature Driven Development (FDD)	9
Fusion	0
KobrA	2
Lean Development	3
Model Driven Architecture (MDA)	23
Object Engineering Process (OEP)	3
Object Modelling Technique (OMT)	9
Object Oriented Design (OOD)	22
Object Oriented Process, Environment and Notation (OPEN)	4
Personal Software Process (PSP)	2
Pragmatic Programming	6
Rational Unified Process (RUP)	18
Rational Unified Process 2004 (RUP 2004)	9
Real Time Object Oriented Method (ROOM)	2
Scrum	5
Seamless Process for Efficient and Economic Development (SPEED)	2
Select Perspective	0
Synchronize and Stabilize Model	1
Team Software Process (TSP)	2
Unified Software Development Process	4

Modell	Nennung
V-Modell 97	15
V-Modell XT	11
Keines	9
Anderes	13
Summe	199

Tabelle 2: Nennungen von Vorgehensmodellen/Methoden des 3ProcSurvey

Modell	Nennung
Rational Unified Process (RUP)	0
Rational Unified Process (RUP), Customized	6
Open Unified Process (OUP)	0
V-Modell XT	6
V-Modell XT, Customized	15
V-Modell 97	5
W-Modell	2
HERMES	0
Xtreme Programming	9
Scrum	25
Kanban	9
Crystal	1
Feature Driven Development (FDD)	2
Keines	2
Anderes	12
Summe	94

Literaturverzeichnis

- [BEJ06] Buschermöhle, R., Eeckhoff, H., Josko, B.: SUCCESS – Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Software-Entwicklungsprojekten in Deutschland. BIS-Verlag, 2006
- [BK13] Broy, M., Kuhrmann, M.: Projektorganisation und Management im Software Engineering. Springer, 2013
- [DS+13] Dorst, W., Simon, F., D'Onorio De Meo, M., Luckhaus, S., Tewes, U., Schneider, C., Gärtner, M., Kuhrmann, M.: Agile Software Engineering Made in Germany. Technical Report, BITKOM, 2013
- [FK07] Fritzsche, M., Keil, P.: Kategorisierung etablierter Vorgehensmodelle und ihre Verbreitung in der deutschen Software-Industrie. Technical Report TUM-I0717, Technische Universität München, 2007
- [HR+14] Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J., Rossi, M.: Situational Method Engineering. Springer, 2014
- [KLS11] Kuhrmann, M., Lange, C., Schnackenburg, A.: A Survey on the Application of the V-Modell XT in German Government Agencies. In Proceedings of the 18th Conference on European System & Software Process Improvement and Innovation (EuroSPI), pp. 49 ff., 172, Springer Verlag, 2011
- [Mah08] Mah, M.: How Agile Projects Measure Up, and What This Means to You. Technical Report, Cutter Consortium, 2008
- [MW13] Mendéz Fernández, D., Wagner, S.: Naming the Pain in Requirements Engineering: Design of a Global Family of Surveys and First Results from Germany. In Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering (EASE '13), ACM, 2013
- [PC86] Parnas, D. L., Clements, P. C.: A Rational Design Process: How And Why To Fake It. IEEE Transactions on Software Engineering, 12(2):1-10, 1986
- [SK+13] Simon, F., Kossmann, A., Kuhrmann, M., Mendéz Fernández, D.: Wunsch oder Wirklichkeit? Professionelle Softwareentwicklung „Made in Germany“. In OBJEKTSpektrum, pp. 16-23, Sigs Datacom, 2013
- [Sta06] Standish Group International: Chaos Reports. Online: <http://www.standishgroup.com>, 2006 (und folgende Jahre)
- [WR+12] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, 2012

Teil II
Hauptprogramm

Von der Langstrecke zum Sprint – Agile Methoden in traditionellen Unternehmen

Bodo Kraft, Axel Zöll

kraft@fh-aachen.de
axel.zoell@generali.com

Abstract: In einem Unternehmen, das in der Linienstruktur traditionell stark hierarchisch und auf der Projektebene stark rollenorientiert geprägt ist, verursacht jede Veränderung dieses Gefüges zunächst Unsicherheit und Widerstände.

Aus Basis der Erfahrungen, die bei der Generali Deutschland gesammelt wurden, werden in diesem Artikel die Herausforderungen für einen nachhaltig erfolgreichen Einsatz agiler Methoden im Konzernverbund beschrieben. Der Fokus liegt hierbei auf den inhaltlichen und sozialen Aspekten, die bei einer neuen Rollenprägung aller Beteiligten erkennbar sind.

1 Fortlaufender Wandel der IT in der Versicherungsbranche

1.1 Sicht auf die Unternehmensstruktur

Die Generali Deutschland Gruppe ist die zweitgrößte Erstversicherungsgruppe in Deutschland. Unter dem Dach der Generali Deutschland Holding arbeiten namhafte Versicherer und Finanzdienstleistungsunternehmen wie AachenMünchener, Generali Versicherungen, CosmosDirekt, Central Krankenversicherung, Advocard Rechtsschutzversicherung, Deutsche Bausparkasse Badenia und die Dialog, die das gesamte Spektrum der Finanzdienstleistung anbieten. Die Generali Deutschland Informatik Services (GD-Informatik) als zentraler Informatikdienstleister realisiert die IT-Projekte, den Betrieb der Versicherungssysteme und die Unterstützung der Innen- und Außendienstmitarbeiter durch Hard- und Software. Das Projektvolumen liegt bei ca. 100.000 Personentagen pro Jahr, die in rund 60 Projekten unterschiedlichster Größe und Struktur geleistet werden.

1.2 Strategische Ausrichtung der Unternehmen

Der Versicherungsmarkt ist in Europa weitestgehend gesättigt. Die erwarteten Ertragssteigerungen lassen sich somit nicht durch Wachstum sondern nur durch Einsparung und Effizienzsteigerung realisieren. Gleichzeitig wurde das Portfolio für IT-Projekte in den

letzten Jahren stark von Pflichtmaßnahmen, die rechtlich oder regulatorisch vorgegeben wurden, in Beschlag genommen.

Aus dieser Ausrichtung ergibt sich unmittelbar ein Interessenskonflikt. Während Auftraggeber und IT verpflichtet sind, die IT Aufwände gering zu halten, möchte der im Projekt eingebundene Fachbereich – selbst wenn er aus derselben Organisation stammt wie der Auftraggeber – maximale Funktionalität.

Somit verbleibt immer weniger Spielraum innerhalb des Konzerns, um Geschäftsprozesse zu optimieren oder Produkte zu entwickeln. Es ist also von größerer Bedeutung als jemals zuvor, die richtigen Dinge zu entwickeln, d.h. Blindleistung zu vermeiden und die weniger wichtigen Anforderungen gar nicht erst in Auftrag zu geben. Auf Portfolio- und auf Projektebene gilt die Zielsetzung, die Entwicklung so effizient wie möglich durchzuführen.

1.3 Methodische Entwicklung in den letzten Jahren

Betrachtet man die Art und Weise, wie IT-Projekte in der Generali Deutschland Gruppe durchgeführt werden, kann man mehrere Epochen ausmachen, die sich in Methode, Kultur und Zielsetzung deutlich voneinander abgrenzen, zeitlich jedoch einander überlappen.

Vor 2006 wurde jedes Projekt individuell durchgeführt. In jedem einzelnen Projekt wurde die Methode immer wieder neu erfunden. Gemäß einer Charakterisierung im Umfeld CMMI bezeichnen wir diese Epoche als das **Zeitalter der Helden**.

Von 2006 an wurde der Rational Unified Process (RUP) [IB] für die Generali Deutschland Gruppe adaptiert und als Standardvorgehen etabliert (Advanced Generali Software Engineering Method, AGSM). Wesentliches Element war die Standardisierung der Rollen im Projekt wie Integrator oder fachlicher Tester, die dann durch entsprechende Qualifizierung und Zertifizierung ausgebildet und gestaffed wurden. In 2012 wurde die Standardisierung und der kontinuierliche Verbesserungsprozess durch ein externes Audit gemäß SCAMPI Appraisal A verifiziert und CMMI Level 3 [SE] erlangt. Es war das **Zeitalter der Standards**.

Bereits zu diesem Zeitpunkt war offensichtlich, dass das angemessene Maß an Standardisierung und Formalismus überschritten war. Zudem war die praktische Anwendung des RUP kaum agil ausgeprägt, obwohl es der Standard erlaubte. Die Projektvorgehensweisen wurden optimiert, indem man nach Mustern suchte und den Standard optimierte. Es war das **Zeitalter der Varianten**.

In der letzten Epoche beginnend in 2013 verschob sich der Schwerpunkt von der Optimierung über den Standard hin zur Optimierung der Anwendung des Standards. Dazu bedient man sich des Werkzeugkastens von Lean und strebt die Kombination klassischer und agiler Ansätze an [KK12]. Ob dieses **Zeitalter** mehr von **Lean** als von **agile** geprägt wird, muss sich noch herausstellen.

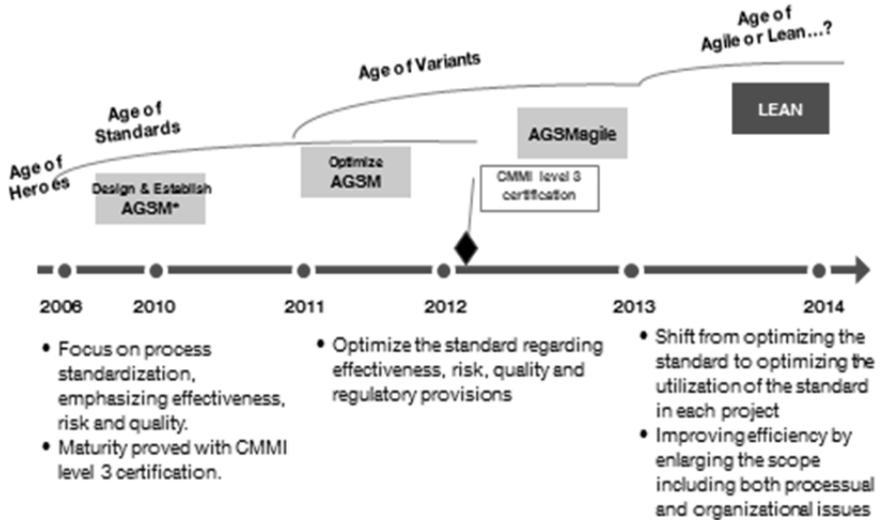


Abbildung 1: Methodische Entwicklungsstufen in der Übersicht

In jedem Fall bildete die Entwicklung einer agilen Vorgehensweise die Brücke. Aus der Sicht der Epoche „Optimiere den Standard“ wurde eine neue schlankere Variante für den Standard angeboten: SCRUM [SE] und RUP wurden kombiniert („AGSM agile“) unter Beachtung der Ansprüche von CMMI Level 3. Gleichzeitig wurden genau die Elemente eingeführt, die für die jüngste Epoche „Optimiere die Anwendung des Standards“ kennzeichnend sind: jedes Team organisiert sich selbst und optimiert sich kontinuierlich.

Jede Epoche förderte andere Handlungsweisen und hatte seinen Protagonisten.

- Der **Held** fühlt sich durch Standards beschnitten und im agilen Vorgehen zu lästigen Abstimmungen verpflichtet.
- Der **Spezialist** aus dem Zeitalter der Standards wird aufgrund seiner Kompetenz bestaunt, wird aber unter Helden zum belächelten, unverständenen und bestenfalls ignorierten Rufer in der Wüste. Im agilen Umfeld wird er tendenziell als der konservative Perfektionist und Formalist wahrgenommen, der nur widerstrebend andere an seinem Know-how teilhaben lässt. Er bringt sich selbst dann nicht in Themen, die nicht seiner Spezialisierung entsprechen, wenn es dort Not tut.
- Dem agilen **Teamplayer** fehlen im Zeitalter der Helden die Verbindlichkeit und das Team, das gemeinsam lernt und sich verbessert. Im Zeitalter der Standards wird ihm sowohl der flexible Einsatz nicht ermöglicht als auch der Spielraum zur Optimierung genommen.

2 Eingriff in die Unternehmens- und Projektrollen

In einem Unternehmen, das in der Linienstruktur traditionell stark hierarchisch und auf der Projektebene stark rollenorientiert geprägt ist, verursacht jede Veränderung dieses Gefüges zunächst Unsicherheit und Widerstände.

Ebenso erfordert jede Veränderung der im Konzernverbund abgestimmten Rollen, Verantwortlichkeiten und Schnittstellen die Zustimmung vieler Beteiligten.

In diesem Abschnitt werden die Herausforderungen für einen erfolgreichen Rollout und den nachhaltig erfolgreichen Einsatz agiler Methoden in Konzernverbund beschrieben. Der Fokus liegt hierbei auf den inhaltlichen und sozialen Aspekten, die bei einer neuen Rollenprägung aller Beteiligten erkennbar sind.

2.1 Veränderung der Projektrollen

Der gewählte Ansatz, eine angepasste SCRUM Variante als Ergänzung zur klassischen Entwicklungsmethode anzubieten, erfordert die Einführung der agilen Rollen Scrum Master (SM), Product Owner (PO) und Scrum Team.

Aus methodischer Sicht ergibt sich hier eine Verschiebung der Aufgaben, die in klassischen Projekten über das Rollenkonzept bereits gut beschrieben sind.

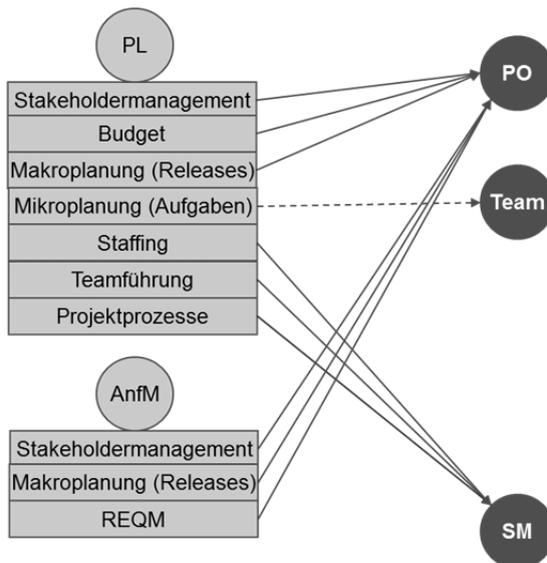


Abbildung 2: Verschiebung der Aufgaben

Die Abbildung zeigt einen Ausschnitt der Aufgaben der klassischen Führungsrollen Projektleiter (PL) und Anforderungsmanager (AnfM) und deren Verschiebung in einem agilen Projekt.

- Der Product Owner bedient im Wesentlichen die Schnittstellen nach Außen, sowie die grobe Projektsteuerung in den Dimensionen Inhalt, Budget und Zeit.
- Das Team erhält die zusätzliche Aufgabe der eigenverantwortlichen Mikroplanung.
- Der Scrum Master übernimmt die Aufgaben des Projektleiters im Bereich der Teamführung und der internen Steuerungsprozesse.

Zu ergänzen ist die Rolle der disziplinarischen Führungskraft, die in der klassischen Variante in den meisten Fällen die Rolle des Paten in dem zugeordneten Lenkungsausschuss übernimmt.

Bei konsequenter Anwendung dieser Rollenverschiebung besteht weder in der Organisation, noch im Projekt ein weiterer Bedarf, die klassischen Projektrollen beizubehalten. In der Praxis zeigt sich jedoch, dass die bestehenden klassischen Rollen auch in den agilen Projekten nicht durchgängig aufgeben werden. Es entstehen Konflikte in der Aufgaben-, Kompetenz- und Machtverteilung.

2.2 Soziale Prägung der Projektrollen

Neben der rein aufgabenorientierten Sichtweise auf die Rollenverteilung wird im Folgenden ein Blick auf die sozialen Aspekte der veränderten Rollenstruktur geworfen. Hierbei wird geprüft, in wie weit die beschriebene Rollenverschiebung kompatibel mit der sozialen Prägung der aktuellen Rolleninhaber ist.

2.2.1 Führungsebene

Die passende Besetzung der Führungsrollen (Product Owner und Scrum Master) in einem agilen Team ist ein kritischer Erfolgsfaktor. In hierarchischen Organisationen werden typischerweise Projektleiter und Linienführungskräfte entwickelt, die einen starken Bezug zu disziplinarischer Macht haben.

Die Besetzung der agilen Führungsrollen stellt damit eine Herausforderung dar.

Der Scrum Master ist der entscheidende Treiber für die Veränderungen in einem agilen Projekt und seinem Umfeld. Ohne formal zugewiesene disziplinarische oder fachliche Führungskompetenz leitet der Scrum Master den Prozess der kontinuierlichen Verbesserung. Hierbei agiert der Scrum Master eher dezent im Hintergrund als Coach, Reviewer oder Moderator und nicht als Vorgesetzter. Damit erreicht er, dass die intrinsische Motivation erhalten bleibt und sich die Beteiligten nicht der Eigenverantwortung entziehen [Pi11].

Trotz der enormen Bedeutung für den Erfolg des gesamten Teams stellt es sich als schwierig heraus, Führungskräfte und Projektleiter für diese Rolle zu gewinnen.

Aufgrund der oben beschriebenen hierarchischen Strukturen ist die persönliche Präsenz und die Möglichkeit zur Selbstvermarktung ein wichtiges Merkmal für erfolgreiche Führungskräfte. Inhaber dieser Rollen zeichnen sich vielfach aus durch eine ergebnisorientierte Arbeitsweise mit eher dominanter Prägung. In den meisten Fällen sind diese Rollen zudem auch in Karrieremodellen eingebettet.

Für diesen Personenkreis, der aufgrund seiner Qualifikation für Führungsaufgaben passend erscheint, ist die Rolle des Scrum Masters wenig attraktiv. Ebenso erscheint auch den Linienvorgesetzten die Rolle des Scrum Masters oftmals nicht als adäquate Position für die von ihnen betreuten Projektleiter. Im Sinne der Personalentwicklung wird die Position des Scrum Masters, der ohne formale Macht ausgestattet ist, eher als Rückschritt für einen Projektleiter gesehen. Der gestandene Projektleiter erscheint als überqualifiziert und zu teuer.

In der Folge werden häufig Mitglieder des Entwicklungsteam als Scrum Master eingesetzt. Auch wenn in Ausnahmen diese Besetzung erfolgreich sein kann, fehlt in den meisten Fällen die Qualifikation Team- und Veränderungsprozesse erfolgreich zu gestalten und zudem die Vernetzung auf der Ebene der Entscheider.

Die in Blogs aktuell geführten Diskussionen zur geringen Durchsetzungskraft des Scrum Masters in vielen Teams belegt die These, dass oftmals falsch qualifizierte Scrum Master eingesetzt werden [G11, C11].

Der Product Owner ist verantwortlich für den langfristigen wirtschaftlichen Erfolg eines Produkts. In vielen Unternehmen verantwortet die disziplinarische Führungskraft die langfristige und strategische Ausrichtung eines Systems oder Produkts. Die konkrete Weiterentwicklung wird in diesen Fällen üblicherweise in Projektform umgesetzt.

Das Aufgabenspektrum des Product Owners setzt sich wie oben beschrieben im Wesentlichen zusammen aus Aufgaben des Projektleiters und des Anforderungsmanagers. Zudem übernimmt er Aufgaben der disziplinarischen Führungskraft im Bereich der langfristigen Produktentwicklung. Im Kreis der relevanten Stakeholder ermittelt er die Anforderungen. Er allein hat die Autorität über die Umsetzungsreihenfolge zu entscheiden.

In der Praxis zeigt sich häufig, dass die disziplinarische Führungskraft nicht bereit ist, die Verantwortung für die strategische Ausrichtung des Produkts abzugeben. Aufgrund der oben beschriebenen Gründe sind in einer hierarchischen Organisation auch diese Rolleninhaber angewiesen auf die Sichtbarkeit innerhalb der Organisation.

In der Folge übernimmt die disziplinarische Führungskraft oftmals die Rolle des Product Owners. Die Rolle ermöglicht die angestrebte strategische Steuerung und schafft Sichtbarkeit. Die umfangreichen operativen Tätigkeiten, die dieser Rolle zugeordnet sind, kann eine disziplinarische Linienführungskraft mit eigenem Aufgabenspektrum allerdings kaum leisten.

Als Lösung wird die Rolle des Product Owners häufig doppelt besetzt. Oftmals wird hierbei die alte Struktur von Projektleiter plus Linienführungskraft verwendet. Auch

wenn diese Konstellation prinzipiell erfolgreich sein kann, kommt es dennoch häufig zu vermehrten Abstimmungsaufwänden, Kompetenzunklarheiten und Missverständnissen.

2.2.2 Team

Beim Übergang von einer klassischen Projektsteuerung zu einem agilen Arbeiten wird das Entwicklungsteam typischerweise beibehalten. Die Herausforderung bei einem eingearbeiteten, lange vertrauten Team besteht nun darin, feste Gewohnheiten umzustellen. Die neue Aufgabe der eigenständigen Teamsteuerung ist hierbei in den meisten Fällen unproblematisch, da die Teams auch in klassischen Projekten die Feinplanung oftmals stark mitgestalten.

Typische Probleme dieser Teams liegen eher in der geforderten Transparenz und der starken Fixierung einzelner auf bestimmte Rollen, Technologien oder fachliche Gebiete.

Durch die Fixierung auf bestimmte Aufgaben entstehenden Kopfmonopole, die für das Unternehmen riskant und teuer sind. Zudem erschweren sie eine kontinuierliche Auslastung des gesamten Teams.

In Scrum Projekten wird ein Cross-funktionales Team angestrebt, bei dem möglichst alle Tätigkeiten von allen Teammitgliedern übernommen werden können. Auch wenn dies die Leistung einzelner Spezialisten zumindest übergangsweise bremst, werden hierdurch mit Blick auf das gesamte Team eine höhere Performanz und mehr Identifikation zum Produkt erwartet. [Ec14, Re12].

Die geforderte Transparenz, z. B. durch tägliche Stand-Up-Meetings, Backlog Grooming und Reviews erzeugt einen kontinuierlichen Leistungsanspruch an jedes Teammitglied, das sich damit der Teamkontrolle aussetzt.

Für eher ergebnisorientiert-arbeitende Mitarbeiter ist diese Transparenz und damit der einhergehende Druck typischerweise eher motivierend. Die Steuerung über User-Stories und Task sowie die tägliche Aktualisierung der Restaufwände entspricht gut ihrer Arbeitsweise. Schwieriger ist es für eher introvertierte Menschen, die sich der Teamkontrolle lieber entziehen.

In jedem Fall offenbart die Transparenz fehlende Motivation, Kompetenz und Aufgaben sowie eine geringe Leistung einzelner Mitarbeiter. Im Sinne einer optimalen Teamperformanz ist genau dies gewünscht und ermöglicht ein Gegensteuern. Für den einzelnen Mitarbeiter kann die Teamkontrolle allerdings bedrohlich sein. Hier liegt es in der Verantwortung des Scrum Masters, sorgsam die Ursachen zu hinterfragen, zu fördern oder ggf. auch das Team zu verändern.

2.2.3 Auftraggeber / Kunde

Die Rolle des Kunden scheint im SCRUM-Modell relativ eindeutig in Bezug auf Aufgaben und Kompetenzen. In Kooperation mit dem Product Owner werden fachliche Anforderungen in Form von User-Stories beschrieben und priorisiert. Kontinuierliche Abnah-

men im Sprint Review ermöglichen die Ergebniskontrolle und schärfen den Blick auf das erzielte Ergebnis.

Der Wechsel zu einer agilen Vorgehensweise in traditionellen Unternehmen im Konzernverbund ist dennoch stark herausfordernd. Die Frage, welche Organisation den Product Owner stellt, ist bereits politisch und unternehmensstrategisch brisant [HKZ11]. Darüber hinaus erscheint vielen Auftraggebern der Verzicht auf klassische Planungs- und Berichtsinstrumente als ein Verlust an Kontrolle. Der enorme Gewinn an Kontrolle, der durch häufige Produktdemonstrationen und -auslieferungen entsteht, wird initial nicht wahrgenommen.

3 Einsatz agiler Vorgehensweisen in der Praxis

In diesem Abschnitt werden die Erfahrungen der Pilotierung von AGSM agile zusammengefasst. Hierbei werden zunächst das Vorgehen erläutert und dann spezifische Erkenntnisse dargelegt.

3.1 Entwicklung und Pilotierung von AGSM agile

Der agile Entwicklungsprozess AGSM agile wurde als Synthese von RUP und SCRUM mit Blick auf CMMI-Konformität entwickelt. Vier Teams wurden daraufhin als Piloten gestartet. Keines der vier Teams wurde neu gebildet, alle betreuten bereits lange jeweils einen bestimmten Systembereich.

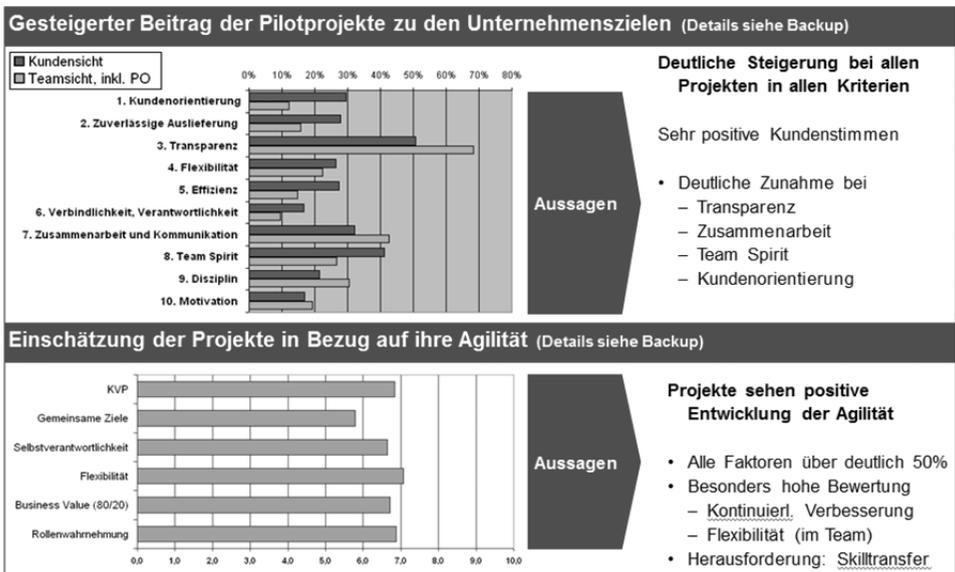


Abbildung 3: Ergebnisse der Pilotierung

Nach einem Jahr Projektlaufzeit wurde das Ergebnis retrospektiv betrachtet. Hierbei wurden drei Aspekte beleuchtet:

1. Umfrage bei Teammitglieder und Auftraggebern, welche Veränderungen sie dem jeweiligen Projekt in Bezug auf zehn unternehmerische Dimensionen bescheinigen
2. Selbsteinschätzung des Teams zum Grad der Agilität auf Basis einer Scorecard mit sechs Parametern
3. **Interview** mit Team und Kunden

Folgende Ergebnisse können zusammengefasst werden:

1. In allen Dimensionen wurden Verbesserungen gesehen, in keiner eine Verschlechterung. Die deutlichste Zunahme war bei Transparenz, Zusammenarbeit, Team Spirit und Kundenorientierung zu erkennen. In den Bereichen, in denen das Team selbst wenig Verbesserung sah, gab der Kunde ein deutlich positiveres Feedback (Verbindlichkeit, Effizienz, Zuverlässigkeit, Kundenorientierung).
2. Die Teams schätzten den Grad ihrer Agilität eher verhalten ein. Der Wert liegt zwar deutlich über 50%, aber kein Team behauptet, mehr als 2/3 des Weges hinter sich zu haben. Es gibt in den einzelnen Werten kaum eine Differenz. Interessant ist, dass der Wert zu KVP nicht relativ groß ist, wo doch die Änderungen für die Teams insgesamt zahlreich und tiefgreifend gewesen sein sollten.

Durch die Interviews konnten weitere Erkenntnisse gewonnen werden, die in Abschnitt 3.2 näher beleuchtet werden.

3.2 Ergebnisse der Pilotierung in Bezug auf das Team

Projektstart: Die erste Hürde ergibt sich durch die Art, wie Projekte ins Leben gerufen werden. Nach Freigabe durch das Portfoliomanagement wird als erstes ein Projektleiter bestellt, der dann sein Kernteam staffed. Dann erst legt er zusammen mit diesem die Vorgehensweise des Projektes fest. Man hat also schon viele Trägheitsmomente in Richtung klassischer Aufstellung geschaffen, bevor die Entscheidung getroffen wird, agil oder klassisch vorzugehen. Insbesondere ruft man den Projektleiter auf, sich selbst abzuschaffen.

Projektleiter, Product Owner, Scrum Master: Für den Projektleiter ist die Rolle des Scrum Masters wie oben beschrieben unattraktiv, weil er die Außensicht verliert. In unseren Piloten wurde er somit der Product Owner. Der Scrum Master wurde in den meisten Fällen aus dem Entwickler-Team gewonnen. Diese Menschen äußerten sich in den Pilotinterviews erstaunt über den Einfluss, den sie aus dieser Rolle heraus haben. Diese späte Selbsterkenntnis verdeutlicht zweierlei. Zum einen wachsen die so ausgewählten Scrum Master nur allmählich und zurückhaltend in ihre Rolle hinein. Zum anderen wurden unsere Scrum Master insbesondere vom Management nicht als Change-

Agents prominent genug gefordert. Sie gehen Konflikten eher aus dem Weg als dass sie das Team und dessen Umfeld zur Veränderung treiben.

Dazu passt auch, dass die Veränderungsgeschwindigkeit vom Team als moderat empfunden wurde. Weiterhin haben sowohl der Scrum Master als auch das Team deutliche Schwierigkeiten, aus dem Schatten der Vergangenheit heraus zu treten. Dies gilt insbesondere dann, wenn der ehemalige Projektleiter nun als Product Owner mit einer ähnlichen Macht auftritt. Die Störung in der Rollenwahrnehmung wirkt sich damit auf die elementare Voraussetzung für agiles Arbeiten aus: eine eigenmotivierte, transparente und vertrauensvolle Zusammenarbeit mit einem effektiven Feedbackzyklus.

Spezialist, Team Player: Ein weiterer von den Piloten angesprochener Themenkomplex bezieht sich auf die Auflösung der Spezialisierung. Fast alle Teams sind begeistert und nehmen die Agilität als Gewinn und Motivation wahr. Das bedeutet leider nicht, dass dies für jeden einzelnen Mitarbeiter im Team zutrifft. Es gibt in etwa einen von zehn Mitarbeitern, der das agile Vorgehen auch dann noch ablehnt, wenn die Umstellung stattgefunden hat.

Der Unterstellung, dies seien nur Mitarbeiter, die sich vor der Transparenz scheuen, können wir mit unseren Erfahrungen entgegentreten: es handelt sich um sehr gute Mitarbeiter. Es sind auch die Protagonisten aus dem Zeitalter der Standardisierung, die Spezialisten. Sie haben sich einen Ruf erworben, gelten als effektiv und effizient, sind bekannt bis zum Kunden. Sie sind die Kopfmönopole im Team. Und sie möchten es auch bleiben.

4 Zusammenfassung und Ausblick

Aus den zurückliegenden Projekten lassen sich Erfolgsfaktoren ableiten. An wenigen Stellen ist dabei die konkrete Aufgabenteilung im Rollengefüge der kritische Faktor, meistens entscheiden eher die Details auf der sozialen Ebene über den Erfolg eines agilen Teams.

Folgende Lessons Learned sind aus unserer Sicht für viele Organisationen relevant:

1. Der Scrum Master muss für Führungs- und Veränderungsprozesse qualifiziert sein. Gerade zu Beginn sollte die Veränderungsgeschwindigkeit hoch sein, um ein Team schnell effektiv aufzustellen. In dem meisten Fällen ist ein Scrum Master aus den Reihen des Entwicklungsteams nicht geeignet.
2. Die Organisation muss die Rolle des Scrum Masters entsprechend hochwertig aufbauen, damit diese für hochqualifizierte Mitarbeiter attraktiv ist.
3. Der Product Owner muss das Produkt fachlich und technisch verstehen. Ein grober Überblick und ein Jourfixe pro Woche reichen an dieser Stelle nicht aus. Wenn die freie Kapazität bei einer Linienführungskraft nicht ausreicht, sollte sie die Rolle nicht beanspruchen.

4. Erster entscheidender Treiber für die Teamperformance ist die Transparenz und die damit einhergehende soziale Kontrolle. Der Scrum Master sollte bei Skeptikern und Verweigerern analysieren, welche Ängste vorliegen und ob diese begründet sind.
5. Zweiter entscheidender Treiber für die Teamperformance ist die Cross-Funktionalität im Team. Die dafür benötigte Verbreiterung von Wissen kommt nicht von selbst sondern muss aktiv gefordert und gesteuert werden.
6. Der Kunde muss den Vorteil des agilen Arbeitens möglichst schnell selbst erleben. Hier sollten möglichst schnell wichtige User-Stories umgesetzt, demonstriert und ausgeliefert werden. Hierdurch wird Vertrauen aufgebaut, das wiederum erleichtert es dem Kunden, auf klassische Steuerungsinstrumente zu verzichten.

Literatur

- [Cl11] Clemens, M.: Der Scrum Master ist nicht dein Freund! <https://blog.codecentric.de/2011/10/der-scrummaster-ist-nicht-dein-freund/>.
- [DS10] Davies, R.; Sedley, L.: Agile Coaching: Praxis-Handbuch für Scrum Master, Teamleiter und Projektmanager in der agilen Software-Entwicklung, mitp.
- [Ec14] Eckstein, J.: Verzögerungskosten aufgrund von Experten, <http://www.heise.de/developer/artikel/Verzoegerungskosten-aufgrund-von-Experten-2153129.html>.
- [Fi00] Fisher, K.: Leading Self-Directed Work Teams: A Guide to Developing New Team Leadership Skills, McGraw-Hill.
- [Gl11] Gloger, B.: Der Scrum Master – Wirklich ein Weichei? <http://borisgloger.com/2011/02/17/der-scrummaster-wirklich-ein-weichei/>.
- [HKZ11] Hacker, T.; Kraft, B.; Zöll, A.: Projektzuschnitt für die inkrementelle Systementwicklung im Konzernverbund, Zusammenspiel von Vorgehensmodellen und Organisationsformen, Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik, 18.
- [KK12] Kirchhof, M.; Kraft, B.: Hybrides Vorgehensmodell - Agile und klassische Methoden im Projekt passend kombinieren, Projekt Magazin 11/2012.
- [Se] SEI: Capability Maturity Model Integration (CMMI) <http://www.sei.cmu.edu/cmmi/>.
- [IB] IBM: IBM Rational Unified Process, <http://www-01.ibm.com/software/rational/rup/>.
- [Sc] Scrum.org: The Scrum Guide, <https://www.scrum.org/Scrum-Guide>.
- [Pi11] Pink, D.H.: Drive, The Surprising Truth About What Motivates Us, Canongate Books Ltd.
- [Re12] Reinertsen, D.: The Principles of Product Development Flow: Second Generation Lean Product Development, Celeritas Publishing.

Standardisierung vs. Kultur: Klassisches und agiles Projektmanagement im Vergleich

Stefan Hilmer, Alexander Krieg

Acando GmbH
Millerntorplatz 1,
20359 Hamburg
stefan.hilmer@acando.de
alexander.krieg@acando.de

Zusammenfassung: Standardisierungen werden im klassischen Projektmanagement eingesetzt, um den Erfolg von Projekten sicherzustellen. Zu diesem Zweck werden oft unternehmensweit oder allgemeingültig standardisierte Vorgehensmodelle eingesetzt. In der agilen Welt wird generell auf weitreichende Standards verzichtet. Dafür wird vermehrt auf die Projektkultur gesetzt. Sie schließt nicht standardisierte Lücken zum Beispiel durch die Selbstorganisation der Projektteams. Dieser Aufsatz zeigt die Vorzüge beider Seiten und macht dabei deutlich, dass Standardisierung und Kultur einander nicht ausschließen. Vielmehr werden vermeintliche Widersprüche in mögliche Synergien überführt.

1 Einleitung

Im Projektmanagement stehen sich die klassische und die agile Welt oft scheinbar konträr gegenüber. Vielfach wird von einem Paradigmenwechsel gesprochen, was die Distanz zwischen beiden Lagern erklären soll. Oft sind es aber auch nur Vorurteile, die zu einer Trennung der Protagonisten führen. Dabei bieten beide Seiten Vorteile, die situationsabhängig, richtig eingesetzt, Projekten große Vorteile bringen können.

Ein typisches Merkmal klassischer Ansätze sind weitreichend standardisierte Vorgehensweisen. Sehr konkret und detailliert beschreiben solche Standards Phasen, Meilensteine, Rollen, Aufgaben, Ergebnisse und anderes mehr. Die Vorzüge dieses Ansatzes galten lange Jahre als unumstritten. Immer mehr allgemeingültige oder auch situationsspezifische Standards wurden definiert. Sie und ihre Vorzüge werden in diesem Beitrag in Kapitel 2 näher betrachtet.

Einen zentralen Aspekt agiler Vorgehensweisen bildet die Projektkultur. Sie basiert im Wesentlichen auf selbstorganisierten Teams und der vollständigen Integration des Kunden in das Projekt. Das Team, einschließlich Kunde, rückt immer mehr in den Mit-

telpunkt. Auch aus diesem Ansatz ergeben sich mächtige, inzwischen unbestrittene Vorteile, die zu einer stetigen Verbreitung der agilen Methoden führen. Die Gründe dafür sollen in Kapitel 3 dieses Beitrags eingegangen werden.

Den Kern dieses Beitrags bildet aber die Gegenüberstellung der in den Kapiteln 2 und 3 thesenartig zusammengestellten Vorzüge beider Ansätze. Auf dieser Basis wird in Kapitel 4 eine Synthese zwischen Standardisierung und Kultur, also zwischen klassischem und agilem Projektmanagement entwickelt.

Die Darstellungen in diesem Beitrag beruhen auf Beratungs- und Projekterfahrungen der Autoren. 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.

2 Standardisierung – der klassische Ansatz

Standardisierung im Wortsinn bedeutet Vereinheitlichung. Im Projektmanagement werden in der Regel Vorgehensmodelle standardisiert. Das bedeutet, es werden Methoden und Artefakte des Projektmanagements zu standardisierten Projektabläufen zusammengefasst. Solche standardisierten Vorgehensmodelle werden auch als Projektmanagementsysteme bezeichnet (vergl. DIN 69904 und 69905).

Standards, die von offiziellen Gremien wie dem Deutschen Institut für Normung (DIN) festgelegt werden, haben generell einen großen Geltungsbereich. Ein Beispiel für ein auf dieser Ebene standardisiertes Projektmanagement Vorgehen ist das vom britischen Office of Government Commerce (OCG) aufgestellte Prince2 [Of09]. Oft entwickeln sich aber auch in bestimmten Branchen eigene standardisierte Vorgehensmodelle für das Projektmanagement. In der Softwareentwicklung zum Beispiel ist der Rational Unified Process [Ve00] sehr bekannt. Einen großen Raum nehmen jedoch auch unternehmensinterne Standards ein. Vielfach entwickeln Großunternehmen oder Konzerne eigene Standards und erlangen so organisationspezifisch gestaltete Modelle, die die organisationseigenen Belange besser berücksichtigen können. Eine Aufgabe, für die vielfach eine eigene Organisationseinheit wie beispielsweise ein „Project Management Office – PMO“ geschaffen wird [Hi13].

Egal auf welcher Ebene ein Standard entsteht, in einem offiziellen Gremium, in einer bestimmten Branche oder in einem Unternehmen, immer werden die gleichen Ziele angestrebt. Ziele, die typischerweise bestimmte Vorteile mit sich bringen:

1. Gemeinsames Verständnis aller Beteiligten
2. Verbesserung der internen und externen Zusammenarbeit
3. Austauschbarkeit von Projektbeteiligten
4. Vergleichbarkeit von Projekten und deren Arbeitsständen

5. Verbessertes Management von Portfolien und Programmen

Wird ein Vorgehensmodell, das alle anstehenden Aufgaben, angestrebten Ergebnisse und ausführenden Rollen klar definiert, zum Standard erhoben, sorgt es für ein ausgeprägtes **gemeinsames Verständnis aller Beteiligten**. Es herrscht ein transparentes Bild bezüglich aller gegenseitigen Erwartungshaltungen.

Eine **Verbesserung der internen und externen Zusammenarbeit** wird immer dann erreicht, wenn ein gemeinsames Vorgehen definiert ist. Normen gelten – auch und gerade im Projektmanagement – als notwendig, um insbesondere die Kommunikation in und zwischen Organisationen zu erleichtern [SOP08]. Dies erreichen aber auch branchen- bzw. unternehmenseigene Standards. Sie müssen jedoch gerade im Außenverhältnis explizit benannt bzw. transparent gemacht werden.

Vorgehensmodelle mit klaren Rollenverteilungen und entsprechenden Beschreibungen führen zu einer hohen **Austauschbarkeit von Projektbeteiligten**. Die hohe Transparenz, die ein Standard mit sich bringt, und die Verbindlichkeit, die durch ihn gewährleistet wird, führen zu einem klaren gegenseitigen Rollenverständnis. Fällt ein Projektbeteiligter aus, ist nicht das Individuum zu ersetzen, sondern lediglich die im Modell beschriebene Rolle neu zu besetzen.

Die **Vergleichbarkeit von Projekten und deren Arbeitsständen** bedeutet, dass verschiedene Projekte an verschiedenen Punkten im Sinne eines Benchmarking miteinander verglichen werden können. Dabei geht es darum, aus einer übergeordneten Management-Perspektive heraus mehrere Projekte vergleichen und damit steuern zu können. Die Erreichung dieses Ziels wird durch die Anwendung eines standardisierten Vorgehens im hohen Maße unterstützt. Ein entsprechendes Modell bietet einzelne vergleichsfähige Meilensteine und im Idealfall auch Messgrößen.

Nicht nur die Vergleichbarkeit von Projekten wird durch einheitliche Vorgehensweisen verbessert. Standards sorgen auch für ein **verbessertes Management von Portfolien und Programmen**. Standardisiert abgewickelte Projekte lassen sich sehr viel leichter in ihrer Ausrichtung im Sinne einer Programmsteuerung optimieren. Standardisierte klassische Vorgehen sind somit auch für sehr große Vorhaben geeignet. Für ein übergreifendes Projekt-Portfoliomanagement ist es zudem wichtig, dass alle Vorhaben des Portfolios in einem hohen Maße messbar und damit vergleichbar sind. Voraussetzungen, die durch ein standardisiertes Vorgehensmodell idealtypisch bereitgestellt werden.

3 Projektkultur – der agile Ansatz

In der agilen Welt wird vielfach auf weitreichende und vor allem detaillierte Standardisierung verzichtet. Daher rührt auch die ältere Bezeichnung der „leichtgewichtigen“ Vorgehensmodelle. Ersatz bietet hier die Projektkultur, die selbstorganisierten Teams und der Ansatz, sich ständig verbessern zu wollen.

Das bedeutet nicht, dass es keine Standardisierungen gibt. Sie beziehen sich jedoch stärker auf die Interaktion im Team. Scrum beispielsweise definiert unter anderem die Meeting-Kultur des Projektes. Grundsätzlich werden agile Prozesse und Vorgehen zwar als Rahmen beschrieben, leben aber von der konkreten Ausgestaltung durch das jeweilige Team.

Im Agilen Manifest [Be01] stehen die Grundwerte und Gedanken der agilen Arbeits-sicht.

- Individuen und Interaktion über Prozesse und Werkzeuge
- Lauffähige Software über ausführliche Dokumentationen
- Zusammenarbeit mit dem Kunden über strikte Vertragsentwürfe
- Umgang mit Veränderungen über festgelegte Pläne

Im agilen Vorgehen geht es in erster Linie um Werte, Transparenz, Kommunikation und Kooperation und nicht um Prozesse oder Werkzeuge. Das bedeutet nicht, dass der betriebswirtschaftliche Projekterfolg in der agilen Welt nicht genauso im Mittelpunkt steht. Nur rücken auch immaterielle Vermögenswerte mehr in den Fokus als klassische Controlling-Kennzahlen.

Weitere Kernpunkte sind schnelles und flexibles Reagieren auf Veränderungen im Projekt, was durch ein flexibles Framework, flache Hierarchien und ein hohes Maß an Kommunikation und Eigenverantwortung im Projektteam gewährleistet wird.

Ein enger und kooperativer Umgang mit dem Kunden durch alle Projektphasen hindurch ist gelebtes Ziel, um eine bestmögliche Kundenzufriedenheit zu erreichen. Das schafft ein hohes Maß an Vertrauen.

Völlig unabhängig, wie sich die Projektrahmenparameter, Stakeholder und Projektziele zusammensetzen und gestalten, folgendes wird immer Kernbestandteil der agilen Kultur sein:

- 1. Lösungsorientierte selbstorganisierte Teams**
- 2. Förderung von Eigeninitiative und Verantwortung**
- 3. Regeln entwickeln und kontrollieren sich selbst**
- 4. Gelebter Wissenstransfer im Projekt**
- 5. Messbarkeit der Fertigstellung**

Lösungsorientierte selbstorganisierte Teams besitzen die Möglichkeit, Führungsverantwortung und Rollen situativ so zu verteilen, wie es zu der gestellten Aufgabe und den einzelnen Personen passt.

Ein agiles Umfeld ist geprägt durch flache Hierarchien, also Organisationsstrukturen, in denen Ranghöhere wenige Eingriffe in Entscheidungen Rangniedrigerer vornehmen. Diese Organisationsstruktur setzt verstärkt **auf die Förderung von Eigeninitiative und Verantwortung**, die von jedem Teammitglied auf das eigene Handeln umgesetzt werden muss. Eigeninitiative und Verantwortung sind ausgerichtet auf die jeweilige Zielerreichung sowie die kontinuierliche Verbesserung der täglichen Prozesse. Im agilen Projektumfeld wird durch den permanenten Focus auf Right Practice eine stetige Verbesserung der Prozesse und Kommunikationsstrukturen im Projekt erreicht.

Regeln und Standards innerhalb eines Teams werden als fester kultureller Bestandteil gesehen aber nicht als starres Regelkorsett. Da sie sich im kontinuierlichen Verbesserungs- und Anpassungsprozess befinden, benötigt es keine übergeordneten Regulierungsebenen und Controlling. **Regeln entwickeln und kontrollieren sich selbst.**

Mit Wissenstransfer ist hier jegliche Form von aktivem Austausch von Wissen gemeint. Auf diesen wird in agilen Teams viel Wert gelegt. Die Meeting-Kultur in agilen Teams, wie auch der transparente und kooperative Umgang sorgt für einen **gelebten Wissenstransfer im Projekt.**

Im agilen Projektmanagement wird pro Projektabschnitt, wie beispielsweise dem Sprint in Scrum, ein getestetes und lauffähiges Produktinkrement ausgeliefert. Innerhalb eines Abschnitts werden die Phasen von Anforderung, Design, Entwicklung und Test durchlaufen. Am Ende eines jeden Abschnitts kann das exakte Verhältnis von erreichten Ergebnissen und von benötigten Ressourcen gemessen und dargestellt werden. Diese frühzeitige **Messbarkeit der Fertigstellung** sorgt neben einem objektiven Projektcontrolling für Kundenzufriedenheit und Vertrauen. Hinzu kommt die Durchführung von Reviews als fester Bestandteil am Ende jedes Projektabschnittes. Sie bringen eine erhöhte Planungssicherheit im laufenden Projekt mit sich. Der Grund dafür ist ein verbesserter Wissensstand und eine daraus resultierende dynamische Plananpassung.

4 Synthese

Um die in den vorangegangenen Kapiteln aufgestellten Thesen zusammenzuführen, werden im Folgenden die auf die Standardisierung bezogenen Aussagen aus Kapitel 2 herangezogen und denen der Projektkultur aus Kapitel 3 gegenübergestellt. Daher bilden die fünf Thesen der Standardisierung auch die Überschriften der folgenden Unterkapitel. Diese Struktur der Darstellung ist zufällig und bestenfalls mit der Chronologie der Entstehung beider Ansätze zu begründen. Sie soll auch lediglich die Struktur der Darstellung festlegen und nicht die inhaltliche Aussage beeinflussen.

Gemeinsames Verständnis aller Beteiligten

In standardisierten Vorgehensmodellen der klassischen Welt werden Rollenprofile detailliert definiert und konkreten Aufgaben, z.B. Arbeitspaketen, zugeordnet. Ein typisches, sehr bekanntes Beispiel hierfür ist der RUP, der Rational Unified Process [Ve00]. Aber auch in organisationseigenen Modellen ist diese Herangehensweise weit verbreitet.

Es gibt aber nur in sehr wenigen Fällen die geeignete Person für ein detailliert definiertes Rollenprofil. Zudem passt häufig die Rollendefinition nicht zu den realen im Projekt benötigten Fähigkeiten.

Im Agilen werden Rollen abstrakter beschrieben, man sucht ein Teammitglied mit einer gewissen fachlichen Prägung. Es werden nicht automatisch feste Aufgaben an eine Rolle gebunden. Beispielsweise werden in Scrum die Aufgaben für das Team definiert und nicht für einzelne Teammitglieder. Mit dem Begriff „Teammitglied“ wird eine der wenigen Rollen in Scrum [Pi07] beschrieben.

Es gibt in der klassischen und der agilen Welt Schlüsselrollen wie z.B. den Projektleiter, den Scrum Master oder den Product Owner, die es zu definieren und zu besetzen gilt. Darüber hinaus erweist es sich als besonders zielführend, Verantwortung und Rollen situativ zu verteilen. **Lösungsorientierte selbstorganisierte Teams** (vergl. Kapitel 3) sind eine gute Möglichkeit, diese Art der Arbeitsteilung und Verteilung zu erreichen. Der Fokus der Teams ist nicht die Einhaltung starrer Rollen, Vorgehen und Prozesse sondern eine offene Kommunikation und die Eigenverantwortung, notwendige Lösungsschritte situativ zu erkennen und umzusetzen. Aus diesem Grund benötigt man zur Einführung agiler Ansätze im Unternehmen auch die richtigen Rahmenparameter. Die Praxis zeigt, dass zwei agile Pilotprojekte geeignet erscheinen. Wichtig sind zudem ein oder zwei praxiserfahrene Teammitglieder zur Vermittlung der Werte und Vorgehen, wobei auch der Einsatz eines agilen Coachs zu empfehlen ist. Grundsätzlich gilt aber, dass die Einführung so individuell ist wie das Unternehmen selbst. Es gibt viele Methoden aber keine Blaupause.

Verbesserung der internen und externen Zusammenarbeit

Im Klassischen werden Beziehungen zwischen Teams und Rollen definiert und in Regeln schriftlich fixiert, inklusive Verantwortungen, Verpflichtungen und Befugnissen [SOP08]. Daraus entsteht eine feste gegenseitige Erwartungshaltung. Das schafft Klarheit und geordnete Verhältnisse. Jeder weiß, was er zu tun hat und was von ihm erwartet wird.

Oft stellt sich der Projektalltag aber nicht so klar strukturiert dar. Häufig sind die gelebten Kommunikationsgewohnheiten einzelner Personen oder ganzer Teams anders als evtl. zu Beginn definiert. Zudem können auch nicht alle Möglichkeiten und Eventualitäten im Voraus abgedeckt werden. Es werden immer Situationen eintreten, die nicht beschrieben wurden.

Im Agilen werden keine strengen Vorgaben getroffen, wie ein Team intern oder mit anderen Teams zu kommunizieren hat. Zur **Förderung von Eigeninitiative und Verantwortung** (vergl. Kapitel 3) gibt man die Organisation der Kommunikation an die Teams ab.

Mit Regeln und Standards wird in beiden Welten das Ziel verfolgt, Zusammenarbeit und Schnittstellen in und zwischen Teams und einzelnen Stakeholdern klar zu definieren. Das wird in der klassischen Welt vehementer und unflexibler verfolgt [SOP08] als in der

agilen. Auch in der agilen Welt gibt es eine große Anzahl fester Größen, Regeln und klar definierter Abläufe [OW07], ohne die das agile Vorgehen nicht funktionieren könnte. Nur sie werden individueller gehandhabt und schneller angepasst, je nach dem was aktuell am passendsten erscheint. Jedes Team und jedes Projekt ist individuell verschieden, **Regeln entwickeln und kontrollieren sich selbst** (vergl. Kapitel 3). Regelkorridore und anpassungsfähige Standardvorgaben sollen eine Hilfe bieten, aber nicht blockieren.

Austauschbarkeit von Projektbeteiligten

Standardisierte Vorgehensmodelle gewähren ein hohes Maß an Austauschbarkeit der Projektbeteiligten. Wenn eine Rolle hinreichend klar beschrieben ist, kann sie auch entsprechend besetzt werden. Muss im Falle eines Austausches eine Rolle neu besetzt werden, sucht man den geeigneten Kandidaten für das Rollenprofil und setzt ihn in das bestehende Team ein. Die Rollenprofile sollen dabei einen reibungslosen Ablauf gewährleisten [SOP08].

In der Regel hängt am Austausch einer Person mehr, als nur die Neubesetzung einer definierten Rolle. Personen bringen mehr mit, als nur die exakte Ausfüllung eines Rollenprofils. Erarbeitetes Wissen geht dem Team durch den Verlust einer Person verloren und sollte bestmöglich aufgefangen werden, ebenso sind soziale Aspekte der Teambildung zu beachten.

Der Teambuildungsprozess ist ein Kernpunkt in agilen Projekten und Teams, er wird intensiver gelebt als im klassischen Umfeld. In agilen Teams ersetzt man eine Person eher situationsabhängig. Es wird mehr nach den aktuellen Teambelangen geschaut, als nach dem exakten Ersetzen einer definierten Rolle. Das Team wählt das neue Teammitglied aus. Dabei kann es auch zu einer weitreichenden Neuverteilung der Aufgaben im Team kommen.

Eine Situation, zu der es immer wieder in laufenden agilen wie klassischen Projekten kommt, ist der Ausfall einer Person und eine damit verbundene Neubesetzung der Position. Die Suche, Auswahl und Einarbeitung einer geeigneten Person nimmt in jedem Fall Zeit in Anspruch und bringt in jedes agile wie klassische Team eine kurzzeitige Unruhe. In agilen Projekten arbeitet man aber im Gegensatz zur klassischen Welt mit abstrakten Rollenbeschreibungen [Pi07]. Das führt in der Regel zu einer besseren, situativ gezielteren Ergänzung des Teams, als eine Person nach einer strikten Rollenbeschreibung zu suchen. Das Team soll durch die neue Person im besten Fall noch verstärkt werden, im Agilen wird dabei mehr Wert auf Erfahrung, Komplementärwissen und soziale Aspekte Wert gelegt, als auf die rein fachliche Beschreibung. Durch die besondere Meeting-Kultur, wie z.B. den Daily Stand-up Meetings, und einen aktiv **gelebten Wissenstransfer im Projekt** (vergl. Kapitel 3) können neue Teammitglieder zudem schnell integriert werden. Dadurch werden der zeitliche Rahmen der Einarbeitung und das Ankommen im Team so kurz als möglich gestaltet.

Vergleichbarkeit von Projekten und deren Arbeitsständen

Die vierte These aus Kapitel 2 sagte aus, dass verschiedene Projekte in standardisierten Vorgehensmodellen verglichen werden können. Ein solches Modell bietet dazu Phasen und Meilensteine mit fest vorgegebenen, also standardisierten Inhalten. Mit geeigneten Messgrößen und Metriken können an den für alle Projekte gleichen Meilensteinen Kennzahlen ermittelt werden, die die Projekte vergleichbar machen [Ge09].

Allerdings werden dabei in der Regel nur Kenngrößen verglichen, die sich auf feste Punkte des Vorgehensmodells beziehen. Oft fehlt die Messbarkeit von Projektergebnissen und vor allem von deren Nutzen. Die Erfahrung zeigt, dass nicht selten Projekte formal gelingen, am Ende aber die Ergebnisse keinen oder nur geringen Nutzen bringen. Viele agile Methoden schaffen an dieser Stelle Abhilfe, indem den Ergebnissen z.B. lauffähiger Software ein höherer Stellenwert zugewiesen wird [Be01, Sc04]. In Kapitel 3 wurde dies durch die These **Messbarkeit der Fertigstellung** dargestellt. Scrum beispielsweise verlangt auslieferbare Software nach jeder Iteration [Pi07]. Über auslieferungsfähige Produkte bzw. über eine Bewertung dieser durch den Anwender kann eine Messbarkeit erreicht werden, die den Vergleich der o.g. Kennzahlen gegenüber dem Fortschritt des Produktes bzw. des tatsächlich erreichten Nutzen ermöglicht. Allerdings fällt es schwerer Aussagen über das noch zu leistende zu treffen, denn im Gegensatz zum klassischen Ansatz wird hier nicht Bezug auf einen festen Punkt in einem vorgegebenen Vorgehensmodell genommen.

Der Unterschied liegt also nicht darin, dass man mit einem der beiden Ansätze, agil oder klassisch, Projekte besser vergleichen kann, sondern darin, wie man die Projekte bewertet. In jedem Fall werden Messgrößen benötigt, die in vergleichbare Kennzahlen überführt werden. Termine und Kosten sind in der Regel leicht zu bewerten. Agile Methoden unterstützen, wie am Beispiel Scrum gezeigt, die Messung des Produktfortschritts im Projekt und die des tatsächlich erreichten Nutzens. Für die Bewertung der Leistung in Projekten, die einem standardisiertem Vorgehensmodell folgen, existieren viele verschiedene Ansätze [Lo08, SOP08]. Der Fertigstellungsgrad gilt hier als zentrale Kenngröße. Man muss also die Kenngrößen für Termine und Kosten auf die der Leistung beziehen, anstatt auf Meilensteine des Vorgehensmodells. Dieser Ansatz muss allen beteiligten Stakeholdern bis hin zum Kunden bewusst gemacht werden. Der Vorteil liegt darin, dass die Kenngrößen unabhängig vom Vorgehensmodell sind und die Projekte trotzdem vergleichbar bleiben. Auf diese Weise lassen sich klassisch durchgeführte Projekte mit agilen vergleichen, was im Falle einer hybriden Anwendung verschiedener Vorgehensmodelle in einer Organisation [Hi12] unabdingbar erscheint. Zusammenfassend kann also festgestellt werden, dass sich Projekte auch unabhängig vom Vorgehensmodell vergleichen lassen. Voraussetzung ist ein angemessenes Kennzahlen-System, das an die Situation und das Umfeld der zu vergleichenden Projekte angepasst sein sollte. Die Entwicklung eines solchen Modells bringt aber in jedem Fall einen hohen Aufwand mit sich.

Verbessertes Management von Portfolien und Programmen

Bei dieser fünften These sind zwei Aspekte zu unterscheiden, das Management von Projekt-Portfolien und das von Programmen. Projekt-Portfoliomanagement ist dabei gemeint als das Management mehrerer Projekte eines Portfolios, also beispielsweise einer Organisationseinheit oder eines Unternehmens. Programmmanagement hingegen befasst sich mit dem Management eines Programms, also mehrerer Projekte, die ein gemeinsames Ziel verfolgen [Lo08].

Für das Projekt-Portfoliomanagement spielt die Vergleichbarkeit von Projekten eine zentrale Rolle. Diese wurde im vorangegangenen Unterkapitel, „Vergleichbarkeit von Projekten und deren Arbeitsständen“, bereits betrachtet. Dort wurde gezeigt, dass bei geeigneter Wahl des Kennzahlensystems auch in einem durch die Projektkultur geprägtem Umfeld eine Mess- und damit eine Vergleichbarkeit von Projekten erzeugt werden kann.

Zum Programmmanagement wurde mit dieser These in Kapitel 2 ausgeführt, dass sich standardisiert abgewickelte Projekte sehr viel leichter auf die Ziele eines übergeordneten Programmes ausrichten lassen. Damit ergibt sich eine hohe Skalierbarkeit bis hin zur Anwendbarkeit in großen Programmen. Dem gegenüber stehen die Ziele des Agilen Manifest [Be01]. Aus ihnen wurde in Kapitel 3 eine **Messbarkeit der Fertigstellung** abgeleitet. Diese Messbarkeit der Fertigstellung, z.B. durch die Forderung nach auslieferbarer Software nach jeder Iteration [Pi07], verbessert ebenfalls die Zielerreichung. Diese Verbesserung kann sich das Programm zu Nutze machen. Dazu muss das jeweilige Programmmanagement die agilen Werte für sich annehmen und im Programm umsetzen. So kann der Programmmanager zusammen mit seinen Projektmanagern die bereits mehrfach erwähnten **Vorzüge Lösungsorientierter selbstorganisierter Teams** nutzen. Ein selbstorganisiertes Team arbeitet von sich aus lösungsorientiert und damit zielgerichtet.

Sollte es zum Abbruch eines Projekts kommen, bietet die regelmäßige Vorlage verwertbarer Ergebnisse, z.B. in Form lauffähiger Software, zusätzliche Optionen für das Management von Portfolien und Programmen. Ergebnisse können in anderen Projekten weiter genutzt werden.

Programm- und Portfoliomanagement nach agilen Werten erfordert allerdings erfahrungsgemäß eine sehr hohe Akzeptanz dieser Werte auf allen Ebenen bei den Projektteams, dem Projektmanagement, dem Programmmanagern und darüber hinaus bis ins Management. Eine agile Kultur gelingt nicht von allein. Sie muss von allen gewollt und gelebt werden.

5 Fazit

Der Einsatz von Standards und allgemeingültigen Vorgehensmodellen ist nach wie vor ein richtiger Ansatz zur Sicherung des Projekterfolges. Ein bis zu einem angemessenen Grad vereinheitlichtes Vorgehen, das auf Best Practice und Lessons Learned basiert, ist in jedem Fall anzustreben. Die Verwendung bereits erprobter, übertragbarer und anpass-

barer Modelle und Standards erleichtert zudem die Messbarkeit und Vergleichbarkeit des Erfolgsgrades von Projekten.

Teams, Kunden und Anforderungen in einzelnen Projekten werden immer komplexer und dynamischer. Daher wird ein bis ins Detail geregeltes Vorgehen nach einem einmalig definierten Standard nicht mehr zum Erfolg führen. Standards sollten daher nur noch als grobe Rahmenwerke mit angemessen geringem Detailierungsgrad vorgesehen werden. Nur so bleiben projektindividuelle Spielräume.

In fast allen oben angeführten Aspekten ist der Detailierungs- und der Konkretisierungsgrad von Standards und Regelungen ein elementarer Punkt. Es gehören viel Erfahrung, eine klare Sicht, Empathie und eine kooperative Führungshaltung dazu, in jeder einzelnen Situation klar zu erkennen, bis wohin ein Standard sinnvoll ist und ab wann situative Anpassungen oder neue Lösungsansätze angewandt werden sollten.

Es wird hierbei sicherlich nie ein „Fertig“ oder „Ideal“ geben. Die Balance zwischen detaillierter Vorgabe auf der einen Seite und Loslassen auf der anderen ist schwer zu finden. Führung als Dienstleistung zu verstehen und Entscheidungen an selbstorganisierte Teams abzugeben sind Fähigkeiten, die immer wieder aufs Neue erarbeitet werden müssen.

Ein modernes Projektmanagement sollte weder eine rein klassische noch eine rein agile Sicht einnehmen. Um aus beidem das Beste zu etwas Flexiblen, Modernen und Neuen zu verbinden, müssen beide Ansätze verstanden werden. Neben der neuartigen Weise mit und in Teams kooperativ zu arbeiten, zeigt uns die agile Welt auch eine neue Sicht auf Controlling und den Umgang mit Kennzahlen. Was nutzen Kennzahlen, die dem Projektmanager zeigen, sein Projekt sei erfolgreich und genau im Plan, wenn für den Kunden kein echter Nutzen und Mehrwert aus dem Projekt entsteht? Was können ein Programm- oder ein Portfoliomanagement mit solchen Kennzahlen anfangen?

Standardisierung und Kultur widersprechen sich nicht. Wenn man eine offene, transparente und flexible Kultur zum Standard erhebt, schafft man ein sich weitestgehend selbstregulierendes Projektumfeld.

Werden diese Ansätze der Zusammenarbeit und der neuen Sicht auf Kennzahlen und Controlling sowie auch verstärkt auf immaterielle Vermögenswerte durch weite Teile des Unternehmens gelebt, befindet man sich mitten in der Entwicklung zum Enterprise 2.0. Die dynamischen und komplexen Anforderungen und Veränderungen der Märkte lassen sich nicht aufhalten, aber durch eine offene und neue Sicht meistern.

Literatur

- [AB01] Abel Beck et al.: Manifesto for Agile Software Development. Siehe: <http://agilemanifesto.org/>, 2001.
- [Ge09] Gessler, M. (Hrsg.): Kompetenzbasiertes Projektmanagement (PM3): Handbuch für die Projektarbeit, Qualifizierung und Zertifizierung auf Basis der IPMA Competence Baseline Version 3.0. GPM - Deutsche Gesellschaft für Projektmanagement, 2009.
- [Hi12] Hilmer, S.: Hybride Vorgehensmodelle für ein unternehmensweit einheitliches, flexibles Projektmanagement. In (Linssen, O.; Kuhrmann, 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.
- [Hi13] Hilmer, S.: Verantwortung übernehmen: Das Project Management Office als Process Owner für das unternehmensweite Projektmanagement. In: (Hanser, E.; Mikusz, M.; Fazal-Baqaie, M. Hrsg.): Vorgehensmodelle 2013. GI-Edition, Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, S. 65-76, 2013.
- [Lo08] Lomnitz, G.: Multiprojektmanagement - Projekte erfolgreich planen, vernetzen und steuern. Moderne Industrie, 2008.
- [OW07] Oestereich, B.; Weiss, C.: APM - Agiles Projektmanagement: Erfolgreiches Timeboxing für IT-Projekte. dpunkt.verlag, 2007.
- [Of09] Office of Government Commerce: Erfolgreiche Projekte managen mit PRINCE2. The Stationery Office Ltd, 2009.
- [Pi07] Pichler, R.: Scrum - Agiles Projektmanagement erfolgreich einsetzen. dpunkt Verlag, 2007.
- [SOP08] Schelle, H.; Ottmann, R.; Pfeiffer, A.: ProjektManager. GPM - Deutsche Gesellschaft für Projektmanagement, 2008.
- [Sc04] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, 2004.
- [Ve00] Versteegen, G.: Projektmanagement mit dem Rational Unified Process. Springer Verlag, 2000.

Agile Entwicklung mit On- und Offshore-Partnern – Methodenverbesserung in der Praxis

Masud Fazal-Baqaie¹, Stefan Sauer¹, Torsten Heuft²

¹s-lab – Software Quality Lab
Zukunftsmeile 1
33102 Paderborn
{mfazal-baqaie,ssauer}@s-lab.uni-paderborn.de

²GEFA Gesellschaft für Absatzfinanzierung mbH
Robert-Daum-Platz
42117 Wuppertal
torsten.heuft@gefa.de

Abstract: Agile Softwareentwicklungsmethoden erfreuen sich einer steigenden Verbreitung, versprechen sie doch risiko-getrieben und leichtgewichtig auch bei sich verändernden Anforderungen gute Ergebnisse zu erzielen. Viele Unternehmen stehen jedoch vor der Herausforderung, agile Prinzipien mit den Gegebenheiten im Unternehmen zu vereinbaren, z.B. mit den internen Prozessen und organisatorischen Gegebenheiten. Ein besonders herausforderndes Beispiel ist die Softwareentwicklung in Kooperation mit Offshore-Partnern. Die Autoren berichten aus einem laufenden Methodenverbesserungsprojekt für ein Unternehmen der Finanzbranche, das für die Softwareentwicklung mit On- und Offshore-Partnern kooperiert. Anders als bisher soll der indische Offshore-Partner in naher Zukunft große Softwaresysteme federführend implementieren. Dabei sollen projektindividuell auch Onshore-Partner eingebunden werden. Der Beitrag stellt den systematischen Ansatz für die inkrementelle Methodenverbesserung vor, mit dem Varianten einer möglichst agilen Methode definiert werden sollen, die sowohl im Projektumfeld als auch im operativen Tagesgeschäft effektiv und effizient eingesetzt werden können. Die Autoren fassen außerdem die im Rahmen von Interviews erhobenen, bisherigen Erfahrungen in der Zusammenarbeit mit dem Offshore-Partner zusammen, die für die Methode berücksichtigt werden müssen.

1 Einleitung

Agile Methoden sind populär, weil sie es erlauben, im Vergleich zu plangetriebenen Methoden, wie z.B. RUP [Kr99] oder V-Modell XT [HH08], risikogetrieben mit wenig zusätzlichem Dokumentationsaufwand schrittweise die Lösung zu entwickeln und dabei Feedback von Kunden frühzeitig zu berücksichtigen. Während mit kurzen Iterationen fester Länge (z.B. Sprints) strenge Vorgaben an den Entwicklungsrahmen und die erwartete Funktionalität gestellt werden, wird dem Projektteam inhaltlich relativ große Frei-

heit bei der Entwicklung eingeräumt. Die Organisation und das inhaltliche Vorgehen obliegt dem Entwicklungsteam, das gemäß dem Motto „easy to learn, difficult to master“ deshalb über entsprechende Expertise und Erfahrung verfügen muss.

Unternehmen können existierende agile Methoden, wie z.B. Scrum [SS11] jedoch aus verschiedenen Gründen nicht einfach übernehmen. Zunächst, weil die *geforderten Gegebenheiten im Unternehmen nicht zutreffen* und nicht einfach hergestellt werden können. Bei dem Unternehmen dieses Praxisbeispiels ist z.B. nicht immer gegeben, dass die Projektmitarbeiter exklusiv für das Entwicklungsprojekt zur Verfügung stehen oder dass sich das Team an einem Standort befindet. Die Einbindung von Offshore-Partnern stellt eine besonders große Herausforderung dar, weil hier die Distanz besonders groß ist: zum einen räumlich, durch die geographischen Distanzen, zusätzlich zeitlich, durch die Zeitverschiebung und außerdem soziokulturell, durch die interkulturellen Unterschiede [Ec09]. Daneben müssen gesetzliche, regulatorische und organisationspezifische *Dokumentationspflichten und Prozessaufgaben berücksichtigt* werden. Außerdem sollten wo sinnvoll möglich, um Brüche zu vermeiden, *bestehende methodische Inhalte integriert* werden, beispielsweise etablierte Werkzeuge.

Wie in diesem Praxisbeispiel müssen Softwareentwicklungsmethoden, insbesondere agile Methoden im Offshoring-Kontext, also an den Unternehmenskontext angepasst werden, bevor sie sinnvoll eingesetzt werden können. In diesem Papier berichten die Autoren aus einem laufenden Praxisbeispiel für die Verbesserung einer Methode für einen Finanzdienstleister, die auf diesen zugeschnitten und für die Softwareentwicklung im Offshoring-Kontext geeignet ist. Die Methode muss (u.a.) für deutsch-deutsche und deutsch-indische Projekte funktionieren und deshalb auch soziokulturelle Aspekte berücksichtigen. Außerdem sollen verschiedenen Methodenvarianten angeboten werden, um die unterschiedlichen Gegebenheiten von Projekten zu berücksichtigen, z.B. durch eine Methodenvariante für kleine Onshore-Projekte und eine Variante für große Projekte mit verteilten Teams. Die Methode soll systematisch entwickelt werden, um zu vermeiden, dass Konflikte mit bestehenden Prozessen und Regeln entstehen oder im Unternehmen fehlende Kenntnisse vorausgesetzt werden. Der iterative Ansatz ist an die in [Fa+13] vorgestellte systematische Verbesserung des Requirements Engineering angelehnt, allerdings nicht auf diese eine Disziplin beschränkt.

2 Problemkontext

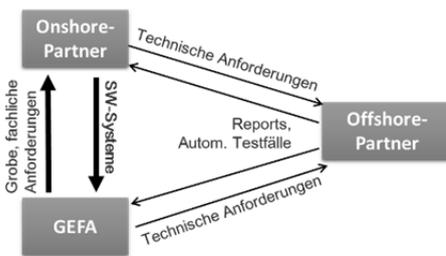
In diesem Abschnitt wird genauer umrissen, wie das Unternehmen der Finanzbranche, die GEFA Gesellschaft für Absatzfinanzierung mbH, zusammen mit ihren Partnern ihre Softwaredienste entwickelt. Es wird dabei zwischen der bisherigen Zusammenarbeit und Aufgabenverteilung und den hieran mittel- und langfristig geplanten Veränderungen unterscheiden.

2.1 Aktuelle Zusammenarbeit mit On- und Offshore-Partnern

Die GEFA Gesellschaft für Absatzfinanzierung mbH (im nachfolgenden GEFA) gehört zur international agierenden Bankengruppe Société Générale, die im Geschäftsbereich Société Générale Equipment Finance in 25 Ländern weltweit tätig ist. GEFA bietet ihren Kunden im Bereich der Absatz- und Investitionsfinanzierung individuelle Finanzierungs- und Leasinglösungen für mobile Wirtschaftsgüter an. Weiterhin gehört zur GEFA-Gruppe auch die GEFA BANK, die Privatkunden Tages- und Festgeldlösungen sowie Sparkonten anbietet. Ihren Hauptsitz hat die GEFA in Wuppertal.

Die GEFA-IT-Abteilung ist sowohl als interner Dienstleister für die verschiedenen Fachabteilungen des Unternehmens, als auch als externer Dienstleister im Rahmen der Zusammenarbeit mit anderen Entitäten innerhalb des Konzerns tätig. Zum Leistungsspektrum zählen dabei u.a. die Entwicklung, Wartung und der Betrieb von Core-Business-Applikationen, sowie das Erbringen diverser IT-Services für die internen und externen Kunden.

Aktuelle Zusammenarbeit



Mittel-/ langfristige Zusammenarbeit:

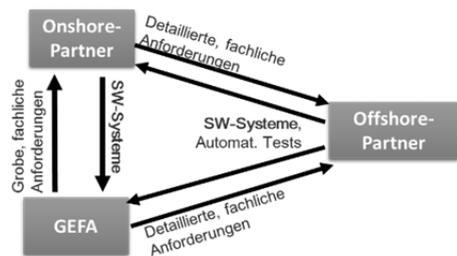


Abbildung 1: Aktuelle und geplante Zusammenarbeit der GEFA mit ihren Partnern

Die GEFA-IT kooperiert in diesem Zusammenhang mit verschiedenen Partnern und Dienstleistern (vergleiche Abbildung 1, links). Der wichtigste Onshore-Partner ist ein IT-Dienstleister aus Paderborn, der in langjähriger, vertrauensvoller Partnerschaft verbunden ist. Die Mitarbeiter des IT-Dienstleisters sitzen teilweise mit im Haus in Wuppertal und verfügen durch die Spezialisierung des Unternehmens und die langjährige Zusammenarbeit mit der GEFA über großes Domänen-Know-how und kommen deshalb insbesondere mit informalen, groben Anforderungen unter mündlicher Absprache zurecht. GEFA arbeitet seit einiger Zeit außerdem im Bereich des Offshoring mit einer konzerninternen Entität zusammen. Diese ist in Bangalore (Indien) ansässig und stellt den Konzerneinheiten Know-how und Entwicklungskapazitäten zur Verfügung. GEFA arbeitet im Rahmen verschiedener Projekte eng mit einzelnen Entwicklern und Teams des Offshore-Partners zusammen. Die Entwickler werden dabei von der GEFA-IT aus koordiniert und bekommen – auch von dem Onshore-Partner – abgegrenzte, technisch durchspezifizierte Aufgaben, z.B. im Bereich Reporting und Testautomatisierung.

2.2 Mittel- und langfristig geplante Zusammenarbeit

In naher Zukunft sollen Entwickler des indischen Offshore-Partners mit dem Aufbau zusätzlicher Kapazitäten neben den bisherigen Aufgaben auch größere Softwareprojekte federführend implementieren und die entstehenden Systeme dann später auch warten. Die Steuerung der Offshore-Mitarbeiter obliegt der GEFA, anders als bei dem Onshore-Dienstleister muss sie jetzt also einem Partner die Ausgestaltung der Entwicklungstätigkeiten vorgeben und zwar so, dass die Zusammenarbeit funktioniert, wenn die GEFA direkt mit dem indischen Partner zusammenarbeitet, aber auch wenn zusätzlich Onshore-Partner eingebunden sind (vergleiche Abbildung 1, rechts).

2.3 Angestoßenes Methodenverbesserungsprojekt

Aktuell führt die GEFA-IT-Abteilung alle Business- und IT-Projekte nach einer selbst entwickelten und in Ihrer Anwendung prämierten¹ Projektmanagementmethodik PROFIT durch, die Elemente agiler Softwareentwicklung (Scrum) mit Elementen des klassischen Projektmanagements (PRINCE2) kombiniert. Um die Weichen für die intensivere Zusammenarbeit mit dem Offshore-Partner zu stellen, wurde Anfang 2014 das Methodenverbesserungsprojekt angestoßen, aus dem in diesem Papier berichtet wird. Im Rahmen dieses Projektes sollen die methodischen Voraussetzungen geschaffen werden, um zukünftig Projekte mit On- und Offshore-Partnern mit einer variablen Methode effizient und effektiv abzuwickeln. Die Methode wird zunächst bis Ende 2014 iterativ weiterentwickelt und ab dem dritten Quartal 2014 laufend in begleitenden Pilotprojekten erprobt. Aktuell sind zwei zeitlich versetzt startende Pilotprojekte geplant. Zunächst gestartet ist ein Projekt für die Aktualisierung und Reimplementierung des Intranet-Auftritts der GEFA. Dieses wird von einem verteilten Team mit etwa zehn Personen in Kooperation zwischen GEFA-IT und dem Offshore-Partner unter Einbeziehung weniger Fachbereiche bearbeitet. In einem weiteren, in Kürze startenden Projekt ist die Neu-Entwicklung des Internetauftritts mit entsprechenden Online-Kundenservice-Funktionen geplant. In diesem Projekt wird es zwei größere Teams geben und neben dem Offshore-Partner werden auch Onshore-Partner eingebunden sein. Außerdem sind mehrere Fachabteilungen der GEFA involviert.

Wenn die entwickelte Methode sich bewährt, wird sie in die bestehende, GEFA-IT-weite Projektmanagementmethodik PROFIT integriert und anschließend großflächig ausgerollt.

3 Ziel und Ansatz der Methodenverbesserung

In diesem Abschnitt wird auf die Ziele und Herausforderungen für die Methodenentwicklung in diesem Methodenverbesserungsprojekt eingegangen und anschließend der

¹ itSMF-Projekt-Award für „Agiles ITSM Prozess-Redesign“, siehe <https://www.itsmf.de/services/projekt-award.html>

verfolgte Ansatz der Methodenverbesserung erklärt, mit dem iterativ Verbesserungen erarbeitet werden sollen.

3.1 Ziel der systematischen Methodenverbesserung

Das Ziel des Projektes ist es, den zukünftigen Rahmen für die systematische, agile Entwicklung und Pflege von Softwaresystemen in Kooperation mit den verschiedenen Partnern zu schaffen. Um eine möglichst effektive und effiziente Projektabwicklung zu ermöglichen, soll die Methode einem modularen Konzept folgen, bei dem verbindliche oder optionale Methodenbausteine anhand der jeweiligen Projektbedingungen gewählt werden. Aus diesen Bausteinen wird dann eine Methodenvariante für ein Projekt zusammengestellt. Die Methodenbasis mit den Methodenbausteinen soll langfristig gepflegt werden, um den Wissensaufbau und einen kontinuierlichen Verbesserungsprozess zu unterstützen. Abbildung 2 verdeutlicht an einem einfachen Beispiel, wie die Charakterisierung unterschiedlicher Projekte über die Wahl entsprechender Bausteine zu unterschiedlichen Methoden führt. Die Idee ist angelehnt an existierende Arbeiten aus dem Forschungsgebiet Situational Method Engineering [FLE13].

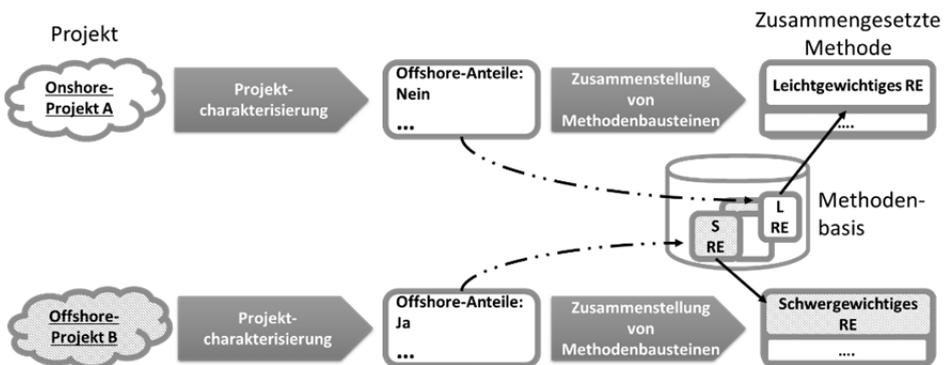


Abbildung 2: Die Wahl von Methodenbausteinen führt zu verschiedenen Methoden

3.2 Ansatz der systematischen Methodenverbesserung

Das Vorgehen ist in Abbildung 3 verdeutlicht. Zunächst wurden die Ziele („warum“) für die Methodenentwicklung genauer beleuchtet und eine Ist-Analyse durchgeführt. Hier hat das Projektteam über ein Materialstudium von Artefakten, z.B. erstellte Anforderungen unterschiedlicher Granularität (Fachkonzepte, einzelne Requirements) und über Experteninterviews verschiedene Stärken und Verbesserungspotentiale sowie den Unternehmenskontext erarbeitet. Beispielsweise wurde als Verbesserungspotentiale identifiziert, dass es in der Vergangenheit zu Missverständnissen mit dem Offshore-Partner kam (P1). Das betraf beispielsweise die unterschiedliche Erwartungshaltung, wann die Entwicklerdokumentation für Reports zu erstellen sei. Neben der Materialstudie und den Interviews wurden strategische Projektziele mit dem Management priorisiert. Strategische Projektziele sind diejenigen Ziele, die über die funktionale Erfüllung des Projektes

hinausgehen und deshalb eher mittel- oder langfristig für das Unternehmen eine Rolle spielen, aber dennoch in den Projekten berücksichtigt werden sollen. Damit sie deswegen nicht aus dem Blick des Projektteams geraten, sollen diese Ziele in der Methode verankert werden. Beispielsweise wurde festgelegt, dass die Verteilung von Projekterfahrungen über „Lessons Learned“ sehr wichtig ist und Projekterfahrungen stärker auch projektübergreifend geteilt werden sollen (P2).

Aus den Ergebnissen der beiden ersten Aktivitäten, wurden Methodenanforderungen abgeleitet, die beschreiben „was“ in der Methode umgesetzt sein soll. Dabei sollen die Methodenanforderungen die Entwurfsziele der Methode konkreter und transparenter machen, denn sie können mit den Stakeholdern validiert und priorisiert werden und stellen so einen wesentlichen Entwicklungsschritt in Richtung Methode dar. Beispielsweise lautet eine Methodenanforderung zu P1, dass die Methode sicherstellen soll, dass das Projektteam die gegenseitige Erwartungshaltung früh im Projekt klärt (M1). Eine Methodenanforderung zu P2 lautet, dass Lessons Learned im Projekt diskutiert werden und auf standardisierte Art an zentraler Stelle gesammelt werden müssen (M2).

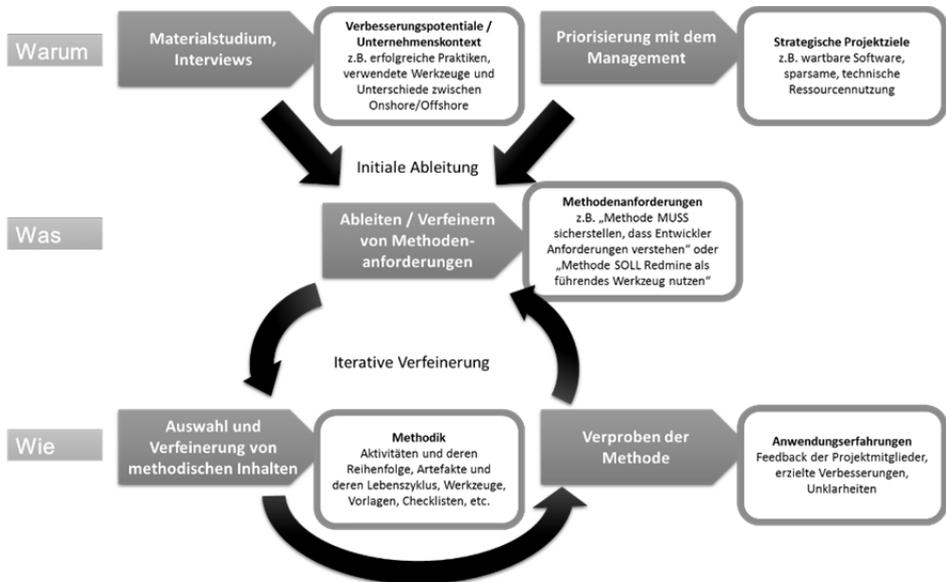


Abbildung 3: Iteratives Vorgehen innerhalb des Methodenverbesserungsprojektes

Auf Grundlage der Methodenanforderungen werden im nächsten Schritt mögliche Lösungsbestandteile („wie“) zusammengetragen und bewertet. Dazu zählen zum einen konkrete Maßnahmen, die sich in einzelnen Projekten innerhalb der GEFA bewährt haben (Good Practices, siehe z.B. Abschnitt 4.2.2). Zum anderen aber auch Maßnahmen, die in der Literatur beschrieben sind (Best Practices), beispielsweise zu agilem Vorgehen allgemein [Co02], [La12], [LA12] oder zu agilem Vorgehen in Offshore-Projekten [WSG10], [Su+10]. Beispielsweise könnte wegen M1 festgelegt werden, dass in dem Projektstartworkshop die gegenseitige Erwartungshaltung bezüglich der Dokumentationszeitpunkte zentraler Artefakte zu diskutieren und festzuhalten ist (L1). Es könnte

außerdem festgelegt werden, dass wegen M2 die in Retrospektiven (siehe [SS11]) gewonnenen Lessons Learned in einem bestimmten Dokument oder Verzeichnis zu dokumentieren sind (L2). Hier können dann auch verschiedene Varianten, also verschiedene Methodenbausteine, definiert werden. Generell werden die methodischen Inhalte in die bestehende Projektmanagementmethode PROFIT integriert, wobei prinzipiell auch Veränderungen an dieser denkbar sind.

Parallel zu der Definition von Methodenbausteinen soll die Methode in den beiden Pilotprojekten erprobt werden, um mit den direkten Rückmeldungen aus den Projekten die methodischen Inhalte zu validieren und die Methode iterativ zu verbessern. Damit die inhaltliche Gestaltung der Methode langfristig nachvollziehbar bleibt, sollen mit den Rückmeldungen aus den Projekten zunächst die Methodenanforderungen verfeinert werden, um dann wieder nach geeigneten Maßnahmen zu suchen. Beispielsweise könnte es sein, dass es trotz dem Austausch in L1 zu vermeidbaren Missverständnissen innerhalb des Projektteams kommt. Dann könnten in einer weiteren Iteration der Methodenverbesserung z.B. eine verfeinerte Definition der „Definition of Done“ (siehe [SS11]) oder verpflichtende Code-Reviews vorgeschlagen werden.

Neben der Erprobung in Pilotprojekten werden die erarbeiteten Inhalte regelmäßig der gesamten Leitung des IT-Managements vorgestellt, darüber hinaus wird bedarfsgetrieben punktuell Feedback einzelner Mitarbeiter eingeholt.

4 Bisherige Erfahrungen mit dem indischen Offshore-Partner

Im Rahmen der Ist-Analyse für die Identifikation von Verbesserungspotentialen und die Erfassung des Unternehmenskontextes (vergleiche Abbildung 3) wurden Interviews mit Mitarbeitern durchgeführt. Ziel war es, unter anderem, die bisherige Zusammenarbeit mit dem Offshore-Partner zu diskutieren. In diesem Abschnitt wird über die bisherigen Erfahrungen mit dem indischen Offshore-Partner berichtet. Zunächst wird darauf eingegangen, wie die Daten erhoben wurden, anschließend werden die Ergebnisse präsentiert.

4.1 Rahmen der Erhebung

Ziel der Interviews war es zum einen Herausforderungen und Besonderheiten und zum anderen Good Practices in der Zusammenarbeit mit dem indischen Offshore-Partner zu erarbeiten. Die Interviewpartner wurden nach ihren weitreichenden Erfahrungen in der Zusammenarbeit mit dem indischen Offshore-Partner oder ihren Schlüsselpositionen im Entwicklungsprozess ausgewählt. Es wurden jeweils zweistündige Interviews mit insgesamt 28 Personen geführt, die aus GEFA Fachabteilungen (6 Personen), der GEFA-IT (15 Personen), dem Onshore-Partner in Paderborn (6 Personen) und dem Offshore-Partner (eine Person) stammen. Die Interviews fanden bis auf Ausnahmen jeweils mit denselben zwei Interviewern und mit zwei Interviewten statt und waren größtenteils frei gestaltet, wobei es einen Fragenkatalog gab, auf den die Interviewer zurückgegriffen haben. Die Interviews wurden mitprotokolliert und die Ergebnisse anschließend in einer Mindmap konsolidiert und thematisch organisiert.

4.2. Erfahrungen in der Zusammenarbeit

In diesem Abschnitt wird der Teil der Ergebnisse der Ist-Analyse präsentiert, der sich auf die Zusammenarbeit mit dem indischen Offshore-Partner bezieht. Es ist jeweils markiert, ob eine Aussage von deutschen Mitarbeitern der GEFA und des Onshore-Partners (On) oder dem Mitarbeiter des Offshore-Partners (Off) getätigt bzw. bestätigt wurde.

4.2.1 Soziokulturelle und organisatorische Fallstricke und Besonderheiten

Nachfolgend werden die genannten Punkte aufgelistet, die Herausforderungen für die Zusammenarbeit bei der Offshore-Entwicklung bedeuten:

Spezialisierungsgrad (On): Verglichen mit deutschen Kollegen wiesen die indischen Entwickler eine höhere Spezialisierung auf. Während man so auf der einen Seite gezielt von der Expertise profitieren könne, sei es damit aber schwieriger, die indischen Kollegen für Aufgaben abseits ihrer Spezialisierung zu begeistern.

Qualifikationsprofil (On): Es herrsche eine unterschiedliche Auffassung zwischen Deutschen und Indern davon, wann man sich als Spezialist und Experte für eine bestimmte Technologie bezeichnen könne. Beim Offshore-Partner könne dies auch schon nach einem einwöchigen Training der Fall sein.

Hierarchiedenken (On): Zur Wahrung der Organisationshierarchien vermieden indische Kollegen den direkten Kontakt über Hierarchieebenen hinweg und kommunizierten bevorzugt über ihre Vorgesetzten.

Zustimmung und Bestätigung (On): Wenn ein indischer Kollege die Frage, ob der Sachverhalt klar sei, bejaht, könne daraus nicht direkt geschlossen werden, dass dem auch wirklich so sei.

Management Attention (Off): Die Motivation und damit Arbeitsleistung der indischen Kollegen könne durch häufige, sichtbare Aufmerksamkeit durch das Management gesteigert werden, beispielsweise durch Rundmails mit Danksagungen, bei erfolgreichen Projektabschnitten.

Aufgabepriorisierung (On): Indische Kollegen falle es schwer, Transparenz über ihre Priorisieren und Abarbeitung von Aufgaben herzustellen. Unter anderem arbeiteten sie, vermutlich durch ihr ausgeprägtes Hierarchiedenken, Aufgaben von Mitarbeitern hoher Positionen bevorzugt ab, dabei aber versäumten sie, die Auswirkungen auf die Fertigstellung anderer Aufgaben entsprechend zu kommunizieren.

Projektauslastung (On): Entwickler des Offshore-Partners hätten das Unternehmen überraschend verlassen, nachdem sie über wenige Wochen hinweg schlecht ausgelastet waren. Es wurde vermutet, dass die besagten Kollegen befürchteten, als Entwicklerkapazität an Bedeutung zu verlieren.

Einarbeitungsaufwand (On): Die Einarbeitungs- und Trainingsintensität indischer Kollegen sei sehr viel höher als bei deutschen Kollegen.

Stellenwert der Spezifikation (On): Indische Kollegen hielten sich strikt an die Spezifikation und hinterfragten diese nicht kritisch als möglicherweise lückenhaft. Dadurch übersähen sie bei der Implementierung leichter Dinge, die nicht explizit gefordert seien.

Entwicklertests (On): Mehrere Interviewpartner berichteten, dass die indischen Kollegen anders als erwartet keine Plausibilitätstests durchführten, um die von ihnen entwickelten Reports zu testen, vermutlich in der Annahme, dass jemand anders dafür zuständig sei.

Entwicklerdokumentation (On): Indische Kollegen kommentierten ihren Quellcode anders als erwartet nicht entwicklungsbegleitend, sondern erst nach Abschluss der Entwicklung, wenn der Code reif und stabil sei.

Implementierungsvorgaben (On): Indische Kollegen hätten sich in der Vergangenheit ohne vorherige oder anschließende Rücksprache über Implementierungsvorgaben hinweggesetzt, z.B., weil die Funktionalität nicht anders zu realisieren gewesen sei.

Mitarbeiterakquise (On): Es seien etwa drei bis vier Monate nötig, um vor Ort in Bangalore geeignete Mitarbeiter zu finden, vor allem wenn eine technische Spezialisierung benötigt werde.

4.2.2 Good Practices

Neben Äußerungen zu möglichen Fallstricken und Besonderheiten gab es auch Aussagen über Dinge, die sich in der Vergangenheit bei der Offshore-Entwicklung bewährt haben. Nachfolgend listen wir diese Punkte auf:

Infrastrukturmaßnahmen (On): Die Einbindung von Offshore-Entwicklern kann Eingriffe in die Unternehmens-Infrastruktur erfordern, die frühzeitig zu planen sind, beispielsweise die Konfiguration der Firmen-Firewall.

Spezifikationsdetails (On): Zu Beginn der Zusammenarbeit mit den Offshore-Entwicklern sollten Spezifikationen detaillierter ausfallen. Sobald ein Grundstock an Funktionalität vorhanden ist, könne man sich auf diese beziehen und die Spezifikationen grober fassen.

Austausch (On/Off): Die Entwickler des Offshore-Partners tauschten sich intensiv aus und würden einander bei Fragen und Problemen weiterhelfen.

Systemumgebungsinformationen (Off): Um den Offshore-Mitarbeitern die Entwicklung möglichst gut designter Software zu ermöglichen, müssten die Systemumgebung und das Entwicklungsziel möglichst gut kommuniziert werden.

Abstimmung (On/Off): Regelmäßige Abstimmungstermine, von mehrmals täglich bis zu einmal wöchentlich seien sehr wichtig. Direktere Kommunikation (Videokonferenz) sollte gegenüber Telefon oder gar E-Mail bevorzugt werden.

Projektmanagement-Software (On/Off): Der Einsatz eines Projektmanagement-Systems wie Redmine² könne die Transparenz und das „Wir-Gefühl“ entscheidend erhöhen und E-Mail-Verkehr vermindern.

Vor-Ort-Besuche (On/Off): Gegenseitige Vor-Ort-Besuche wirkten sich sehr positiv auf die Steigerung des „Wir-Gefühls“, des Domänenwissens, sowie des gegenseitigen Verständnisses und den Abbau von soziokulturellem Konfliktpotential aus.

4.3 Interpretation der geschilderten Erfahrungen

Die in diesem Abschnitt beschriebenen Erfahrungen aus der Zusammenarbeit mit dem indischen Offshore-Partner bieten einen interessanten Einblick in die Besonderheiten, die für die Methode berücksichtigt werden müssen. Es gilt allerdings zu beachten, dass es sich um subjektive Eindrücke verschiedener Personen handelt, deren Allgemeingültigkeit zunächst hinterfragt werden muss. Allerdings wurden die allermeisten Punkte durch verschiedene Interviewpartner genannt und die Beobachtungen decken sich mit dem, was in der Literatur beschrieben ist (z.B. [Ec09], [WDH08]). Die Erkenntnisse werden wie in Abschnitt 3.2 beschrieben über die Ableitung von Methodenanforderungen für die Methodenentwicklung berücksichtigt.

5 Zusammenfassung und Ausblick

In diesem Papier haben die Autoren aus der Praxis eines Methodenverbesserungsprojektes berichtet. Die resultierende Methode soll insbesondere die agile Entwicklung in Kooperation mit On- und Offshore-Partnern ermöglichen und dabei die soziokulturellen Unterschiede sowie die Gegebenheiten des Unternehmens berücksichtigen. Zum einen haben die Autoren einen systematischen Ansatz zur iterativen Methodenverbesserung vorgestellt. Zum anderen haben sie die bisherigen Erfahrungen mit der Offshore-Softwareentwicklung zusammengefasst, die im Rahmen der Ist-Analyse ermittelt wurden und in die Definition von Methodenanforderungen eingeflossen sind.

Aktuell wird die Konsolidierung der Methodenanforderungen abgeschlossen (siehe Abbildung 3) und die Zusammenstellung, Auswahl und Implementierung geeigneter Maßnahmen für die Methode steht kurz bevor. Damit werden dann auch die ersten alternativen Methodenbausteine entstehen. In wenigen Monaten sollen dann die ersten Ergebnisse der Methodenanwendung in den Pilotprojekten erzielt und die Methode daraufhin kontinuierlich verfeinert werden.

² siehe <http://www.redmine.org/>

Literatur

- [Kr99] Abel Kruchten, P.: The Rational Unified Process. An Introduction. Addison-Wesley, 1999.
- [HH08] Höhn, R., Höppner, S.: Das V-Modell XT. Anwendungen, Werkzeuge, Standards. Springer, 2008.
- [SS11] Schwaber, K., Sutherland, J.: The Scrum Guide, 2011.
- [Ec09] Eckstein, J.: Agile Softwareentwicklung mit verteilte Teams. dpunkt.verlag, 2009.
- [Fa+13] Fazal-Baqaie, M., Güldali, B., Luckey, M., Sauer, S., Spijkerman, M.: Maßgeschneidert und werkzeugunterstützt – Entwickeln angepasster Requirements Engineering-Methoden. In OBJEKTSpektrum (Online Themenspecials), no. RE/2013, S. 1–5, 2013.
- [FLE13] Fazal-Baqaie, M., Luckey, M., Engels, G.: Assembly-Based Method Engineering with Method Patterns. In Software Engineering 2013, Fachtagung des GI-Fachbereichs Softwaretechnik, S. 435–444, 2013.
- [Co02] Cockburn, A.: Agile Software Development. Addison-Wesley, 2002.
- [La12] Lacey, M.: The Scrum Field Guide. Addison-Wesley, 2012.
- [LA12] Lines, M. W., Ambler, S.: Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press, 2012.
- [WSG10] Woodward, E., Surdek, S., Ganis, M.: A Practical Guide to Distributed Scrum. IBM Press, 2010.
- [Su+10] Sutherland, J., Schoonheim, G., Kumar, N., Pandey, V., Vishal, S.: Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India. In Proc. of AGILE '09, S. 277–282, 2010.
- [WDH08] Winkler, J.K., Dibbern, J., Heinzl, A.: The Impact of Cultural Differences in Offshore Outsourcing – Case Study Results from German–Indian Application Development Projects. In Information System Frontiers, no. 10 (2), S. 243–258, 2008.

Erweiterung von Vorgehensmodellen um Aspekte der Information Governance

Daniel Burgwinkel

Kompetenzzentrum Records Management GmbH, Schweiz
daniel.burgwinkel@outlook.com

Abstract: Das Konzept der Information Governance gewinnt immer mehr an Bedeutung, da Organisationen sicherstellen müssen, dass die ständig wachsende Menge an Unternehmens- und Kundendaten ordnungsgemäß und rechtskonform aufbewahrt und fristgerecht gelöscht wird. Der Artikel analysiert Standards und Referenzmodelle im Bereich der Information Governance. Es wird ein Ansatz vorgestellt, wie sich prüfen lässt, ob ein Vorgehensmodell für die Einführung und Betrieb von IT-Systemen die Aspekte der Information Governance genügend berücksichtigt.

1 Einleitung

Der ordnungsgemäße Umgang mit Unternehmensinformationen und Kundendaten steht im Mittelpunkt der Disziplin Information Governance und wird im Zuge von Big Data und fast wöchentlichen Pressemeldungen über Datenstörfälle immer wichtiger.

Information Governance soll die Antwort auf die Frage geben, wie eine Organisation sicherstellt, dass die Gesamtheit der Informationen rechtskonform, ordnungsgemäß und produktiv verwaltet wird. Gartner definiert Information Governance als ein strategisches Framework aus Standards, Prozessen, Rollen und Metriken, welche die Verantwortlichkeiten festlegen, wie die gesamte Organisation bis auf Ebene des Mitarbeiters den Lebenszyklus der Information ordnungsgemäß und nach den Zielen der Organisation verwaltet [Ga14].

Insbesondere global agierende Unternehmen, u.a. in der Finanz- und Pharmaindustrie, haben in den letzten Jahren Organisationseinheiten für Information Governance eingeführt. Es gibt Beispiele in denen die Information Governance Funktion dem Bereich Legal&Compliance zugeordnet ist. In anderen Unternehmen wurden Fachstellen in der IT-Organisation aufgebaut, um den rechtskonformen Umgang mit Informationen sicherzustellen.

Die Aufgabenstellungen der Information Governance werden seit 2010 vermehrt in verschiedenen wissenschaftlichen Publikationen und in Fachzeitschriften der Disziplin Records Management (Schriftgutverwaltung) diskutiert [KML11], [Jo10], [Ho11]. Einige Autoren sehen Information Governance als ein neues interdisziplinäres Aufgabengebiet, welches starke Verbindungen zum Records Management und zu anderen

IT-Disziplinen hat. In 2014 wurden verschiedene Fachbücher publiziert, welche sich explizit mit Information Governance beschäftigen [Sm14], [Ba14].

Dieser Artikel soll einen Überblick über die Schnittstellen von Information Governance zu anderen Disziplinen geben (Kapitel 2). Zudem soll der aktuelle Stand der Entwicklung bei Information Governance Standards und Referenzmodellen gegeben werden. Hierbei wird analysiert, ob die bestehenden Standards auch Empfehlungen zum Projektvorgehen abgeben (Kapitel 3). Aus Sicht der Unternehmenspraxis stellt sich die Frage, ob das bereits im Unternehmen implementierte Vorgehensmodell angepasst werden kann, um den Aspekten der Information Governance gerecht zu werden. Hierfür wird in Kapitel 4 ein Leitfaden für die Prüfung vorgestellt.

2 Schnittstellen der Information Governance zu anderen Disziplinen

Bevor auf spezifische Modelle des Information Governance eingegangen wird soll aufgezeigt werden, welche Abhängigkeiten zwischen den verschiedenen Disziplinen bestehen. Obwohl jeder der aufgeführten IT-Disziplinen einen Aspekt der Verwaltung von Informationen beleuchtet, fällt es vielen Unternehmen schwer einen gesamthaften und aktuellen Überblick über alle geschäftsrelevanten Informationen zu erhalten und zu beurteilen, ob alle aufbewahrungswürdigen Informationen ordnungsmässig und sicher gespeichert sind. Hier setzt die Information Governance an, um geeignete Rollen und Prozesse im Unternehmen zu etablieren. In der folgenden Abbildung sind die relevanten Business- und IT-Aufgabengebiete aufgeführt.

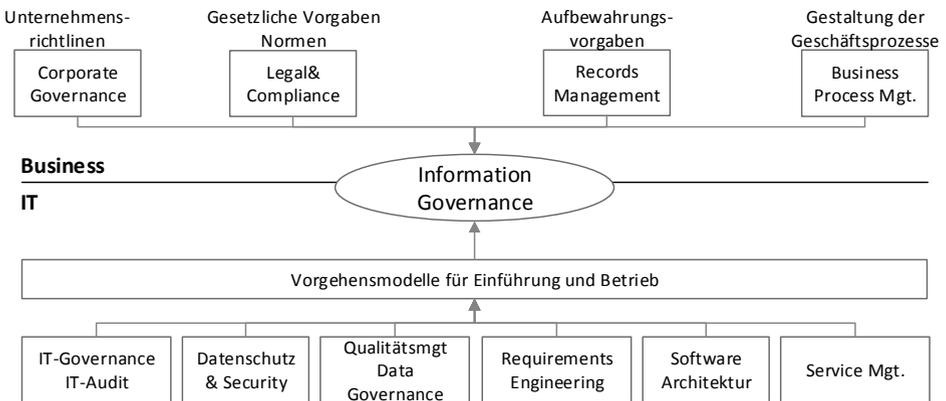


Abbildung 1: Schnittstellen Information Governance zu anderen Disziplinen

2.1 Schnittstelle Projektvorgehensmodell

Unter dem Begriff Vorgehensmodell werden Beschreibungen der Aufbau- und Ablauforganisation von Projekten zur Entwicklung und Wartung von Anwendungssystemen verstanden [GI14]. Vorgehensmodelle koordinieren die Aktivitäten der einzelnen IT-

Disziplinen, wie z.B. das Zusammenspiel von Anforderungsanalyse und Softwareentwurf. Um die Information Governance bei der Einführung eines IT-Systems sicherzustellen, muss die Organisation entsprechende Checkpunkte und Meilensteine in ihr Projektvorgehensmodell implementieren. Ein typischer Meilenstein ist hierbei, ob die rechtlichen Anforderungen an die Daten- und Dokumentenaufbewahrung berücksichtigt worden sind und Aufbewahrungsfristen definiert wurden.

2.2 Schnittstelle IT-Governance

Der Begriff Information Governance ist vom Begriff der IT-Governance und der Data Governance abzugrenzen, wobei Schnittstellen zu diesen Disziplinen bestehen. Bei der IT-Governance wird u.a. kontrolliert, ob die IT-Systeme den gesetzlichen Anforderungen entsprechen und die Geschäftsstrategie unterstützen. IT-Governancekonzepte regeln auch die Verantwortlichkeiten zwischen den Business- und IT-Organisationen.

2.3 Schnittstelle Datenschutz und IT-Security

Der ordnungsgemäße Umgang mit Kundendaten und geschäftsrelevanten Unternehmensdaten ist ein wesentliches Ziel der Datenschutz und IT-Sicherheitsmaßnahmen. Die Maßnahmen können aber nur richtig umgesetzt werden, wenn das Unternehmen weiß, welche Informationen geschäftskritisch sind und wo diese aufbewahrt werden.

2.4 Schnittstelle Qualitätsmanagement und Data Governance

Eine Aufgabe des Qualitätsmanagements ist es, sicherzustellen dass alle relevanten Vorgaben und Branchennormen bei einem IT-System berücksichtigt und geprüft sind. Insbesondere in der Pharmaindustrie ergibt es deshalb eine enge Verbindung von Qualitätsmanagements und Information Governance. Bei der weiteren Disziplin der Data Governance steht die Qualität und Korrektheit der Daten im Mittelpunkt.

2.5 Schnittstelle Requirements Engineering

Die Erfassung von Anforderungen an die rechtskonforme und ordnungsmäßige Aufbewahrung von geschäftsrelevanten Informationen muss Bestandteil jedes Softwareprojektes im Unternehmen sein. In der öffentlichen Verwaltung wurde der Ansatz verfolgt, Musteranforderungskataloge für Aktenverwaltungssysteme zu etablieren (Records Management Systeme). Beispielsweise wurde in Deutschland der DOMEA-Standard¹ [BM05] entwickelt, welcher 2013 durch das „Organisationskonzept elektronische Verwaltungsarbeit - OeV“ abgelöst wurde [BM14]. In der Schweiz wurde ein ähnlicher Standard mit dem Namen „GEVER Geschäftsverwaltung“ etabliert [Sc03].

Für die Privatwirtschaft konnten sich solche Anforderungskataloge, wie z.B. „MoReq - modular requirements for records systems“ nicht durchsetzen [DL10]. Anforderungen an

¹ Dokumentenmanagement und elektronische Archivierung im IT-gestützten Geschäftsgang

die IT-Systeme sind durch Gesetze und Vorschriften auf nationaler Ebene geregelt, z.B. in der Schweiz durch die Geschäftsbücherverordnung (GeBüV) und in Deutschland u.a. durch die „Grundsätze ordnungsmäßiger Buchführungssysteme“ (GoBS) und „Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen“ (GDPdU).

2.6 Schnittstelle Software Architektur und Unternehmensarchitektur

Beim Entwurf der Softwarearchitektur eines Anwendungssystems müssen Aspekte zur rechtskonformen und sicheren Dokumentenaufbewahrung berücksichtigt werden. Mit der Verbreitung von Cloudanwendungen stellt sich auch die Frage in welchem Land die Speicherung der Daten erfolgt und welche Risiken damit verbunden sind. In der Disziplin der Unternehmensarchitektur werden typischerweise u.a. Geschäftsarchitektur und Informations- und Datenarchitektur analysiert, geplant und dokumentiert. Detaillierte Angaben zu den Informationsbeständen, wie Aufbewahrungsfristen und Risiken, werden heute meist noch nicht in Enterprise Architecture Managementtools dokumentiert.

2.7 Schnittstelle IT Service Management und Betrieb

Die IT Infrastructure Library (ITIL) beschreibt u.a. die Prozesse für den Betrieb und die Weiterentwicklung von Anwendungen. Bei Erweiterungen und Änderungen und müssen nicht nur die IT-technischen Aspekte geprüft werden, sondern auch ob Auswirkungen auf die Informationsobjekte, z.B. geänderte Aufbewahrungsfristen, bestehen.

3 Überblick Standards Information Governance

In den letzten Jahren sind verschiedene Modelle und Standards publiziert worden, die sich explizit mit dem Vorgehen für die Einführung einer unternehmensweiten Information Governance bzw. der Einführung von Anwendungssystemen für die ordnungsgemäße Daten- und Dokumentenaufbewahrung beschäftigen.

Information Governance Standards und Modelle	Standard enthält Empfehlung zum Projektvorgehen
ISO-Norm 15489 Records Management	Ja
ISO-Norm 30300 - Management systems for records	Ja
Records Management Standards in der öffentlichen Verwaltung	Ja
GARP Generally Accepted Recordkeeping Principles	Nein
IGRM - Information Governance Reference Model	Ja, für eDiscovery
Cobit 5 Enabling Information	Nein

Tabelle 1: Information Governance Standards und Modelle

3.1 ISO-Norm 15489 Records Management

Die ISO-Norm 15489 wurde als Standard für die Schriftgutverwaltung 2001 publiziert und gibt u.a. Empfehlungen für das Projektvorgehen bei der Einführung eines Schriftgutverwaltungssystems (engl. Records Management) [IS01], [Ko11]. Typische Anwendungsfälle sind die Konzeption von Aktenmanagementsystemen und unternehmensweiten Archivierungssystemen.

3.2. ISO-Norm 30300 - Management systems for records

In 2011 wurde die ISO-Norm 30300 publiziert, welche die Einführung eines unternehmensweiten Managementsystems für Records beinhaltet [IS11]. Im zugehörigen Standard ISO 30301 werden die Anforderungen an ein unternehmensweites Records Management System definiert. Die ISO-Norm 30302 enthält Empfehlungen für die Projektumsetzung.

3.3. Records Management Standards in der öffentlichen Verwaltung

Insbesondere in der öffentlichen Verwaltung existieren Standards und Anforderungskataloge an Records Management, wie in Deutschland der „DOMEA-Anforderungskatalog 2.0“ [BM05], in der Schweiz der „Leistungskatalog GEVER-Anwendungen“ [Sc03]. Für das Projektvorgehen verweisen diese Standards auf die länderspezifischen Vorgehensmodelle der öffentlichen Verwaltung und auf die ISO-Norm 15489.

3.4 Generally Accepted Recordkeeping Principles

Der Verband ARMA hat im Jahr 2009 unter dem Titel „Generally Accepted Recordkeeping Principles (GARP)“ Prinzipien zur ordnungsgemäßen Aufbewahrung von Geschäftsdokumenten veröffentlicht [AR09] und 2010 um ein Reifegradmodell erweitert [AR13]. Der Verband ist als Vereinigung von Records Managern entstanden und hieß ursprünglich „Association of Records Managers and Administrators“ während heute immer mehr der Begriff der „Information Professionals“ im Vordergrund rückt. Unter der Bezeichnung GARP werden acht Prinzipien zusammengefasst, welche als Best-Practice im Records Management gelten. Diese Prinzipien können als Basis für ein Assessment der Situation des Information Management im Unternehmen genutzt werden. ARMA empfiehlt hierzu ein fünfstufiges Reifegradmodell. In einem Technical Report werden Handlungsempfehlungen für die Gestaltung solcher Assessments gegeben [AR14b].

3.6 IGRM - Information Governance Reference Model

Das „IGRM - Information Governance Reference Model“ wurde 2011 von der Vereinigung EDRM veröffentlicht und vereinigt die ARMA GARP Prinzipien mit den Vorgehensmodellen von eDiscovery, d.h. dem Management von elektronischen Beweismitteln

im Kontext von Gerichtsprozessen [ED11]. Das eDiscovery-Vorgehensmodell beschreibt die notwendigen Schritte, falls ein Unternehmen digitale Informationen in einen Gerichtsprozess als Beweismittel vorbringen muss.

3.7 Cobit 5 Enabling Information

Die Publikation „Cobit 5 Enabling Information“ wurde 2013 von ISACA als Ergänzung zum bestehenden COBIT 5 Kontrollframework publiziert [IS13]. Sie führt aktuelle Themen im Bereich der Information Governance an und ordnet diese dem COBIT IT-Governance Framework zu. Es wird ein Informationsmodell mit zugehörigem Phasenmodell für den Lebenszyklus von Informationen beschrieben. Es werden detailliert Qualitätskriterien für Informationsverarbeitung aufgeführt, wie z.B. Relevanz, Verfügbarkeit. Ein Schwerpunkt wird auf die Diskussion von neun aktuellen Themenbereichen gelegt, von denen sich drei Themenfelder mit Big Data beschäftigen. Für jedes der Themen wird ein Beispiel beschrieben und die betroffenen Informationen, Ziele und Lösungsansätze diskutiert. Die Themenfelder decken bekannte Bereiche wie Datenschutz, Compliance und Master Data Management ab. Durch die Nutzung von mobilen Endgeräten und Clouddiensten entstehen hier neue Chancen und zugleich Risiken. Die Diskussion von Information Governance bei Big Data ist aktuell und dürfte für ausgewählte Branchen, wie z.B. der Versicherungsbranche, von hohem Interesse sein. Durch die breite Akzeptanz von COBIT im Bereich der unternehmensinternen Prüfung ist davon auszugehen, dass dem Thema Information Governance zukünftig mehr Bedeutung in IT-Audits zugemessen wird.

3.8 Weitere Modelle

Weitere Empfehlungen und Best-Practicemodelle zu Information Governance werden vom Verband AIIM² publiziert [AI14]. Ein weiterer Ansatz für das unternehmensweite Informationsmanagement ist die MIKE 2.0 Methode³, welches auch Aktivitäten im Bereich Information Governance beinhaltet [Ri13].

3.9 Anwendbarkeit in der Praxis

Stand 2014 sind verschiedene Standards und Referenzmodelle für Information Governance publiziert, aber noch kein Modell konnte einen großen Bekanntheitsgrad erreichen. Während im Bereich Records Management die ISO-Norm 15489 sich in den letzten 10 Jahren etablieren konnte ist die neue ISO-Norm 30300 aktuell nur in Fachkreisen bekannt. Andere Standards fokussieren entweder auf bestimmte Branchen, wie z.B. die öffentliche Verwaltung oder sind eher für den IT-Audit Bereich oder auf Spezialgebiete wie eDiscovery fokussiert. Um ein Unternehmen bei der Einführung von Information Governance zu unterstützen stellen wir im nächsten Kapitel einen Ansatz vor, bestehende Vorgehensmodell um Aspekte der Information Governance zu erweitern.

² Association for Information and Image Management

³ Method for an Integrated Knowledge Environment

4 Erweiterung von Vorgehensmodellen um Aspekte der Information Governance

4.1. Leitfaden für Prüfung und Erweiterung von Vorgehensmodellen

Will eine Organisation unternehmensweite Informationen Governance sicherstellen, so muss geprüft werden, ob das im Unternehmen eingesetzte Vorgehensmodell alle Aspekte der Information Governance berücksichtigt. Im Folgenden wird ein vom Autor entwickelter Ansatz vorgestellt, welcher auf dem GARP Modell basiert [AR13]. Mit Hilfe von acht Prinzipien wird geprüft, ob ein Unternehmen die Information Governance in ihrem Projektvorgehens- und Betriebsmodell genügend berücksichtigt.

1. Principle of Accountability – Verantwortlichkeit für Information Governance
2. Principle of Transparency – Transparenz der Informationsrichtlinien
3. Principle of Integrity – Vollständigkeit/Richtigkeit
4. Principle of Protection – Schutz der Informationen
5. Principle of Compliance – Einhaltung rechtlicher Rahmenbedingungen
6. Principle of Availability – Verfügbarkeit der Informationen
7. Principle of Retention – Aufbewahrung der Informationen
8. Principle of Disposition – Entsorgung und Vernichtung der Informationen

4.2 Verantwortlichkeiten für Information Governance

Das Prinzip der Verantwortung (Accountability) ist eine wesentliche Voraussetzung für eine erfolgreiche Information Governance. Ist für einen Datenbestand oder sogar unternehmensweit nicht definiert, wer für welche Informationen in den verschiedenen Phasen des Lebenszyklus die Verantwortung trägt, so kann die Ordnungsmäßigkeit nicht sichergestellt werden. Das GARP Modell empfiehlt die Verantwortung für ein Records Management Programm im Top-Management zu verankern und durch Richtlinien die Verantwortlichkeiten bis zum Mitarbeiter klar zu definieren. Zur Prüfung des Vorgehensmodells können folgende Fragen gestellt werden:

1. Wie regelt die Organisation unternehmensweit die Verantwortlichkeiten für Daten und Dokumenten?
2. Wer ist auf Businessseite für die Information zuständig? Wer ist in der IT-Organisation zuständig? Wie sind die Aufgaben zwischen Business und IT verteilt?
3. Gibt es eine zentrale Stelle, welche die Aufbewahrung koordiniert oder ist jeder Geschäftseinheit selbst verantwortlich?

4. Ist für jedes Anwendungssystem definiert wer für die Definition der Aufbewahrungspflichten, rechtlichen Aspekte und die korrekt Datenlöschung verantwortlich ist?
5. Was passiert wenn die Verantwortlichen wechseln – wie wird sichergestellt, dass keine Datenfriedhöfe entstehen?

4.3 Transparenz und Kommunikation der Informationsrichtlinien

Will man den Reifegrad einer Organisation überprüfen, so stellt sich in diesem Bereich die Frage, ob die Weisungen und Richtlinien zum Umgang mit Information transparent sind und von den Mitarbeitern angewandt werden. Folgende Fragen können als Leitfaden dienen:

1. Wer erstellt, kommuniziert und kontrolliert die Richtlinien? In ausgewählten Unternehmen gibt es zentrale Einheiten für die Erstellung der Richtlinien und deren Kommunikation. In anderen Unternehmen sind beispielsweise Kompetenzzentren für Dokumentenmanagement und Archivierung etabliert.
2. Wie wurde den Mitarbeitern die Wichtigkeit des korrekten Umgangs mit den Daten kommuniziert, z.B. gibt es kommunizierte Richtlinien welche von der Geschäftsleitung bestätigt sind?
3. An welche Stelle können sich Mitarbeitern in der Projektphase und im Betrieb wenden, wenn es Fragen bezüglich Aufbewahrungsfristen und Löschung von Daten gibt?

4.4 Vollständigkeit und Korrektheit der Informationen

Unter der Forderung Integrität sind nicht nur die sicherheitstechnischen Aspekte zu verstehen, sondern auch das Organisationskonzept (Rollen und Prozesse) wie Glaubwürdigkeit und Verlässlichkeit der Informationen erreicht werden. In der papierbasierten Dokumentenverarbeitung nahm der Archivar eine wichtige Rolle ein. Für die digitale Verarbeitung müssen passende Rollen und Verantwortungen definiert werden.

1. Wie stellt das Unternehmen sicher, dass z.B. im Fall von Gerichtsprozessen die relevanten Dokumente vollständig und unverändert sind?
2. Gibt es eine Abteilung die sich mit eDiscovery beschäftigt oder wird eine Task Force adhoc gebildet?

4.5 Schutz und Klassifikation der Informationen

Um Informationen zu schützen sind nicht nur technische Maßnahmen notwendig, sondern auch Konzepte und Richtlinien wie Informationen im Unternehmen klassifiziert

werden, um die geschäftsrelevanten Informationen ordnungsgemäß in geeigneten IT-Systemen aufzubewahren.

1. Ist im Vorgehensmodell sichergestellt, dass Dokumente und Applikationen korrekt klassifiziert werden und geeignete Schutzmaßnahmen umgesetzt werden?
2. Wie erfolgt das Changemanagement, falls Applikationen um neue Dokumentensammlungen erweitert werden?

4.6 Compliance - Einhaltung rechtlicher Rahmenbedingungen

Bei der Einführung von neuen Informationsbeständen bzw. Migration müssen die rechtlichen und unternehmensinternen Anforderungen geklärt und umgesetzt werden.

1. Ist im Projektvorgehensmodell des Unternehmens ein Checkpunkt vorhanden, bei dem Anforderungen bzgl. Datenschutz, Aufbewahrungsdauer, Recht und Branchenstandards geprüft werden?
2. Wie werden die Attribute bzgl. Datenschutz, Aufbewahrungsdauer etc. gepflegt?
3. Was passiert bei Änderungen der rechtlichen Rahmenbedingungen – werden diese in regelmäßigen Abständen überprüft?

4.7 Verfügbarkeit der Informationen

In der papierbasierten Dokumentenwelt war das physische Archiv der Ort der Aufbewahrung und der Archivar war verantwortlich die Unterlagen zu lokalisieren und auszuhandigen. Beim Prinzip der Verfügbarkeit der digitalen Dokumente geht es nicht nur um Service Levels für die Verfügbarkeit der IT-Systeme, sondern auch um die Prozesse der Suche und Aufbereitung. In einigen Branchen verlangen Normen, dass Information innerhalb 24 Stunden einer Behörde ausgeliefert werden müssen.

4.8 Aufbewahrung der Informationen

Ein unternehmensweiter Informationsplan (auch Archivplan oder Aktenregister genannt) muss dokumentieren, wo welche geschäftsrelevanten Dokumente aufbewahrt werden und wann diese vernichtet werden müssen.

1. Werden im Vorgehensmodell bei der Einführung, Migration und Dekommissionierung von Informationssystemen die Aufbewahrungsfristen definiert?
2. Wird geprüft, ob die Applikation im Falle von Gerichtsprozessen die Aufbewahrungsfristen aussetzen muss?

3. Sind für die Applikation die Löschrprozesse definiert und werden diese protokolliert?

4.9 Entsorgung und Vernichtung der Informationen

Für die Definition von Aufbewahrungsfristen müssen die rechtlichen Vorgaben geprüft und mit den Vorgaben der Geschäftsstrategie und Risikomanagements abgestimmt werden. Aus rechtlicher Sicht gibt es Mindestauern für die Aufbewahrung von Geschäftsunterlagen und maximale Zeitspannen, z.B. Löschung von Kundendaten nach Ablauf einer Frist. Eine typische Frage ist hier wie lange der interne E-Mailverkehr aufbewahrt wird.

4.10 Einsatz in der Praxis

Da noch keine etablierten Vorgehensmodelle für Information Governance am Markt sind wurde in verschiedenen Praxisprojekten die vorgestellten Kriterien genutzt, um existierende Vorgehensmodelle von Unternehmen zu prüfen. Basis für die Analyse sind Experteninterviews mit Unternehmen in der Schweiz. In der Schweiz wurde 2011 der Verein „ARMA Swiss Chapter“ gegründet. Die Verband ARMA ist seit 1955 als Non-Profit Vereinigung von Records Managern aktiv und in 30 Ländern präsent [AR14a].

Das angewendete Vorgehen hatte verschiedene Vorteile. Zum einen sind die acht Prinzipien kurz und verständlich ohne technische Fachbegriffe formuliert und können auf das gesamte Unternehmen oder ein Organisationsbereich bzw. Datenbestand angewandt werden. Analysiert man die acht Prinzipien im konkreten Kontext eines Unternehmens kann relativ schnell aufgedeckt werden, in welchen Bereichen Handlungsbedarf besteht. Die Verantwortlichen im Unternehmen können somit für diese Schwachstellen sensibilisiert werden. Schwerpunkt der Diskussion sind typischerweise die Aspekte der Informationsaufbewahrungen, Richtlinien und Organisation.

In den GARP Prinzipien wird der Begriff eines Records Management Programms aufgeführt. Insbesondere im deutschsprachigen Umfeld ist der Begriff Records Management nicht weit verbreitet, da das Berufsbild des Records Managers eher in den USA etabliert ist. Die Idee eine Records Management Programm zu initiieren ist auch in der ISO 15489 vorhanden. Doch in Zeiten von Wettbewerbsdruck, Globalisierung und Kostenreduktion gelingt es nur wenigen Unternehmen solche Programme zu finanzieren.

5 Ausblick

Eine Herausforderung für Unternehmen ist sicherzustellen, dass in allen Projekten die Daten und Dokumente in den Anwendungssystemen rechtskonform, sicher und nutzbringend aufbewahrt und zum korrekten Zeitpunkt gelöscht werden. Hierfür bedarf es ein abgestimmtes Konzept von Richtlinien, Projektvorgehensmodell, Verantwortlichkeiten und Kontrollmechanismen.

Für die zukünftige Entwicklung stellt sich die Frage, ob sich dezidierte Standards für Information Governance durchsetzen werden. Eine Alternative zu spezialisierten Modellen ist es bestehende Vorgehensmodelle um die Aspekte der Information Governance zu erweitern.

Jede Organisation ist mit der Herausforderung konfrontiert, Unternehmens- und Kundeninformationen sicher und rechtskonform aufzubewahren. Ein Ansatz ist es für die Steuerung der Information Governance dezidierte Rollen im Unternehmen einzuführen. So haben ausgewählte Unternehmen die Rolle eines Information Governance Managers eingeführt. In den USA wird der Ansatz verfolgt das Berufsbild des Records Managers durch den Fokus auf „Information Governance“ den neuen Gegebenheiten der digitalen Welt anzupassen und den „Records Manager“ als kompetenten Ansprechpartner für die Information Governance zu etablieren, da das traditionelle Verwalten von papierbasierten Archiven immer weiter in den Hintergrund tritt. Eine Alternative zur Einführung einer neuen Rolle ist die interdisziplinären Aufgaben auf bestehenden Rollen zu verteilen. In beiden Ansätzen müssen aber die im Unternehmen eingesetzten Vorgehensmodelle so abgestimmt sein, dass alle Aspekte der Information Governance berücksichtigt werden.

Literaturverzeichnis

- [AI14] AIIM - Association for Information and Image Management: Information-Governance. <http://www.aiim.org/Resource-Centers/Information-Governance>, 31.08.2014.
- [AR09] ARMA: GARP - Generally Accepted Recordkeeping Principles, 2009.
- [AR13] ARMA: Generally Accepted Recordkeeping Principles - Brochure, 2013.
- [AR14a] ARMA: Who We Are - ARMA International.
<http://www.arma.org/r2/who-we-are>, 31.08.2014.
- [AR14b] ARMA: Auditing for Records and Information Management Program Compliance - Technical Report TR 25-2014. ARMA International, 2014b.
- [Ba14] Ballard, C.: IBM information governance solutions. IBM Corp., International Technical Support Organization, Poughkeepsie, NY, 2014.
- [BM05] BMI Bundesministerium des Innern der Bundesrepublik Deutschland: Lexikon -DOMEA-Konzept.
<http://www.bmi.bund.de/SharedDocs/Glossareintraege/DE/D/domea-konzept.html>, 31.08.2014.
- [BM14] BMI Bundesministerium des Innern der Bundesrepublik Deutschland: Organisationskonzept elektronische Verwaltungsarbeit (OeV).
http://www.verwaltung-innovativ.de/DE/E_Government/orgkonzept_everwaltung/orgkonzept_everwaltung_node.html;jsessionid=1EE2D09888D8FD00BCFF55382EE07D2F_2_cid362, 31.08.2014.
- [DL10] DLM-Forum: MoReq - modular requirements for records systems.
<http://www.moreq.info/>, 31.08.2014.

- [ED11] EDM: How the Information Governance Reference Model (IGRM) Complements ARMA International's Generally Accepted Recordkeeping Principles (GARP®), 2011.
- [Ga14] Gartner: IT Glossary Information Governance. <http://www.gartner.com/it-glossary/information-governance>, 8.05.2014.
- [GI14] GI WI-VM Fachgruppe Vorgehensmodelle für die betriebliche Anwendungsentwicklung: Homepage. <http://fg-wi-vm.gi.de/>, 01.09.2014.
- [Ho11] Hoke, G. E.: Shoring up Information Governance with GARP. In ARMA Information Management Journal, 2011.
- [IS01] ISO Internationale Organisation für Normung: Information and documentation - records management. ISO, Geneva, 2001.
- [IS11] ISO Internationale Organisation für Normung: ISO 30300:2011 - Information and documentation -- Management systems for records -- Fundamentals and vocabulary. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53732, 31.08.2014.
- [IS13] ISACA: Cobit 5 Enabling Information. ISACA, 2013.
- [Jo10] Johannsen, W.: Information-Governance: Herausforderungen in verteilten Umgebungen. In (Fähnrich, K.-P.; Franczyk, B. Hrsg.): GI Jahrestagung (2). GI, 2010; S. 311–316.
- [KML11] Kooper, M. N.; Maes, R.; Lindgreen, E. R.: On the governance of information: Introducing a new concept of governance to support the management of information. In International Journal of Information Management, 2011, 31; S. 195–200.
- [Ko11] Kos, P.: Rechtliche Anforderungen an die elektronische Schriftgutverwaltung in der Privatwirtschaft und Zertifizierungen nach ISO 15489-1 und ISO/IEC 27001. Diss. Nr. 3831 Rechtswiss. St. Gallen, 2010. s.n, S.l, 2011.
- [Ri13] Rindler, A.: Information development using mike2.0. Motion Publishing, 2013.
- [Sc03] Schweizerische Bundeskanzlei: GEVER - Geschäftsverwaltung. <http://www.bk.admin.ch/themen/egov/08205/index.html?lang=de>, 31.08.2014.
- [Sm14] Smallwood, R. F.: Information Governance: Concepts, Strategies, and Best Practices. John Wiley & Sons, 2014.

Zur Rolle des Stakeholder-Managements in IT-Projekten an Hochschulen – Erfahrungen aus der Einführung eines integrierten Campus-Management-Systems

Gunnar Auth

Hochschule für Telekommunikation Leipzig
Gustav-Freytag-Str. 43-45
04277 Leipzig
gunnar.auth@hft-leipzig.de

Abstract: Während Stakeholder-Management als Teilaufgabe in allgemeinen Prozessmodellen des Projektmanagements fest verankert ist, besteht bei IT-Projekten die Gefahr, dass durch einen nutzerzentrierten Ansatz die Bedeutung von Stakeholdern unterschätzt wird. Durch die besondere Organisationsform von Hochschulen im Spannungsfeld von akademischer Selbstverwaltung, öffentlichem Dienstrecht und studentischer Beteiligung treffen IT-Projekte dort auf ein komplexes Interessengeflecht, das sowohl Risiken als auch Chancen birgt. Der Beitrag beschreibt am Beispiel der Einführung eines integrierten Campus-Management-Systems an einer großen Volluniversität, wie sich mit Hilfe des Stakeholder-Managements diese Interessen systematisch analysieren und zur Erreichung der Projektziele gewinnbringend nutzen lassen.

1 Einleitung

Die hohe Bedeutung des Stakeholder-Managements für die erfolgreiche Durchführung von Projekten aller Art kann als weithin anerkannt betrachtet werden (bspw. [AHS13], [Ke09], [Ko10]). Zuletzt wurde dies besonders deutlich durch die Einführung eines eigenen Wissensgebiets „Project Stakeholder Management“¹ in der 2013 veröffentlichten Version 5 des weitverbreiteten Projektmanagement-Standards der internationalen Fachgesellschaft Project Management Institute (PMI) [Pr13]. Auch die bereits im Juli 2012 erschienene ISO-Norm 21500 „Leitlinien Projektmanagement“ beschreibt eine eigene Themengruppe Stakeholder², wobei die ISO-Themengruppen und die PMI-Wissensgebiete gezielt vereinheitlicht wurden [ZS13]. Weniger prominent ist dagegen häufig die Einordnung im Management von IT-Projekten. Hier wird in Literatur und

¹ In der Vorgängerversion war das Stakeholder-Management zwar bereits vorhanden, aber dem Wissensgebiet „Project Communications Management“ untergeordnet.

² Im Deutschen auch Anspruchsgruppen oder Interessengruppen.

Praxis die Sicht oftmals auf offizielle Vertreter³ des Auftraggebers sowie die Anwender beschränkt, wodurch die Gefahr entsteht, dass andere wichtige Stakeholder nicht ausreichend berücksichtigt werden [PT07]. So nennen bspw. Wieczorrek und Mertens [WM11] unter den Erfolgsfaktoren des IT-Projektmanagements lediglich die „Nutzer-Einbeziehung“ und das „Top-Management-Engagement“, eine weitergehendere Einbeziehung von Stakeholdern ist nicht vorgesehen. Allerdings finden sich auch für IT-Projekte Quellen, welche die Bedeutung eines umfassenderen Stakeholder-Managements für den Projekterfolg betonen (bspw. [Sw13, BM08, PT07]). Insbesondere bei der Erstellung bzw. Einführung neuer IT-Systeme kann ein entsprechendes Projekt erst dann als erfolgreich angesehen werden, wenn das zugehörige System die Akzeptanz der Anwender, deren Führungskräfte sowie weiterer Stakeholder (bspw. Datenschutzbeauftragter) gefunden hat. Es gilt daher in solchen Projekten, die maßgeblichen Stakeholder frühzeitig zu identifizieren, zu analysieren und in ein aktives Stakeholder-Management einzubinden [Sw13].

Im Vergleich zu Wirtschaftsunternehmen oder Verwaltungsorganisationen unterscheidet sich das Stakeholder-Umfeld an Hochschulen deutlich. Durch Faktoren wie die sog. akademische Selbstverwaltung, den Beamtenstatus von Professoren und anderem Hochschulpersonal, die besondere Stellung der Studenten oder die immer wieder wechselnde Zusammensetzung von Hochschulgremien entsteht ein komplexes Organisationsgefüge, in dem Entscheidungen eher bottom-up als top-down, eher dezentral statt zentral und eher konsensbasiert als auf Basis formaler Entscheidungskompetenz getroffen werden. Aufgrund dieser Unterschiede sind andernorts bewährte Konzepte, Strategien und Techniken des Stakeholder-Managements im Hochschulbereich wirksamer, wenn ein Bewusstsein für diese spezifischen Gegebenheiten vorhanden ist und das Stakeholder-Management darauf hin ausgerichtet wird.

Der vorliegende Artikel beschreibt anhand eines Fallbeispiels Erfahrungen mit Stakeholder-Management an einer großen deutschen Volluniversität. Den Rahmen für das Fallbeispiel bildet ein Einführungsprojekt für ein integriertes Campus-Management-System (CMS) zur Unterstützung von Verwaltungsprozessen für Lehre und Studium an dieser Universität. Es handelte sich dabei für Hochschulverhältnisse um ein vergleichsweise großes IT-Projekt in dem Sinne, dass weite Teile der Ablauf- und Aufbauorganisation von der Systemeinführung betroffen waren. Daraus resultierte wiederum eine große Anzahl von Stakeholder-Gruppen auf unterschiedlichen Hierarchieebenen und mit verschiedenen Interessen. Der engere Betrachtungsfokus liegt auf einer frühen Projektphase, in der Auswahl und Beschaffung der CMS-Standardsoftware erfolgten. Der Artikel analysiert zunächst allgemein Stakeholder-Gruppen an Hochschulen und geht dabei auf Besonderheiten ein. Anschließend wird aus Sicht des Stakeholder-Managements das Fallbeispiel der Auswahl und Beschaffung einer CMS-Standardsoftware im Rahmen eines Einführungsprojekts beschrieben. Die dabei gemachten Erfahrungen werden einer kritischen Bewertung unterzogen, um abschließend ein Fazit ziehen zu können.

³ Im Sinne der Lesbarkeit wird auf geschlechtergerechte Sprache verzichtet. Personenbezogene Bezeichnungen in diesem Beitrag sind allgemein nicht geschlechtsspezifisch zu verstehen.

2 Stakeholder-Management in Hochschul-Projekten

Wie in nahezu allen Organisationen ist auch an Hochschulen das Arbeiten in Projekten eine weit verbreitete Form der Arbeitsorganisation. Dabei kann zunächst grob zwischen den beiden Bereichen akademische Forschung und Hochschulverwaltung unterschieden werden. Im dritten Kernaufgabenbereich von Hochschulen, der akademischen Bildung, werden zwar ebenfalls bspw. Studienprojekte durchgeführt. Allerdings sind dies Projekte mit vergleichsweise kleinem Umfang, kurzer Dauer und wenig Beteiligten. Zudem zielen sie i.d.R. nicht auf Veränderungen oder Neuerungen in den Hochschulverhältnissen. Letzteres ist auch bei Forschungsprojekten typischerweise nicht der Fall. Der Fokus dieses Beitrags lässt sich damit auf Projekte der Hochschulverwaltung eingrenzen, wobei diese durchaus Auswirkungen auf die anderen Bereiche haben oder auch aus diesen Bereichen initiiert sein können.

2.1 CMS-Einführung als Beispiel für IT-Projekte an Hochschulen

In den Hochschulverwaltungen werden zunehmend Projekte gestartet, die auf eine organisatorische Transformation zur Verbesserung der Leistungsfähigkeit vor dem Hintergrund eines steigenden Veränderungsdrucks zielen (vgl. [SAA12]). Die Ziele solcher Projekte beinhalten dabei häufig die Verbesserung von Effizienz und Effektivität der Verwaltungsprozesse, übergreifender Prozesse für die Zusammenarbeit mit Ministerien, anderen Behörden oder Forschungspartnern sowie der Servicequalität für Hochschulangehörige wie Studenten, (wissenschaftliche) Mitarbeiter, Gäste etc. (vgl. [Bi13]). Dabei gilt die bedarfsgerechte Verfügbarkeit entscheidungsrelevanter Informationen als ein Schlüsselfaktor für den Erfolg dieser Verbesserungsbestrebungen (vgl. [Ha01], [Ko14]).

Als Konsequenz handelt es sich bei den zugehörigen Projekten häufig im Kern um IT-Projekte zur Gestaltung, Einführung oder Weiterentwicklung von Informationssystemen. Der Begriff des Informationssystems beinhaltet in der Wirtschaftsinformatik seit jeher sowohl eine technische Perspektive, welche maschinelle Aufgabenträger (bspw. Rechner, Rechnersysteme) umfasst, als auch eine soziale Perspektive, in deren Mittelpunkt die Aufgabendurchführung durch Menschen steht. Menschliche und maschinelle Aufgabenträger sind durch Kommunikationsbeziehungen miteinander verbunden, wodurch die arbeitsteilige Durchführung von Aufgaben der Informationsverarbeitung erst ermöglicht wird. Informationssysteme sind somit sozio-technische Systeme, deren Gestaltung immer auch wechselseitigen Einflüssen mit der umgebenden Organisation unterliegt, die es zu berücksichtigen gilt (vgl. [Ga13], [AI+11]).

In der Diskussion um die gegenwärtig an vielen Hochschulen zu beobachtende Einführung oder Erneuerung von geschäftsprozessunterstützender Standardsoftware (bspw. [Is14], [OA14]) wird der sozialen bzw. organisatorischen Perspektive eine besondere Bedeutung beigemessen [Ja+09]. Diese Hervorhebung geht im Extremfall bis zu einer weitgehenden Konzentration auf organisationssoziologische Aspekte, welche die Gestaltung technischer Systemkomponenten nur noch als untergeordnete Faktoren für eine erfolgreiche Softwareeinführung betrachtet [MW13].

Aus Sicht des Projektmanagements verspricht die Antwort auf die Frage, ob eine Softwareeinführung (an einer Hochschule) eher ein IT- oder eher ein Organisationsprojekt darstellt, eher wenig Erkenntniszuwachs. In beiden Fällen gehört das Stakeholder-Management zu den Kernaufgaben des Projektmanagements und bietet einen methodischen Zugang zur Organisationsentwicklung. Durch Anwendung des Stakeholder-Managements werden organisationsbezogene Risiken erkannt und können frühzeitig berücksichtigt werden. Auswirkungen des Projekts auf soziale Gesichtspunkte wie Arbeitsteilung, Kooperation und Machtausübung lassen sich untersuchen und hinsichtlich ihrer Wechselwirkung mit dem Projekt analysieren. Eine zusätzliche Relevanz ergibt sich für CMS-Projekte durch das Ziel einer durchgängigen IT-Unterstützung der organisatorischen Prozesse für Lehre und Studium [AA13]. An diesen Prozessen sind Professoren, akademische Mitarbeiter, Studenten und Verwaltungsmitarbeiter beteiligt. Damit sind alle wesentlichen Personengruppen der Organisation Hochschule von projektinduzierten Veränderungen betroffen. Vor dem Hintergrund des eingangs beschriebenen Veränderungsdrucks lässt sich ein solches Vorhaben als Schritt in einem IT-basierten Transformationsprozess von Hochschulen begreifen (vgl. [SAA12]).

2.2 Stakeholder und Stakeholder-Management

Der ursprüngliche Stakeholder-Begriff lässt sich auf Freeman [Fr84] zurückführen, der ihn als ein Konzept des strategischen Managements zur Analyse und Beeinflussung von Personengruppen auf den Erfolg eines Unternehmens entwickelte. Dieses Konzept wurde bald auch auf das Projektmanagement übertragen, wo es heute unter der Bezeichnung Stakeholder-Management als wichtige Teilaufgabe gilt, die in der Verantwortung des Projektleiters liegt [LJA10]. Die Grundzüge des Stakeholder-Managements in Projekten können als allgemein anerkannt betrachtet werden, was sich u. a. durch die Beschreibung in internationalen Standards wie dem „Project Management Body of Knowledge“ (PMBoK) des PMI [Pr13] oder der ISO-Norm 21500 [ZS13] ausdrückt. Jedoch besteht nach wie vor noch Forschungsbedarf bspw. hinsichtlich des Einflusses und der Beeinflussbarkeit von Stakeholdern, deren Beziehungen untereinander sowie zu Akteuren und Ablauf von Projekten (z. B. [AHS13, Da14, ML14]).

Der Schwerpunkt des vorliegenden Beitrags liegt allerdings nicht auf einer Weiterentwicklung der Stakeholder-Theorie, sondern er soll das Verständnis der Anwendung anerkannter Konzepte des Stakeholder-Managements in Projekten einer spezifischen Anwendungsdomäne, hier IT-Projekte an Hochschulen, verbessern. Zu diesem Zweck wird als methodischer Rahmen das prozessorientierte Stakeholder-Management des PMI-Standards PMBoK Version 5 zugrunde gelegt. Ein Stakeholder ist dort definiert als „an individual, group or organization who may affect, be affected by, or perceive itself to be affected by a decision, activity or outcome of the project“ [Pr13]. Das PMBoK-Wissensgebiet Project Stakeholder Management beschreibt folgende (Teil-)Prozesse [Pr13], die sich als an den Deming-Zyklus angelehntes, iteratives Vorgehensmodell für das Stakeholder-Management in einem Projekt interpretieren lassen:

1. Identify Stakeholder: Erforderliche Aktivitäten zur Erkennung von Einzelpersonen, Personengruppen oder Organisationen, die eine Motivation haben oder entwickeln könnten, auf das Projekt in ihrem jeweiligen Interesse Einfluss zu

nehmen (positiv oder negativ). Beinhaltet außerdem die Analyse, Bewertung und Dokumentation der Beweggründe.

2. Plan Stakeholder Management: Ableitung von Strategien und Planung von Maßnahmen zum Umgang mit Stakeholdern auf Basis der vorangegangenen Analysen.
3. Manage Stakeholder Engagement: Umsetzung der Maßnahmen durch Kommunikation und Zusammenarbeit mit den Stakeholdern.
4. Control Stakeholder Engagement: Kontrolle der Maßnahmenwirksamkeit und Überwachung der Stakeholder-Beziehungen über den Projektverlauf.

Die Anwendung dieser allgemeinen Schrittfolge bei einer CMS-Einführung wird im nächsten Abschnitt beschrieben. Der Hauptfokus liegt dabei einerseits auf der inhaltlichen Ausgestaltung der beiden ersten Schritte, da hier die Besonderheiten der Anwendungsdomäne den größten Einfluss haben. Andererseits stehen aus Sicht des Einführungsprojekts Auswahl und Beschaffung der CMS-Standardsoftware im Mittelpunkt. Dabei wird deutlich gemacht, dass in dieser frühen Projektphase dem Stakeholder-Management eine besondere Bedeutung zukam, da mit der Auswahlentscheidung zugleich der Grundstein für die spätere Akzeptanz des Systems gelegt wurde.

3 Fallbeispiel Einführung eines integrierten Campus-Management-Systems

Bei der betrachteten Hochschule handelt es sich um eine Volluniversität klassischen Typs mit einem breiten Fächerspektrum, in dem bis auf Fächer der Ingenieurwissenschaften nahezu alle Studienrichtungen vertreten sind. Mit rund 28.000 Studierenden und 480 Professoren⁴ gehört sie zu den größeren Universitäten in Deutschland. Die Universität gliedert sich in 14 Fakultäten, die üblichen zentralen Einrichtungen (bspw. Bibliothek, Rechenzentrum, Sprachenzentrum, Archiv) sowie weitere Einrichtungen, die spezifische Aufgaben in Forschung und Lehre wahrnehmen. Die Universitätsleitung besteht aus einem klassischen Rektorat⁵, dem neben dem Rektor als Vorsitzenden der für die Verwaltung zuständige Kanzler sowie drei Prorektoren angehören. Wichtigstes Entscheidungsgremium ist der Senat, in dem die Professoren gegenüber den anderen Mitgliedsgruppen mit einer Stimme die Mehrheit haben. Mit der Umsetzung der Bologna-Reform wurde frühzeitig begonnen, so dass zum Start des hier betrachteten Projekts nahezu alle Studiengänge modularisiert und auf Bachelor- bzw. Masterabschluss umgestellt waren.

⁴ Zahlenangaben mit Stand vom WS 2013/14.

⁵ Davon unterscheiden lassen sich Präsidialhochschulen, die von einem Präsidenten geleitet werden, der häufig weitergehende Befugnisse als ein Rektor hat.

3.1 Projekt zur Einführung eines integrierten Campus-Management-Systems

Wie an vielen deutschen Hochschulen wird in der zentralen Verwaltung der betrachteten Universität die Anwendersoftware des Herstellers HIS⁶ eingesetzt, so bspw. für die Verwaltung von Personal, Finanzen, Gebäude und Flächen sowie auch für Studentenverwaltung und Studienorganisation. Diese Software wurde in ihrer Kernfunktionalität für die Anforderungen einer Studienorganisation entwickelt, wie sie vor der Bologna-Reform bestand. Modularisierte Studiengänge waren daher nicht vorgesehen und können nur rudimentär und mit hohem manuellem Aufwand verarbeitet werden. Aus dieser Erkenntnis heraus war bereits 2005 ein Projekt zur Einführung einer neuen Software zur Unterstützung der modularisierten Studiengänge gestartet worden. Trotz intensiver Bemühungen konnte dieses Projekt aufgrund von organisatorischen und technischen Problemen nicht erfolgreich abgeschlossen werden und wurde Mitte 2007 endgültig abgebrochen.

Aufgrund der negativen Erfahrungen dauerte es bis 2009, bis das (inzwischen neu besetzte) Rektorat sich entschied, die Einführung einer dringend benötigten CMS-Lösung nunmehr auszuschreiben, um auf diese Weise eine funktional adäquate und wirtschaftliche Beschaffung zu gewährleisten. Wegen der besonderen Vorgeschichte und der hohen Bedeutung, die einer Bologna-konformen Ausgestaltung von Lehre und Studium an der Universität zugemessen wird, erschien eine Ausschreibung nicht nur gerechtfertigt, sondern letztlich unumgänglich. Als Vergabeverfahren wurde der Wettbewerbliche Dialog gewählt, um für die spezifischen Anforderungen gemeinsam mit den Anbietern passende IT-gestützte Verfahren erarbeiten zu können.

3.2 Stakeholder-Management im Projekt

Die erweiterte Projektleitung (Projektleiter, Teilprojektleiter und Leiter des Projektbüros) hatte vor dem Hintergrund der vielschichtigen Interessenlage die hohe Bedeutung des Stakeholder-Managements frühzeitig erkannt und den unter 2.2 beschriebenen Prozess im Projektmanagement etabliert, u. a. durch Dokumentation im Projekthandbuch. Nachfolgend werden das konkrete Vorgehen und die erzielten Ergebnisse vom Start des Projekts bis zum Treffen der Auswahlentscheidung für eine CMS-Standardsoftware beschrieben.

3.2.1 Identifizierung und Analyse von Stakeholdern

Als wesentliche Prozessinputs wurden für das Identifizieren der Stakeholder auch in diesem Projekt organisationsbezogene Dokumente ausgewertet. Während man in Unternehmen dazu häufig aussagekräftige Organigramme vorfindet, waren diese an der betreffenden Universität nur teilweise vorhanden und nicht einheitlich aufgebaut. Wie allgemein in der öffentlichen Verwaltung sind stattdessen auch bei zentralen Hochschulverwaltungen Geschäftsverteilungspläne üblich, die neben der Organisationsstruktur auch die Aufgabenverteilung bis auf Ebene der Sachbearbeiter dokumentieren. In den dezent-

⁶ HIS Hochschul-Informationen-System eG, <http://www.his.de/>.

ralen Organisationseinheiten (Fakultäten und Einrichtungen) waren solche oder vergleichbare Dokumente zwar teilweise ebenfalls vorhanden, aber nicht immer ohne Weiteres für das Projektteam zugänglich.

Als Ausgangspunkt für die Erschließung der inneren Struktur der Universität wurde die Grundordnung herangezogen. Laut einschlägigem Landeshochschulgesetz hat sich jede Hochschule eine Grundordnung zu geben, welche die innere Struktur und die innere Organisation im Grundsatz ausgestaltet. Neben dieser Bestimmung lieferte das Landeshochschulgesetz weitere wesentliche Informationen über Aufbau und Organisation der Hochschule, Zuständigkeiten und Verantwortlichkeiten.

Besonders hilfreich für die Stakeholder-Identifizierung waren hierbei die Bestimmungen über die Gruppen, in welche die Gesamtheit der Universitätsmitglieder⁷ unterteilt ist. Je eine Gruppe bilden:

1. die Hochschullehrer (Professoren, Juniorprofessoren),
2. die akademischen Mitarbeiter (wissenschaftliche oder künstlerische Mitarbeiter einschließlich der Akademischen Assistenten, Lehrkräfte für besondere Aufgaben, wissenschaftlichen oder künstlerischen Hilfskräfte),
3. die Studenten sowie
4. die sonstigen Mitarbeiter (Beschäftigte in der Verwaltung).

Die Gruppen sind deshalb so bedeutsam, weil sowohl Leistungserstellungsprozesse (bspw. Durchführung einer Prüfung) als auch Entscheidungsprozesse kooperativ zwischen ihnen ablaufen. Als Beteiligte an den Leistungserstellungsprozessen im Bereich Lehre und Studium zählen alle Gruppen zu den späteren Nutzern des Systems, woraus sich ein unmittelbares Projektinteresse ergibt. Die Berücksichtigung der jeweiligen Anforderungen beeinflusst maßgeblich die spätere Akzeptanz eines IT-Systems und damit den Projekterfolg. Zudem ist gemäß Landeshochschulgesetz die Mitwirkung in der Selbstverwaltung der Hochschule „Recht und Pflicht aller Mitglieder“. Aus diesem Grund haben die Gruppen Anspruch auf Vertretung in wichtigen Organen wie bspw. dem Senat oder für bestimmte Aufgaben gebildete Kommissionen. Die Gruppen verfügen über einen gesetzlich legitimierten Einfluss auf die Hochschule betreffende Entscheidungen, der über die betriebliche Mitbestimmung hinausgeht. Als öffentlich-rechtliche Einrichtung unterliegt eine staatliche Hochschule zudem dem Personalvertretungsrecht und hat daher einen Personalrat zur Arbeitnehmervertretung zu bilden, ähnlich dem Betriebsrat von Unternehmen der Privatwirtschaft. Wegen seiner Beteiligungsrechte bei Entscheidungen in personellen, sozialen, organisatorischen und weiteren Angelegenheiten wurde der Personalrat als wichtiger Stakeholder betrachtet, da es bei der CMS-Einführung fast zwangsläufig zur organisatorischen Veränderungen kommt.

Auch für das Verständnis der Entscheidungsprozesse einer Hochschule ist das Hochschulgesetz eine wesentliche Informationsquelle, da hier die Zuständigkeit der Zentral-

⁷ Neben Mitgliedern kennt das betreffende Landeshochschulgesetz auch noch Angehörige, die nur geringfügig an der Hochschule beschäftigt sind, bspw. externe Lehrbeauftragte.

organe⁸ geregelt ist und damit die Antwort auf die Frage „Wer entscheidet über was?“. Hierin liegt ein weiterer Unterschied zu Unternehmen, bei denen Entscheidungen zu meist entlang einer klaren Hierarchie getroffen werden. An der betrachteten Universität hängt die Zuständigkeit vom inhaltlichen Bezug der Entscheidungsfrage ab, wobei grundsätzlich zwischen akademischen Angelegenheiten und Verwaltungsangelegenheiten unterschieden wird. Gerade bei weitreichenden Entscheidungen, die sich zudem auf die Mittelverteilung auswirken, ist die Zuständigkeit jedoch häufig strittig, wodurch Entscheidungen immer wieder verzögert werden. In der Zusammenarbeit der Gremien gibt es zudem noch feine Unterscheidungen bei der Art der Beteiligung, die von Informationspflicht über das Recht zur Stellungnahme bis zur Beschlussfassung reichen. Gerade auch im Falle der Einführung eines Campus-Management-Systems lässt sich trefflich darüber diskutieren, ob es eher in die alleinige Zuständigkeit des Kanzlers und/oder des Rektorats fällt oder wichtige Entscheidungen auch einen Beschluss des Senats erfordern. Die Kenntnis der komplexen und oft langwierigen Entscheidungsprozesse ist daher eine wesentliche Voraussetzung, um Terminpläne in Hochschulprojekten realistisch planen und einhalten zu können. Die jeweiligen Entscheidungsträger sind entsprechend als wichtige Stakeholder zu betrachten.

Im Sinne eines systematischen Vorgehens zu einer möglichst umfassenden Erkennung von Stakeholdern empfiehlt sich vor diesem Hintergrund ein Top-Down-Ansatz, der mit der Auswertung des jeweiligen Landeshochschulgesetzes beginnt. Dieses bietet eine Übersicht über nachrangige Hochschulordnungen, in denen bspw. Fakultäten ihre Angelegenheiten regeln oder die als übergreifende Ordnungen akademische Angelegenheiten wie Immatrikulation, Studium oder Prüfungen hochschulweit festlegen. Schließlich bot das Hochschulgesetz im vorliegenden Fall weitere Unterstützung durch eine Klausel, wonach Hochschulordnungen öffentlich bekannt zu machen sind. Die betreffende Universität hatte dies in Form einer Online-Datenbank umgesetzt, wodurch die Recherche deutlich vereinfacht wurde.

In den betrachteten Projektzeitraum fiel auch die Konstituierung eines Hochschulrates als neuartiges Zentralorgan, das als mehrheitlich extern besetztes Gremium eine gesetzlich verankerte Kontrollfunktion ausübt, dabei nicht unähnlich der Funktion eines unternehmerischen Aufsichtsrates. Da der Hochschulrat u. a. für die Genehmigung der Entwicklungsplanung und des Wirtschaftsplanentwurfes zuständig ist, befasst er sich zwangsläufig auch mit der Ressourcenplanung von strategischen Projekten, zu welchen auch das betrachtete CMS-Einführungsprojekt zählte.

Aus den im Projektauftrag formulierten Verbesserungszielen für die Organisation von Lehre und Studium ergab sich direkt ein besonderes Projektinteresse für die Gruppe der Studenten, die sog. Studentenschaft. Gerade Hochschulexterne, wie bspw. Berater, neigen nicht selten dazu, den Einfluss der Studentenschaft zu unterschätzen. Auch wenn die gesetzliche Grundlage in den Bundesländern teils unterschiedlich geregelt ist und den Studenten mal mehr und mal weniger Rechte zuerkennt, sollte das Aktionspotenzial der Studenten nicht unterschätzt werden. Weil sie nicht in einem dienstrechtlichen Unterstellungsverhältnis zur Hochschule stehen, sind den Ausdrucksformen studentischen Protests kaum Grenzen gesetzt. An der FU Berlin organisierten bspw. Studierende Ende

⁸ Im Fallbeispiel: Senat, Erweiterter Senat, Rektor, Rektorat, Prorektoren, Kanzler, Hochschulrat.

2005 als Ausdruck ihrer Kritik an der Einführung eines SAP-basierten Campus-Management-Systems einen einwöchigen „Warnstreik“, durch den das Projekt Gegenstand einer öffentlichen Diskussion wurde [Wo14]. Andererseits sind die Studenten auch potenzielle Verbündete für die Sache des Projekts, da sie ja mit der CMS-Einführung von verbesserten Studienbedingungen profitieren sollen. Diese Sichtweise führte im vorliegenden Fallbeispiel zur Planung und erfolgreichen Umsetzung einiger speziell auf die Studentenschaft zielenden Maßnahmen, die im folgenden Abschnitt beschrieben werden.

Durch die vergleichsweise komplexe und vielschichtige Organisation entsteht die Gefahr, bei der Identifizierung wichtige Stakeholder zu übersehen. Um dies zu vermeiden und auch neuen Projektmitarbeitern⁹ eine schnelle Orientierung über die Grundzüge der Universitätsorganisation zu ermöglichen, empfiehlt sich eine grafische Darstellung in einem konzeptionellen Bezugsrahmen, ähnlich einem Organigramm. Abbildung 1 zeigt eine solche Darstellung für die betrachtete Universität:

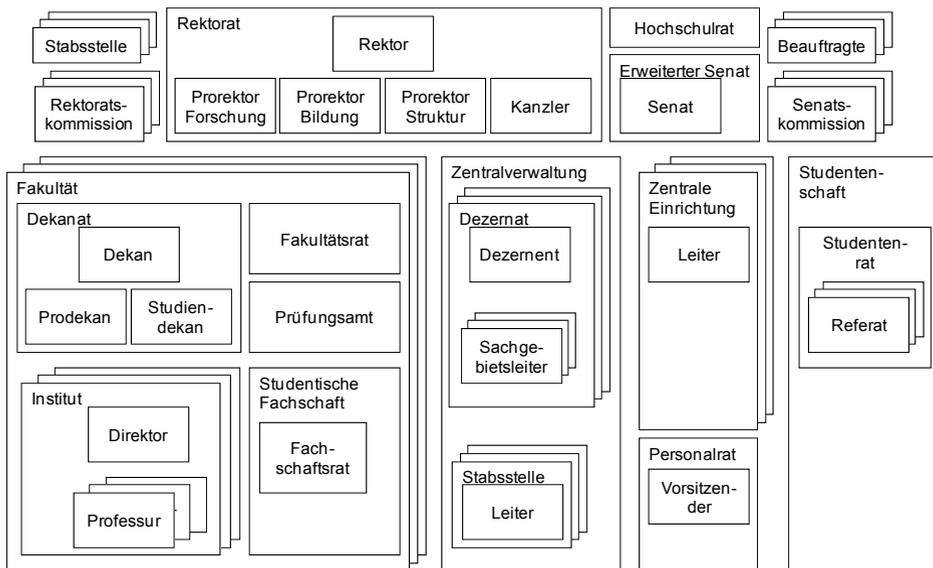


Abbildung 1: Organisation der Universität des Fallbeispiels

Aus dem Bezugsrahmen lassen sich zunächst direkt Stakeholder-Gruppen ableiten, wobei i. d. R. nicht alle Mitglieder einer Stakeholder-Gruppe von gleicher Relevanz für das Projekt sind. Bspw. wurde aus der Gruppe der Beauftragten der Datenschutzbeauftragte als sehr relevant bewertet, der Umweltschutzbeauftragte dagegen als nicht relevant.

Neben der formalen Organisation wurden bei der Stakeholder-Erkennung auch informelle Beziehungen und Einflüsse untersucht, um im nächsten Schritt Meinungsführer innerhalb der Stakeholder-Gruppen zu identifizieren. Für die Ermittlung solcher Informationen erwiesen sich bestimmte Universitätsmitarbeiter als besonders wertvoll. Aufgrund

⁹ Nicht selten kommen diese von außerhalb der Hochschule wie bspw. externe Berater oder eigens eingestellte Projektmitarbeiter.

langjähriger Dienstzeit und die Art ihrer Arbeitsaufgaben verfügten diese Mitarbeiter über Einblicke in vielfältige, übergreifende Prozesse und haben so ein umfangreiches Wissen über Zuständigkeiten, bereichsübergreifende Zusammenhänge sowie ein weitreichendes Netzwerk, das sich über die ganze Universität spannt. Zu diesem Personenkreis zählten bspw. persönliche Referenten der Rektorsratsmitglieder, Dekanatsräte, Mitarbeiter von Gremiengeschäftsstellen oder auch Institutssekretärinnen. Als hilfreich erwies sich hierbei die Rolle des Projektpaten, der bereits in einem ganz frühen Stadium für das Projekt gewonnen werden konnte. Es handelte sich dabei um einen emeritierten Professor für Wirtschaftsinformatik, der als ehemaliger Institutsdirektor und Dekan weiterhin Aufgaben in Lehre und Forschung wahrnahm. Neben der fachlichen Kompetenz waren vor allem seine persönlichen Kontakte in nahezu alle Bereiche und auf allen Hierarchieebenen für die Projektleitung eine große Unterstützung.

Für die Analyse und Bewertung der Stakeholder wurden die üblichen Kriterien „Interesse“ und „Einfluss“ angewandt. Mit den jeweiligen Ausprägungen „hoch“ und „niedrig“ ergibt sich die klassische Vierfeldermatrix für die Stakeholder-Analyse (vgl. [Pr13]).

3.2.2 Planung und Umsetzung von spezifischen Maßnahmen

In Abhängigkeit von der Positionierung eines Stakeholders anhand seines Interesses und seines Einflusses in der Vierfeldermatrix wurde eine von vier Grundstrategien zum weiteren Umgang mit diesem Stakeholder gewählt [Pr13, BM08]:

Interesse	Einfluss	Strategie	Beschreibung
Hoch	Hoch	Manage Closely	Stakeholder mit dem größten Potenzial zur Projektbeeinflussung (positiv/negativ). Als Unterstützer wertvoll, als Gegner gefährlich. Erfordern hohe Aufmerksamkeit und aktive Einbeziehung.
Hoch	Niedrig	Keep Informed	Im Austausch für Informationen über das Projekt lassen sich von Unterstützern andere relevante Informationen gewinnen. Bei Gegnern führt zielgerichtete Information ggf. zu besserem Verständnis/positiverer Haltung.
Niedrig	Hoch	Keep Satisfied	Fortlaufende Risikobetrachtung, um Auftreten von negativem Interesse zu vermeiden bzw. Chancen auf positives Interesse zu nutzen.
Niedrig	Niedrig	Monitor	Mit geringstmöglichem Aufwand beobachten, um auf Veränderungen reagieren zu können.

Tabelle 1: Grundlegende Stakeholder-Strategien

Aus den bereits ausgeführten Gründen wurde die Studentenschaft bereits zu Projektbeginn als Stakeholder-Gruppe mit hohem Interesse und hohem Einfluss identifiziert. Innerhalb dieser Gruppe übernehmen an der betrachteten Universität auf zentraler Ebene der Studentenrat und auf dezentraler Ebene die Fachschaftsräte die offizielle Vertretung der studentischen Interessen (siehe Abbildung 1). Der Studentenrat hatte früh Interesse an dem CMS-Vorhaben gezeigt, wobei ein eher kritischer Standpunkt eingenommen wurde, was sich u. a. auf das gescheiterte Vorläuferprojekt zurückführen ließ. Innerhalb des Studentenrats wurde die Zuständigkeit für das Projekt an das Referat Datenschutz übergeben, womit ein konkreter Ansprechpartner für das Projekt zur Verfügung stand, der ebenfalls als wichtiger Stakeholder eingestuft wurde. Im Folgenden werden am Beispiel der Studentenschaft die Planung und Umsetzung von spezifischen Stakeholder-Maßnahmen von Projektbeginn bis zum Treffen der Auswahlentscheidung für ein CMS-Produkt dargestellt.

Da zu Beginn des Projekts die Grundhaltung der Studentenschaft als kritisch/negativ eingeschätzt wurde, setzte sich die Projektleitung das Ziel, durch geeignete Maßnahmen die Grundhaltung positiv zu verändern und dabei die „Manage Closely“-Strategie umzusetzen. Folgende Maßnahmen erwiesen sich als besonders wirksam.¹⁰

Einbindung in Projektgremien

Bei der Zusammenstellung des Projektleitungskreises wurde für den Studentenrat ein stimmberechtigter Sitz vorgesehen. Dazu wurde der Studentenrat gebeten, einen offiziellen Vertreter sowie einen Stellvertreter für die Mitarbeit im obersten Projektgremium zu benennen. Der Studentenrat übertrug diese Aufgabe an den Referenten für Datenschutz, wodurch das Projekt in der internen Organisation der Studentenschaft gut verankert wurde.

Für das Ausschreibungsverfahren wurde im Auftrag des Lenkungskreises eigens eine Beschaffungskommission gebildet, um eine breitere Beteiligung der Fakultäten und anderen betroffenen Bereichen als im Lenkungskreis zu ermöglichen. Auch hier wurde der Studentenrat zur Mitarbeit aufgefordert, für die sinnvollerweise die beiden bereits im Lenkungskreis eingebundenen Studenten benannt wurden.

Exklusive Informationsveranstaltungen

Dazu zählten sowohl Projektpräsentationen vor dem Plenum des Studentenrates als auch speziell auf die studentischen Belange ausgerichtete Informationsveranstaltungen, zu denen die gesamte Studentenschaft eingeladen wurde.

¹⁰ Wie auch andere Stakeholder-Gruppen wurde die Studentenschaft darüber hinaus explizit im Kommunikationsmanagement als Zielgruppe berücksichtigt.

Umfrage zur Erwartungshaltung

Im Rahmen der Anforderungsspezifikation für die Leistungsbeschreibung des Ausschreibungsverfahrens wurde an einer ausgewählten Fakultät eine Umfrage zur Erwartungshaltung der Studenten durchgeführt. Mit der Konzeption und Ausführung der Umfrage wurde die an der Universität bestehende studentische Unternehmensberatung beauftragt. Damit konnte zusätzlich zu den ermittelten Informationen auch das Interesse des Projekts bei den beteiligten Studenten gesteigert werden. Die Beschränkung auf eine Fakultät war dabei alleine der Knappheit von Zeit und Ressourcen geschuldet.

Anforderungsworkshop mit Studentenvertretern

Basierend auf den Umfrageergebnissen wurde ein Anforderungsworkshop durchgeführt, zu dem alle Fachschaften offiziell eingeladen wurden. Leider war die Resonanz bei den Fachschaften sehr unterschiedlich, was wiederum auf Handlungsbedarf bei der Steigerung des (positiven) Interesses hinwies.

Einbeziehung in Prozessanalyse

Vorbereitend und auch noch parallel zur Ausschreibung wurde in allen Fakultäten eine Prozessanalyse durchgeführt, die ebenfalls der Erhebung von Anforderungen bzw. dem Erkennen von Gemeinsamkeiten und Unterschieden sowie Stärken und Schwächen diente. Hierzu wurden die Fakultätsräte im Rahmen ihrer Sitzungen informiert und gebeten, geeignete Interviewpartner zu benennen. Dabei wurden die studentischen Ratsmitglieder explizit mit angesprochen und um Einbeziehung der jeweiligen Fachschaft gebeten.

3.2.3 Überwachung und Wirksamkeitskontrolle

Die Wirksamkeit der Maßnahmen wurde durch die Projektleitung im Rahmen des Stakeholder-Managements überprüft, indem das Interesse der studentischen Stakeholder nach jeder Maßnahmen-Durchführung neu bewertet wurde. Dabei konnte eine zunehmende Verbesserung hin zu einem positiven Interesse festgestellt werden. Neben der subjektiven Bewertung durch die Mitglieder der Projektleitung wurde die Verbesserung auch anhand der guten Zusammenarbeit in Projektgremien und -arbeitsgruppen begründet. Ein weiterer Indikator wurde in konstruktiven Äußerungen der studentischen Vertreter außerhalb des Projekts gesehen, bspw. in Form von Sitzungsprotokollen oder Information auf der Studentenrats-Webpräsenz sowie in Sozialen Netzen. Darin wurden vorhandene Probleme zwar benannt, aber das Projekt nicht grundlegend in Frage gestellt.

Am Ende des Vergabeverfahrens stimmten sowohl in der Beschaffungskommission als auch im Projektleitungskreis die studentischen Vertreter für die von der Projektleitung favorisierte Lösungsalternative, womit die Entscheidung unter offizieller Beteiligung der Studentenschaft getroffen wurde. Aus Sicht der Projektleitung war dies ein wesentlicher Faktor für die Akzeptanz der Entscheidung an der gesamten Universität.

4 Bewertung und Fazit

Für die Bewertung der Rolle des Stakeholder-Managements im vorliegenden Fallbeispiel werden die Kriterien Aufwand und Nutzen gegenüber gestellt.

Die wesentliche Bewertungsgröße für den Aufwand ist die entsprechende Arbeitszeit der am Stakeholder-Management beteiligten Projektmitarbeiter. Obwohl es grundsätzlich möglich gewesen wäre, den Zeitaufwand während der Projektlaufzeit zu erfassen, wurde dies nicht getan, sodass präzise Angaben nicht möglich sind. Allgemein lässt sich sagen, dass Stakeholder-Identifikation und –Analyse sowie Ableitung von Maßnahmen in eigens dafür angesetzten, mehrstündigen Terminen der Projektleitung durchgeführt wurden. Eine Abhandlung als Tagesordnungspunkt im allgemeinen Regelkommunikationstermin der Projektleitung hatte sich nicht bewährt, da die Diskussionen häufig den zeitlichen Rahmen sprengten. Mit Ausplanung und Durchführung der vorgesehenen Maßnahmen wurden nach inhaltlichen Gesichtspunkten die jeweiligen Teilprojekte beauftragt. Der zeitliche Aufwand und ggf. erforderliche Sachmittel hingen unmittelbar von Inhalt und Umfang der Maßnahme ab. Die Überwachung und Kontrolle der Wirksamkeit ließ sich schließlich wiederum gut in die wöchentliche Regelkommunikation der Projektleitung integrieren.

Auch der wahrgenommene Nutzen lässt sich nur qualitativ beschreiben und erschließt sich am ehesten durch eine Betrachtung im Zusammenhang mit Risiken und Chancen. Durch die zu Projektbeginn kritische, tendenziell negative Haltung des Studentenrats gegenüber der CMS-Einführung bestand ein hohes Risiko, dass der Studentenrat sich explizit gegen das Vorhaben stellen würde. In Kombination mit dem hohen Einfluss, den der Studentenrat generell an der betrachteten Universität hat (sowohl in den Gremien als auch im Sinne der allgemeinen Meinungsbildung), wäre dadurch die Projektdurchführung enorm erschwert, wenn nicht sogar total blockiert worden. Dies konnte durch die beschriebenen Maßnahmen des Stakeholder-Managements verhindert werden. Darüber hinaus konnte im Hinblick auf die Realisierung von Chancen erreicht werden, dass Studenten mit hohem Engagement bei der Erhebung von Anforderungen, Ausgestaltung von Prozessen und anderen Aufgaben mitarbeiteten.

Eine besondere Herausforderung im Umgang mit der Studentenschaft als Stakeholder ergab sich durch das vergleichsweise kurze Zeitfenster, in dem für einen Studenten während seiner begrenzten Studienzeit die Mitarbeit in studentischen Gremien möglich war. So wechselte der Referent für Datenschutz alle zwei bis drei Semester, was die Zusammenarbeit sehr erschwerte. Auch bei den übrigen akademischen Gremien und Ämtern stellt sich diese Problematik, da Mitgliedschaft und Amtszeit i. d. R. zeitlich befristet sind. Als Konsequenz für das Stakeholder-Management sind die Wahltermine wichtiger Gremien (bspw. Senat, Dekane) systematisch zu verfolgen, um anschließend die registrierten Stakeholder anhand der Wahlergebnisse zu überprüfen und ggf. mit Maßnahmen zu reagieren.

Ein Verzicht auf das Stakeholder-Management aufgrund des damit verbundenen Aufwands wäre mit Blick auf den Projekterfolg im vorliegenden Fall kontraproduktiv gewesen. Vielmehr konnte für das Beispiel gezeigt werden, dass das Stakeholder-Management einen wertvollen Beitrag bei der Erreichung der Projektziele leistete. Die

Besonderheiten der Organisationsform von Hochschulen lassen sich durch die Anwendung des Stakeholder-Managements systematisch erschließen, wodurch Risiken einfacher erkannt und vermieden sowie Chancen besser genutzt werden können.

Einschränkend bleibt allerdings noch einmal anzumerken, dass die beschriebenen Erkenntnisse nur aus einer frühen Projektphase stammen, nämlich der Auswahl und Beschaffung mittels Ausschreibung. Obwohl als wahrscheinlich angenommen wird, dass sich diese Erkenntnisse in der bisher noch nicht abgeschlossenen Implementierungsphase bestätigen werden, kann dies aufgrund des Projektfortschritts noch nicht als gesichert gelten.

Literatur

- [AHS13] Aapaoja, A.; Haapasalo, H.; Söderström, P.: Early Stakeholder Involvement in the Project Definition Phase: Case Renovation. In: ISRN Industrial Engineering, Jg. 2013, Artikel-ID 953915, <http://dx.doi.org/10.1155/2013/953915> (Abruf: 27.05.2014).
- [Al+11] Alpar, P.; Alt, R.; Bensberg, F.; Grob, H. L.; Weimann, P.; Winter, R.: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. Vieweg+Teubner, 6. akt. u. erw. Aufl., 2011.
- [AA13] Alt, R.; Auth, G.: Campus-Management-System. In: Kurbel, Karl et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik, 7. Aufl., Oldenbourg, 2013, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de> (Abruf: 24.02.2014).
- [BM08] Ballejos, L. C.; Montagna, J. M.: Method for stakeholder identification in interorganizational environments. In: Requirements Engineering, Jg. 13, Nr. 4, November 2008, S. 281-297.
- [Bi13] Bick, M.: Was muss die Hochschulleitung über IT wissen? CIO/IT-Governance-Modelle in deutschen Hochschulen. In: Stratmann, F. (Hrsg.): IT und Organisation in Hochschulen. Ausgewählte Beiträge einer HIS-Fachtagung (HIS: Forum Hochschule 4/2013), http://www.his-he.de/pdf/pub_fh/fh-201304.pdf (Abruf: 15.04.2014).
- [Da14] Davis, K.: Different stakeholder groups and their perceptions of project success. In: International Journal of Project Management, Jg. 32, Nr. 2, Februar 2014, S. 189-201.
- [Fr84] Freeman, R. E.: Strategic management: A stakeholder approach. Pitman, 1984.
- [Ga13] Gabriel, R.: Informationssystem. In: Kurbel, Karl et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik, 7. Aufl., Oldenbourg, 13.9.2013. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de> (Abruf: 26.05.2014).
- [Ha01] Haneke, U.: Integrierte Standardsoftware an deutschen Hochschulen: Entwicklungen, Folgen und Aussichten. In: Von Knop, J.; Haverkamp, W. (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN-Arbeitstagung über Kommunikationsnetze, Düsseldorf, 2001, S. 179-187.
- [Is14] Israel, T.: Allianz fürs Campus-Management. SoniC-Projekt der Sächsischen Fachhochschulen zur Einführung eines neuen Campus-Management-Systems macht gute Fortschritte. In: H[och]S[chule]M[it]t[W[eida]-News, 03.04.2014. <https://www.hsmittweida.de/newsampservice/hsmw-news/detailansicht-hswm-news/article/allianz-fuers-campus-management.html> (Abruf: 12.04.2014).
- [Ja+09] Janneck, M.; Adelberger, C.; Fiammingo, S.; Luka, R.: Von Eisbergen und Supertankern: Topologie eines Campus-Management-Einführungsprozesses. In: Hansen, H. R.; Karagiannis, D.; Fill, H.-G. (Hrsg.): Business Services: Konzepte, Technologien, Anwendungen. 9. Int. Tagung Wirtschaftsinformatik. Österreichische Computergesellschaft, Wien, 2009, S. 453-462.

- [Ke09] Kerzner, H.: Project Management. A systems approach to planning, scheduling, and controlling. Wiley, 10th ed., 2009.
- [Ko10] Kommission für IT-Infrastruktur der Deutschen Forschungsgemeinschaft (Hrsg.): Informationsverarbeitung an Hochschulen – Organisation, Dienste und Systeme. Empfehlungen der Kommission für IT-Infrastruktur für 2011-2015. Bonn, 2010. http://www.dfg.de/download/pdf/foerderung/programme/wgi/empfehlungen_kfr_2011_2015.pdf (Abruf: 12.04.2014).
- [Ku+11] Kuster, J.; Huber, E.; Lippmann, R.; Schmid, A.; Schneider, E.; Witschi, U.; Wüst, R.: Handbuch Projektmanagement. Springer, 3. Aufl., 2011.
- [LJA10] Littau, P.; Jujagiri, N. J.; Adlbrecht, G.: 25 years of stakeholder theory in project management literature (1984–2009). In: Project Management Journal, Jg. 41, Nr. 4, September 2010, S. 17-29.
- [ML14] Missonier, S.; Loufrani-Fedida, S.: Stakeholder analysis and engagement in projects: From stakeholder relational perspective to stakeholder relational ontology. In: International Journal of Project Management, In Press, Corrected Proof online verfügbar, <http://www.sciencedirect.com/science/article/pii/S0263786314000325> (Abruf: 27.05.2014), 2014.
- [MW13] Mormann, H.; Willjes, K.: Organisationsprojekt und Projektorganisation. Softwareeinführungsprojekte in Hochschulen aus einer organisationssoziologischen Perspektive. In: Stratmann, F. (Hrsg.): IT und Organisation in Hochschulen. Ausgewählte Beiträge einer HIS-Fachtagung (HIS: Forum Hochschule 4/2013), http://www.his.de/pdf/pub_fh/fh-201304.pdf (Abruf: 24.03.2014), S. 23-41.
- [OA14] o. A.: Homepage zur Einführung des kaufmännischen Rechnungswesens an der Universität Freiburg. <http://www.sap.uni-freiburg.de/> (27.05.2014).
- [PT07] Pacheco, C.; Tovar, E.: Stakeholder Identification as an Issue in the Improvement of Software Requirements Quality. In: Krogstie, J.; Opdahl, A.L.; Sindre, G. (Hrsg.): CAI-SE 2007, LNCS 4495, S. 370-380, 2007.
- [Pr13] Project Management Institute (Hrsg.): A Guide to the Project Management Body of Knowledge (PMBOK Guide). Selbstverlag, 5th ed., 2013.
- [SAA12] Schreiter, J.; Alt, R.; Auth, G.: Business Engineering bei der Einführung von Campus-Management-Systemen – Herausforderungen und Potenziale. In: Workshop Hochschule 2020, Informatik 2012, Braunschweig, 2012.
- [Sw13] Swain, S. N.: Stakeholder Management in SAP Implementation Projects. PMI Virtual Library: <http://www.pmi.org/> (Abruf: 27.05.2014), 2013.
- [WM11] Wieczorrek, H. W.; Mertens, P.: Management von IT-Projekten. Von der Planung zur Realisierung. Springer, 4. Aufl., 2011.
- [WO14] Wondracek, N.: Kritik am SAP Campus Management System. <http://www.brandstuve.de/wp-content/cmkritik.pdf> (Abruf: 07.05.2014)
- [ZS13] Zandhuis, A.; Stellingwerf, R.: ISO 21500 Guidance on project management. A pocket guide. Van Haren, 2013.

Agil in großen Organisationen: Eine neue Rolle im Scrum-Framework

Patrick Daut

Cassini Consulting
Cassini Consulting Nord GmbH
Johannisbollwerk 16
20459 Hamburg
patrick.daut@cassini.de

Abstract: Scrum ist zunächst definiert für einzelne Teams und ist auf dieser Ebene gut erprobt und weit verbreitet. Die Realität zeigt jedoch den Bedarf, das agile Framework auch im Konzern-Kontext zur Anwendung zu bringen. Und große Organisationen erhoffen sich die Agilität einer Kleinen. Die Anwendung von Scrum in großen Organisationen mit einer Vielzahl von Teams ist jedoch weit weniger gut beschrieben.

Nach Scrum soll die Rolle des Product Owners explizit durch eine einzige Person und nicht durch ein Komitee wahrgenommen werden. Diese Singularität erschwert eine Skalierung von Scrum, zumal der Product Owner neben der Wertmaximierung des Produkts eine Fülle weiterer Aufgaben verantwortet. Der Product Owner wird so schon früh bei der Skalierung zum Engpass; entstehende Wartezeiten sind kontraproduktiv und widersprechen der „Lean“-Idee.

Zur Lösungsfindung wird ein Vorgehen entsprechend Goldratt's Theory of Constraints [Go90] aufgegriffen. Diesem Prinzip folgend werden mit wachsender Größe der Organisation verschiedene Lösungen diskutiert, die den Engpass „Product Owner“ jeweils voll auslasten und dann beseitigen. Zur finalen Beseitigung des Engpasses wird die Verteilung der Verantwortlichkeiten auf zwei Rollen vorgeschlagen: Der Product Owner übernimmt nur noch die Verantwortung für die Anforderungspriorisierung während eine weitere Rolle, der „Story Owner“, für die Detailausgestaltung und Klärung jeweils einer Story zuständig aber nicht einem festen Team zugeordnet ist. Der Story Owner lässt sich beliebig skalieren. Dieser Vorschlag folgt in Form von Delegation und Empowerment Prinzipien aus dem „Lean“-Konzept.

1 Der Product Owner als Engpass

Scrum beruht auf der Arbeit individueller Teams. Die Zusammenarbeit auf dieser Ebene ist gut erprobt und weit verbreitet. Bei einem Einsatz von Scrum in großen Unternehmen und bei der Entwicklung eines gemeinsamen Produkts durch eine große Anzahl von Teams werden wir in der Praxis aber zunächst allein gelassen: Literatur wie auch Erfahrungsberichte haben bislang keine Patentrezepte zur Skalierung von Scrum über die Teamebene hinaus geliefert. Die Realität zeigt jedoch den Bedarf, das agile Framework auch im Konzern-Kontext zur Anwendung zu bringen. Und große Organisationen erhoffen sich die Agilität einer Kleinen.

Die Rolle des Product Owners soll nach Scrum explizit durch eine einzige Person und nicht durch ein Komitee wahrgenommen werden, da sie über die Prioritäten entscheidet, in denen Anforderungen umgesetzt werden. So wird das Produkt definiert und dessen Wertmaximierung zentral verantwortet. Darüber hinaus hat der Product Owner weitere Aufgaben, wie die Detailabstimmung von Anforderungen und der Ansprechpartner für die Entwicklungsteams und Stakeholder zu sein. Wird Scrum in einer großen Organisation eingesetzt, zeigt sich ein Engpass in der Singularität des Product Owners und damit in der Beziehung zum Kunden: Durch den Einsatz mehrerer Teams kann die Kapazität der Organisation erhöht werden aber der Product Owner wird zum Engpass.

Im Fokus der Betrachtung steht hier nicht ein einzelnes Entwicklungsprojekt sondern eine Organisation oder ein Teil einer Organisation, die zu dem Zweck geschaffen ist, kontinuierlich neue Produkte zu entwickeln oder bestehende Produkte weiterzuentwickeln. Die Argumentation ist aber für Projekte ab einer gewissen Größenordnung ebenso gültig.

2 Ein Blick über den Tellerrand

Wie können wir uns methodisch diesem Problem nähern? Die Theory of Constraints, die auf den Arbeiten von Goldratt beruht, floss in Methoden ein, die in erster Linie in Bereichen wie Produktion und Logistik Anwendung finden. Im Kern besagt sie [Go90]:

- (A) Ein Engpass ist alles, was das System von der Erreichung einer höheren Leistung – im Sinne des Durchsatzes – abhält.
- (B) Jedes System muss über mindestens einen solchen Engpass verfügen.

Weiter wird im Rahmen der Theory of Constraints ein Vorgehen vorgeschlagen, wie in Abbildung 1 dargestellt.

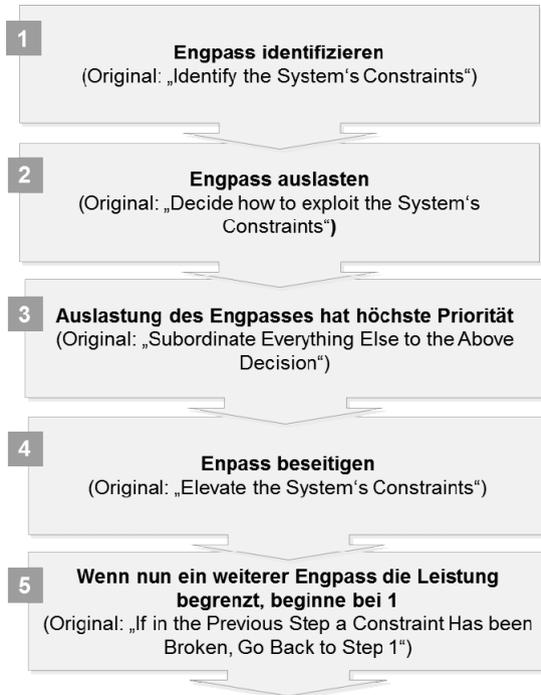


Abbildung 1: Vorgehen der Theory of Constraints [Go90]

Dahinter steht der folgende Gedanke: Ist ein System soweit ausgelastet, dass der Engpass den Durchsatz begrenzt, so bringt das Hinzufügen von Ressourcen an anderer Stelle keinen weiteren Nutzen bis der Engpass aufgehoben wurde. Im Fokus liegt also, zunächst alles daran zu setzen, den Engpass auszulasten und ihn im Anschluss zu beseitigen.

3 Die Theory of Constraints und die Skalierung von Scrum

Übertragen wir nun diese Methode auf die Skalierung von Scrum in großen Organisationen.

Scrum ist zunächst nur für ein Team definiert. Ein ideales Scrum Team sollte nicht mehr als 7-9 Mitglieder umfassen. Mit dem Begriff „Skalierung“ im Rahmen dieses Beitrags einer der beiden folgenden Fälle gemeint:

(A) Die Einführung von Scrum als Arbeitsmodell in einer Entwicklungsorganisation, die mehr Mitarbeiter umfasst als in einem Scrum Team organisiert werden können.

(B) Das Wachstum einer nach Scrum arbeitenden Organisation über die ideale Größe eines Teams hinaus.

Die Skalierung kann durch die Aufteilung von Arbeitsaufträgen bzw. Kundenanfragen auf mehrere Scrum Teams erreicht werden: Wächst eine Organisation (Fall B) oder in einer bestehenden Organisation wird Scrum schrittweise eingeführt (Fall A), wird im Zuge dessen die Anzahl an Teams erhöht, um den benötigten Durchsatz erreichen zu können. Entsprechend der Theory of Constraints können immer weitere Scrum Teams hinzugefügt werden, bis der erste Engpass auftritt. Erkennbar ist diese Situation daran, dass weitere Ressourcen in Form eines zusätzlichen Teams keine oder nur noch eine geringe Steigerung des Durchsatzes in der Entwicklungsorganisation bewirken können. Der Engpass ist ausgelastet.

4 Die Organisation wächst

4.1 Der Anfang: Warum wird der Product Owner zum Engpass?

Gehen wir somit im Falle einer Skalierung davon aus, dass mehrere Teams existieren. Weiter wird angenommen, dass diese an einem einzigen gemeinsamen Produkt arbeiten und somit Koordinationsbedarf zwischen den Teams besteht. Zunächst belassen wir es auch bei steigender Anzahl von Teams bei einem einzigen Product Owner, den sich alle Teams teilen. Damit ist eine grundlegende Anforderung von Scrum erfüllt: Es ist explizit vorgesehen, dass die Rolle des Product Owner durch eine Person ausgefüllt wird, nicht durch ein Komitee oder eine Gruppe. Dies liegt darin begründet, dass der Product Owner inhaltlich für das Produkt verantwortlich sein und verantwortlich gemacht werden soll. Dieser darf als einziger Anforderungen priorisieren und übernimmt damit die Verantwortung für die Ausgestaltung des Produkts zu einem bestimmten Zeitpunkt. Stakeholder nehmen lediglich indirekt darauf Einfluss.

Entsprechend der Theory of Constraints kann es sinnvoll sein, allen Teams einen einzigen Product Owner zuzuweisen bis dieser zum einschränkenden Faktor wird. Je nach Komplexität der Arbeitsaufgaben und des zu entwickelnden Produkts kann dieser Fall bereits bei zwei oder auch erst mehr Teams eintreten. Solange es nicht nötig ist, muss die oben beschriebene Singularität des Product Owners nicht aufgelöst werden. Wichtig ist, dass der Product Owner von allen Aufgaben befreit wird, die nicht zu seiner Rolle gehören und alle nötige Bedingungen geschaffen werden, die ihm als Engpass möglichst effizientes Arbeiten ermöglichen. Diesem muss höchste Priorität zugeordnet werden entsprechend Schritt 3 des Vorgehens der Theory of Constraints [Go90].¹ Es ist allerdings unwahrscheinlich, dass eine Person mehr als zwei bis drei Teams als Product Owner dienen kann, ohne die Arbeit dieser Teams zu bremsen.

Die Erfahrung zeigt, dass – wie in Abschnitt 1 postuliert – dieser erste Engpass oft an der Rolle des Product Owners auftritt. Wie kommt es dazu? Der Product Owner verantwortet einerseits die Priorisierung von Anforderungen, andererseits ist es von großer

¹ S. auch Abbildung 1

Bedeutung, dass er während der Detaillierung von Anforderungen bzw. während deren Umsetzung dem Team als Ansprechpartner und ggf. als zentraler Kanal zum Kunden zur Verfügung steht. Die ständige Verfügbarkeit des Product Owners in seiner Funktion als „on-site customer“ ist für den Erfolg agilen Vorgehens von großer Bedeutung, da Kommunikation nach Bedarf einer detaillierten Spezifikation im Vorfeld vorgezogen wird.² Mit steigender Anzahl von Teams wird der Product Owner überlastet und bremst die Arbeit der Teams aus. Darüber hinaus kommen mit steigender Organisations- und Produktgröße weitere Stakeholder, darunter verschiedene Managementpositionen, unternehmensweite Querschnittsfunktionen wie Security und Compliance oder Investoren, weitere Kunden, Dienstleister, Partner und Regulierungsinstanzen hinzu. Für all diese – verschiedenen – Aufgaben sieht Scrum die Rolle des Product Owners vor.

Eine Überlastung des Product Owners führt dazu, dass Entwicklungsteams bei Design und Implementierung auf den Product Owner warten müssen: Der Product Owner ist ein ausgelasteter Engpass und auch durch weitere Teams kann der Durchsatz der Organisation kaum noch erhöht werden. Weiterhin versucht ein Team, die entstehende Wartezeit anderweitig zu nutzen um seine Auslastung zu erhöhen. Es erhöht damit aber auch seine „Work in Progress“ und durch häufige Fokuswechsel der Entwickler entsteht eine Verschwendung im Sinne der *Lean*-Idee, also Aufwand und Zeit, die nicht zur Schaffung von Wert beitragen [Oh88].³

4.2 Eine Lösung für kleinere bis mittelgroße Organisationen

Um den Engpass aufzulösen wird in diesem Beitrag vorgeschlagen, in einem ersten Schritt mehrere Product Owner – jeweils einen pro Team – einzuführen. Durch die Abhängigkeiten der Arbeit der einzelnen Teams entsteht Koordinationsbedarf unter den Teams bzw. ihren dedizierten Product Ownern. Ein übergeordneter Chief Product Owner verantwortet das Gesamtprodukt, zerlegt vom Kunden beauftragte Features⁴ in kleinere Teile und verteilt sie an die Product Owner der Teams. Diese Organisationsstruktur deckt den entstandenen Koordinationsbedarf. Der Chief Product Owner bildet – im Rahmen seiner Gesamtverantwortung – auch nach der Umsetzung in den Teams und Abnahme durch die Product Owner wieder die schließende Klammer. Eine solche Struktur ist in der Praxis häufig anzutreffen. Erfahrungsgemäß kann sie gut funktionieren für eine Teamanzahl bis etwa sechs, sieben oder acht Teams je nach Komplexität der Aufgaben. Doch tut sie das auch für unbegrenzt viele?

Die geschaffene Organisationsstruktur stellt die Konsistenz des Gesamtprodukts sicher, aber mit steigender Anzahl an Teams muss sich irgendwann wiederum ein Engpass an der Stelle des Chief Product Owners bilden: Er kann nicht unbegrenzt viele Product Owner koordinieren. Es zeigt sich erneut ein Engpass gestoßen und dessen Auslastung

² Die Relevanz eines „on-site customers“ wird schon mit dem Konzept des eXtreme Programming beschrieben, s. dazu [Ex01, BT04].

³ [PP07], Kapitel 2.2, und [Re09], z.B. S. 32 ff, übertragen dies auf die Software- bzw. Produktentwicklung und konkretisieren.

⁴ Ein „Feature“ wird hier definiert als eine Gruppe von User Stories, deren Nutzen gemeinsam höher ist als Summe des Nutzens der einzelnen Stories. Es handelt sich um eine übergeordnete Ebene, in der betriebswirtschaftliche Betrachtungen, beispielsweise des ROIs, im Fokus stehen.

muss zur höchsten Priorität erhoben werden. Eine Möglichkeit dazu ist die Erweiterung der entstandenen Hierarchie um eine – oder bei ansteigender Anzahl an Teams um mehrere Ebenen. Es entsteht eine hierarchische Struktur wie in Abbildung 2 dargestellt. Die Leitungsspanne des Chief Product Owners wird so verkleinert.

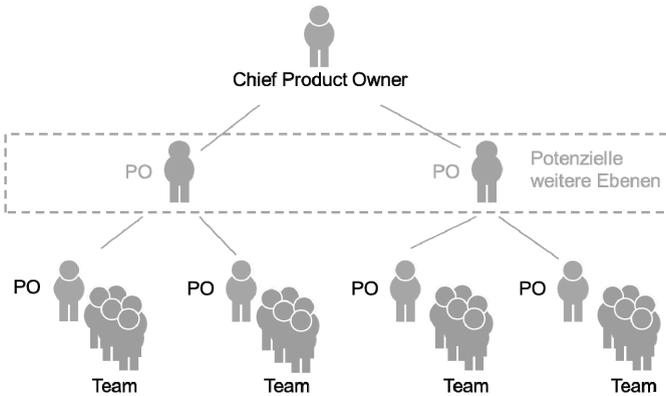


Abbildung 2: Wachsende hierarchische Product Owner-Struktur

Spätestens hier wird deutlich, dass dieses Modell seinerseits auf Grenzen stößt. Abbildung 3 zeigt, dass Entscheidungswege letztlich den Chief Product Owner aufgrund seiner Gesamtverantwortung mit einbeziehen. Der Koordinationsaufwand ist hoch und steigt mit weiteren Ebenen noch stark an. Die Product Owner warten auf Entscheidungen, was in der Begrifflichkeit der Theory of Constraints den Durchsatz irgendwann begrenzen muss. Auch hier bedeuten Wartezeiten Verschwendung in einem *Lean*-Sinne.

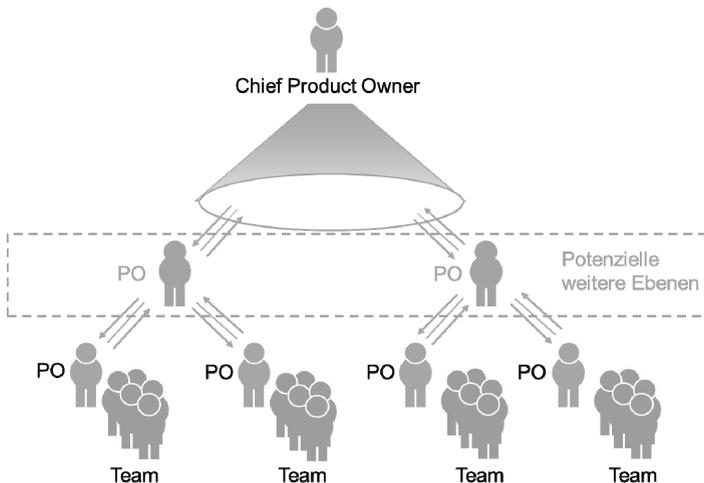


Abbildung 3: Steigender Koordinationsaufwand in einer wachsenden hierarchischen Struktur

Des Weiteren nimmt eine solche Struktur die Vorteile agilen Vorgehens wenn sie zu komplex wird: Je mehr Personen involviert sind, desto weniger kann ein Product Owner die Verantwortung und Aufgaben wahrnehmen, die ihm im Scrum-Framework zuge-dacht werden. Er hat nur noch sehr eingeschränkte Verantwortung für die Wertmaximierung in seinem Bereich, vieles ist bereits in der Hierarchie über ihm entschieden. Es besteht die Gefahr, dass zur Verteilung von Arbeitspaketen auf Product Owner und Teams vermehrt vorgelagerter Analyseaufwand ohne Beteiligung der Teams entsteht. Entgegen der agilen Idee wird „Big upfront design“ betrieben, dessen Wert immer stärker verfällt während es auf die Umsetzung wartet.⁵

Diese Faktoren zusammen bewirken, dass ab einer gewissen Anzahl von Teams weitere Ressourcen kaum zusätzlichen Durchsatz bewirken können. Wiederum ist der Product Owner – genauer: die Hierarchie von Chief Product Owner und Product Owner – ein ausgelasteter Engpass.

4.3 Mittlere und große Organisationen: Skalierung durch eine neue Rolle

Das bisherige Vorgehen bei der Skalierung scheint an diesem Punkt gänzlich ausgereizt. Welche Möglichkeiten haben wir? Schauen wir uns dazu die Product Owner-Rolle noch einmal genauer an. Die Rollendefinition umfasst grob die folgenden Verantwortlichkeiten:

- Priorisierung der Anforderungen
- Unterstützung bei der Detaillierung von Anforderungen, Repräsentation des Kunden, Klärung von Detailfragen
- Unterstützung bei Konzeption und Design
- Durchführung der Akzeptanztests
- Stakeholdermanagement

Die in Abschnitt 4.1 beschriebene Singularität der Product Owner-Rolle in Verbindung mit seinen breiten Verantwortlichkeiten ist der Grund für die Schwierigkeit, diese Rolle zu skalieren. Die Hypothese ist nun, mit der Verteilung der Aufgaben auf mehrere Rollen eine besser skalierende Struktur zu erreichen. Eine klassische funktionale Trennung auf mehrere Rollen, entsprechend der Funktionen Business Analyse, Architektur, Konzeption und Abnahme, erlaubt die Skalierung durch Spezialisierung. Jedoch ist ein solches Vorgehen nicht mehr agil: Es würde beispielsweise klassisches Requirements Engineering betrieben und Konzepte weit im Voraus und stark getrennt von der Umsetzung erstellt. Wir sind erneut beim oben erwähnten „Big upfront design“. Unter der Voraussetzung einer veränderlichen Umwelt und veränderlicher, komplexer Anforderungen

⁵ Reinertsen beispielsweise spricht von den durch Wertverfall entstehenden Kosten als „Holding Costs“ und nennt sie als einen Parameter für die Bestimmung der optimalen „Batch Size“, s. [Re09], 35f.

verfällt der Wert der Konzepte während sie auf ihre Umsetzung warten und stellen Verschwendung im *Lean*-Sinne dar.

Bleiben wir bei der Lösungssuche bei den *Lean*-Prinzipien: Ein zentraler Aspekt ist ein Empowerment der Basis [LeMa, LePr] und mehr Verantwortung soweit entlang einer Hierarchie herunter zu delegieren, bis genug Wissen für eine Entscheidung vorhanden ist. Eine solche Dezentralisierung und Delegation von Entscheidungen soweit abwärts wie möglich in einer Hierarchie ermöglicht die benötigte Skalierbarkeit. Eine grundsätzliche Empfehlung ist, Entscheidungen, die in unregelmäßigen Abständen zu treffen sind, große Auswirkungen nach sich ziehen oder signifikante Economies of Scale haben, zu zentralisieren und dagegen schnell zu treffende Entscheidungen, deren Effekte einem Wertverfall im Laufe der Zeit unterliegen, zu dezentralisieren [Re09]. Dementsprechend wird in diesem Beitrag vorgeschlagen, neben dem Product Owner eine weitere Rolle einzuführen, die die Verantwortlichkeiten übernimmt, die dezentralisiert werden können. Ein einziger Product Owner ist weiter verantwortlich für die Anforderungspriorisierung und bestimmt damit über das Produkt. Davon getrennt und in einer zweiten Rolle, dem „Story Owner“, werden die folgenden Aufgaben zusammengefasst:

- Die Detailabstimmung mit dem Kunden zu einer Story,
- deren Ausarbeitung zusammen mit dem Entwicklungsteam und die
- Funktion der Schnittstelle zum Kunden bei Rückfragen auf Story-Ebene.

Abbildung 4 zeigt den entstehenden Schnitt durch die Funktionen.

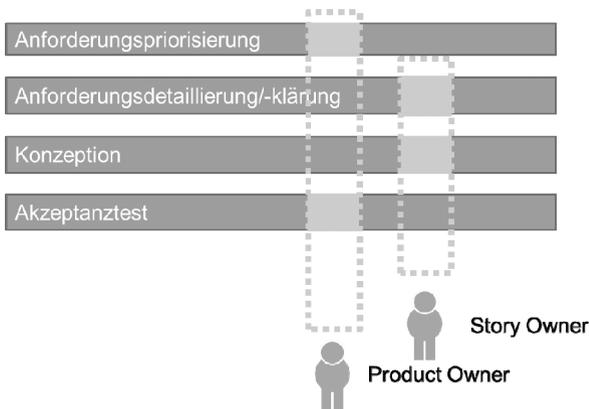


Abbildung 4: Vertikaler Schnitt in Rollen

Es ergibt sich eine Struktur wie in Abbildung 5 dargestellt: Mehrere Story Owner – jeweils einer pro Story untersteht einem zentralen Product Owner.

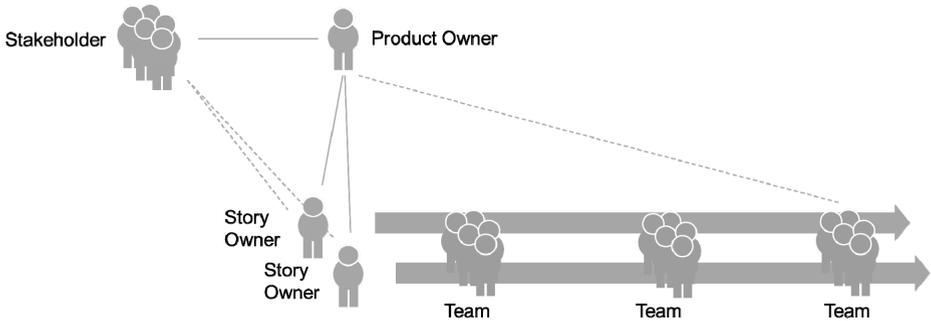


Abbildung 5: Story Owner-Struktur

Ein Story Owner ist nicht einem Team sondern einer Story dediziert zugeordnet. Er betreut diese möglicherweise über mehrere an der Umsetzung beteiligte Teams hinweg.⁶ Erhöht sich die Anzahl an Teams, zieht dies also nicht notwendigerweise nach sich, dass mehr Story Owner benötigt werden. Andererseits können mit steigenden Kundenanforderungen flexibel weitere Story Owner hinzugefügt werden. Für jedes Feature und jede Story existiert mit dem Story Owner eine Person mit fachlichem Verständnis des Themas sowie der ungeteilten Verantwortung dafür, das Thema voranzubringen und mit den Teams zusammen im Detail auszugestalten. Er „repräsentiert“ eine Story, ist die Schnittstelle zwischen Teams und Kunden in Bezug auf sein Thema und koordiniert sich eigenverantwortlich mit anderen Story Ownern im Fall von Schnittmengen.

Mit der Anzahl der konkreten Stellen, die die Story Owner-Rolle ausfüllen, begrenzt man den Work in Progress in der Struktur. Durch die Möglichkeit, nach Bedarf Story Owner hinzuzufügen, ist der Engpass durch die Kapazität des Product Owners bei der Skalierung aufgehoben.

Warum bleibt diese Struktur trotzdem schlank und agil? Die Herleitung der Aufgabentrennung in Product Owner und Story Owner basiert auf der Lean-Idee und dem Gedanken der Empowerment der Basis – in diesem Fall der Story Owner. Es werden genau die Entscheidungen an die Story Owner delegiert, die sie mit ihrer Nähe zur Umsetzung und zu der Umsetzung in den Teams am besten treffen können. Abbildung 6 verdeutlicht dies.

⁶ Mit „Story“ ist hier nicht notwendigerweise eine User Story gemeint, es kann sich auch um ein größeres zusammenhängendes Feature handeln.

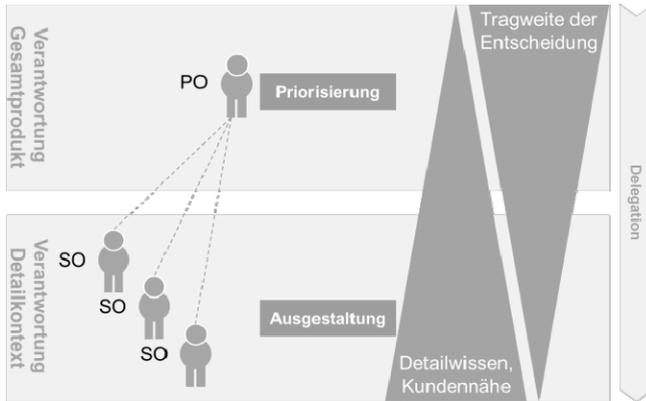


Abbildung 6: Verantwortungsbereiche von Product Owner und Story Owner

Eine Entscheidung oder Freigabe durch den Product Owner ist nicht notwendig. Die Story Owner stimmen sich auf ihrer Ebene direkt miteinander ab und arbeiten eng mit den Teams zusammen.

Diese beschriebene Struktur wird hier für eine Entwicklungsorganisation mit bis zu – in Abhängigkeit von der Produktkomplexität – mehreren Dutzend Teams vorgeschlagen. Bei hoher Komplexität oder noch größeren Organisationen kann ein Framework wie Leffingwell's *Scaled Agile Framework* [SAFe (für einen Überblick), Le07, Le11] Lösungsideen liefern.

5 Verschiedene Lösungen – je nach Organisationsgröße

In diesem Beitrag wird gedanklich das von der Theory of Constraints vorgeschlagene Vorgehen auf die Skalierung einer nach Scrum arbeitenden Entwicklungsorganisation angewandt. Entlang dieses Vorgehens wird argumentiert, dass der aufgezeigte Engpass an der Rolle des Product Owners auf verschiedene Weise gelöst werden kann. Je nach Größe der Entwicklungsorganisation genügt eine andere Struktur damit diese Rolle nicht den Durchsatz in der Organisation limitiert.

Ab einer gewissen Größe der Organisation aber verlangt die weitere Skalierbarkeit nach der Aufteilung der Verantwortlichkeiten des Product Owners auf zwei Rollen und damit der Einführung der Story Owner-Rolle, wodurch eine weiterhin schlanke und agile Struktur entsteht. Es stellt sich die Frage, warum das Story Owner-Konzept nicht in jedem Fall – schon bei kleinen Organisationen – eingesetzt werden sollte. Auch wenn diese Struktur die flexibelste der hier beschriebenen ist, zeigt der im Beitrag verfolgte Gedanke der Theory of Constraints dass ihre Einführung schlicht nicht nötig ist solange der Engpass auf eine der anderen beschriebenen Weisen noch nicht voll ausgelastet ist. Denn neben seiner Vorteile in der Skalierbarkeit birgt der Ansatz auch eine gewisse Komplexität.

Schlussendlich ist der durch die Product Owner-Rolle bei der Arbeit nach Scrum in großen Organisationen induzierte Engpass aufgelöst. Folgen wir der Theory of Constraints, so wird eine weitere Skalierung irgendwann wohl erneut durch einen Engpass limitiert; wahrscheinlich an ganz anderer Stelle. Die Theory of Constraints gibt uns die generischen Hinweise, wie wir auch damit umgehen können.

Literaturverzeichnis

- [ExPr] <http://www.extremeprogramming.org/rules/customer.html>. Letzter Zugriff: 20.08.2014.
- [BT04] Boehm, Barry W.; Turner, Richard: *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, 2004.
- [Go90] Goldratt, Eliyahu M.: *Theory of Constraints*. North River Press, 1. Aufl., 1990.
- [Le07] Leffingwell, Dean: *Scaling Software Agility – Best Practices for Large Enterprises*. Addison Wesley, Pearson Education, 2007.
- [Le11] Leffingwell, Dean: *Agile Software Requirements – Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison Wesley, Pearson Education, 2011.
- [LeMa] http://de.wikipedia.org/wiki/Lean_Management. Letzter Zugriff: 20.08.2014.
- [LePr] http://de.wikipedia.org/wiki/Lean_Production. Zugriff: 20.08.2014.
- [Oh88] Ohno, Taiichi: *Toyota Production System: Beyond Large-scale Production*. Productivity Press, 1988.
- [PP07] Poppendieck, Mary B.; Poppendieck, Tom: *Implementing Lean Software Development: From Concept to Cash*. Addison Wesley, 2007.
- [Re09] Reinertsen, Donald G.: *The Principles of Product Development Flow – Second Generation Lean Product Development*. Celeritas Publishing, 2009.
- [SAFe] <http://scaledagileframework.com>. Letzter Zugriff: 20.08.2014.

Teil III

Eingeladene Beiträge der Session „Future Track“

Formale Systemmodelle der Informatik als Basis für eine bessere Beherrschung von KPM-Szenarien

Jan Friedrich

4Soft GmbH
Mittererstr. 3
80336 München
friedrich@4soft.de

Abstract: Kollaborativ gemanagte Projekte (KPM-Szenarien) werden noch nicht ausreichend durch Werkzeuge und Vorgehensmodelle unterstützt. Ursache sind u.a. eine fehlende theoretische Durchdringung der PM-Domäne und die Fokussierung der SW-Prozessforschung auf Prozessautomatisierung. Die guten Erfahrungen mit der Übertragung formaler Systemmodelle auf Prozesse sollten herangezogen werden, um auch Planung und Kontrolle auf diese Weise formal-theoretisch zu durchdringen. Der vorliegende Impulsbeitrag führt in die oben formulierten Thesen ein, belegt sie und leitet in die Diskussion zum Thema über.

1 Einführung und Thesen

Kollaboratives Projektmanagement (KPM) dient zur Planung, Kontrolle und Steuerung¹ großer, komplexer und sowohl räumlich als auch organisatorisch verteilter Projekte und gewinnt zunehmend an Bedeutung [CN+03]. Statt eines einzelnen zentralen, alles überblickenden Projektmanagers existiert eine Hierarchie (bzw. ein Netz) von miteinander interagierenden Managern, die jeweils einzelne Projektteile mit eigenen Projektteams planen, kontrollieren und steuern und dabei mit den anderen Managern interagieren und kooperieren. Diese Definition macht deutlich, dass KPM Überschneidungen und Ähnlichkeiten mit Programm- und Projektportfoliomanagement, *Cross Company Planning* (CCP) und *Network-Centric Planning* (NCP) besitzt.

Zusätzlich zu den „normalen“ Herausforderungen eines Projekts bestehen in KPM-Szenarien meist folgende Schwierigkeiten: 1.) *Größe und Komplexität* erfordern eine wohlgedachte Projektstrukturierung und Schnittstellenmanagement. Außerdem muss ein adäquater Informationsfluss sichergestellt werden, sodass jeder Manager die notwendigen Informationen erhält, aber trotzdem nicht in Informationen „ertrinkt“. 2.) Die einzelnen Teams stammen oft aus unterschiedlichen Organisationen und befolgen *unter-*

¹ Die genannten Begriffe werden im Rahmen dieses Beitrags als deutsche Äquivalente der entsprechenden Schritte im PDCA-Zyklus verwendet: Planung (PLAN), Kontrolle = Berichtswesen und Soll/Ist-Vergleich (CHECK) und Steuerungsentscheidungen (ACT).

schiedliche Prozess- und Vorgehensmodelle, die oft nicht kompatibel zueinander sind und im Projekt an den Schnittstellen ad-hoc miteinander abgeglichen werden müssen. Außerdem werden klassische und agile PM-Techniken oft im selben Projekt kombiniert. 3.) Die im Projekt miteinander kooperierenden Organisationen sind an anderer Stelle *Konkurrenten* oder stammen aus *unterschiedlichen Bündnissystemen* (z.B. deutsch-chinesische Kooperationsprojekte). Informationssicherheitsaspekte müssen deswegen auch im Projektmanagement beachtet werden, da beispielsweise Pläne auch Ingenieurs- oder Prozesswissen enthalten können [DS02]. Dabei existiert das Dilemma: Zu wenig Informationsweitergabe gefährdet den Projekterfolg, zu viel Informationsweitergabe schwächt ggf. die eigene Position gegenüber dem Konkurrenten.

Es existieren mannigfaltige Projekterfolgskriterien, die natürlich auch in KPM-Szenarien gelten. Im Folgenden sind lediglich drei ausgewählte näher beschrieben, weil sie in KPM-Szenarien schwer (bzw. schwerer) erreichbar sind: 1.) *Vorgehensmodelle* können ein Projekt positiv beeinflussen, wenngleich der Erfolg nicht immer statistisch signifikant ist [BE+10]. Wie bereits eingangs erwähnt, finden sich in KPM-Szenarien aber oft mehrere Vorgehensmodelle, die ggf. inkonsistent zueinander sind. 2.) Die *Kontrollfrequenz* (also die Häufigkeit des Soll/Ist-Vergleichs) beeinflusst den Projekterfolg positiv [BE+10]. Bei größeren Projekten braucht es aber Zeit, entsprechende Berichte in einer Projekthierarchie „von unten nach oben“ (bzw. in einem Netzwerk „von links nach rechts“) weiterzuleiten, zu filtern und zu verdichten, insbesondere wenn dies manuell erfolgt. Je größer ein Projekt ist, desto kleiner kann in der Regel die Kontrollfrequenz sein. 3.) Die *Planqualität* hat ebenfalls einen positiven Einfluss auf den Projekterfolg, wie Dvir und Lechler in [DL04] belegen. Sie zeigen außerdem, dass die wahre Herausforderung nicht darin besteht, zu Beginn des Projekts einen „guten“ Plan zu besitzen, sondern diesen Plan über die Projektlaufzeit „gut“ zu erhalten. In KPM-Szenarien gibt es aber meist viele Planer, die nicht am selben Standort sind, unabhängig voneinander umplanen und sich darüber hinaus ihre jeweiligen Pläne nicht offenlegen. Inkonsistenzen sind vorprogrammiert, die Planqualität leidet.

Folgende zentrale Thesen bilden die Grundlage dieses Beitrags:

- 1.) Das theoretische Fundament des Projektmanagements ist – im Gegensatz zur praktischen Erfahrung in dem Bereich – mangelhaft ausgeprägt. Dies wirkt sich gerade in KPM-Szenarien negativ aus.
- 2.) Die Übertragung formaler Systemmodelle auf Software-Prozesse hat sich bewährt und wertvolle wissenschaftliche Beiträge hervorgebracht. Allerdings führte der Schwerpunkt der Prozessautomatisierung dazu, dass in der Praxis nicht Projekte, sondern insbesondere Geschäftsprozesse profitieren konnten.
- 3.) Formale Systemmodelle der Informatik lassen sich gewinnbringend auf KPM-Szenarien adaptieren. Dabei sollte vom Ziel der Planautomatisierung Abstand genommen werden. Vielmehr sollte der Schwerpunkt auf die Planung an sich (z.B. auf Qualitätsaspekte von Plänen) und die Unterstützung des Soll/Ist-Vergleichs gelegt werden.

2 Argumente und Belege

Die folgenden Unterkapitel liefern Argumente und Belege für die genannten Thesen. Jedes Unterkapitel ist dabei genau einer These zugeordnet.

2.1 Theoretisches PM-Fundament (These 1)

Koskela und Howell appellieren mit ihrem Beitrag [KH02] an die „PM-Gemeinde“, die theoretischen Grundlagen des Projektmanagements besser zu erforschen. Sie argumentieren, dass spätestens durch das Aufkommen agiler PM-Ansätze die PM-Theorie überholt ist, wenn sie überhaupt jemals in adäquater Form existiert hat. Sie belegen, dass das mangelhafte theoretische Fundament insbesondere bei großen, komplexen Projekten mit vielen Änderungen zum Problem wird. Unter anderen weisen sie auf die Planungsproblematik hin: Pläne werden zu Projektbeginn einmal erstellt und dann nicht weiter angepasst und aktuell gehalten – wenngleich sie deutlich machen, dass Pläne in der Praxis niemals 100%ig aktuell und konsistent sein können.

Alberts und Hayes gehen im Kontext militärischer Planung auf Unterschiede zwischen dem klassischen Planungsansatz mit einem alles überblickenden Planer und dem Network-centric Planning (NCP) mit vielen verteilten Planern ein [AH07]. Die Analyse und die Schlussfolgerungen können in weiten Teilen direkt auf klassisches PM und KPM übertragen werden. Die Autoren unterscheiden zwischen Planungsqualität und Planqualität und bemängeln ebenfalls eine mangelnde theoretische Durchdringung der Domäne. Zudem regen sie weitere Forschung beispielsweise zur Ableitung konkreter Qualitätsmetriken für Pläne und Planung an, um das NCP-Feld weiter zu entwickeln.

Auf Werkzeugseite sind insbesondere KPM-Software und Projektcockpits zu nennen. An diesen lässt sich das ausbaubedürftige theoretische PM-Fundament ebenfalls erkennen. Die Software RPlan unterstützt explizit KPM/CCP-Szenarien und ist in der deutschen Automobilindustrie weit verbreitet. Sie unterstützt die Konsistenthaltung von Plandaten zwischen verschiedenen Planern. Auf der Kontrollseite sieht das Werkzeug ein Ampelsystem vor. Ein Automatismus für das „Rechnen“ mit Projektampeln war in früheren Versionen vorhanden, scheiterte jedoch an mangelnder theoretischer Grundlage: Beispielsweise müsste eine solche Theorie eine Aussage darüber erlauben, ob und ggf. wie weit sich eine rote Ampel auf unterster Ebene nach oben durchschlägt.

Münch und Heidrich beschreiben in [MH04] den Stand in Praxis und Wissenschaft zum Thema Projektcockpits. Sie weisen auf grundsätzliche Defizite bei der Verknüpfung zwischen der Planung und dem Soll/Ist-Vergleich (Kontrolle) in einem Projektcockpit hin. Die in [RB+10] gezeigte Cockpit-Implementierung zeigt außerdem Probleme bei der Beherrschung der Informationssicherheit und damit bei der Anwendung in KPM-Szenaren: Repositorydaten aus dem gesamten Projekt werden zunächst in einem großen zentralen „Datentopf“ gesammelt, bevor daraus die Statusinformationen für die einzelnen Verantwortlichen abgeleitet werden.

2.2 Forschung zu SW-Prozessen und Vorgehensmodellen (These 2)

Seit den 1980er Jahren befasst sich das Software Engineering intensiv mit der Modellierung von Software-(Entwicklungs-)Prozessen. Richtungsweisend war Osterweils Idee des *Process Programmings*: Er votierte dafür, die Erkenntnisse über Software-Programme auf den SW-Entwicklungsprozess selbst zu übertragen [Ost87]. In Folge entstanden viele wertvolle Beiträge (z.B. FUNSOFT Netze [EG91] oder Little-JIL [WC+11]), die aber zumeist das direkte Enactment von Vorgehensmodellen adressieren, sie also davon ausgehen, dass sich die Handlungen eines Mitarbeiters direkt (und idealerweise werkzeuggestützt) aus dem Prozess-/Vorgehensmodell ableiten lassen. Profitiert von den Ergebnissen haben insbesondere wiederholbare Prozesse (z.B. Geschäftsprozesse), die durch Workflow-Maschinen unterstützt werden können [FB11].

Beim planbasierten Enactment wird aus dem Vorgehensmodell zunächst ein Plan abgeleitet, aus dem sich dann die Handlungen der Mitarbeiter ableiten [FB11]. Diese Enactment-Variante ist prinzipiell kompatibel zum Projektmanagement, insbesondere wenn davon ausgegangen wird, dass der abgeleitete Plan ständig aktualisiert und an den tatsächlichen Projektverlauf angepasst werden muss. Obwohl in diese Richtung geforscht wurde (z.B. [BL+95] [Fis10] [Gna05]), genoss diese Variante weitaus weniger Aufmerksamkeit. Manche Beiträge gehen sogar so weit, sie nicht als „richtiges Enactment“ zu zählen, weil sie nicht vollständig automatisiert abläuft und projektspezifische Informationen manuell hinzugefügt werden müssen [BC+07].

Auch Mischformen existieren, z.B. MILOS [MD+00]: Der Ansatz geht prinzipiell von einem Plan aus, der aus dem Vorgehensmodell abgeleitet werden muss. Allerdings wird hier der Plan als automatisierbare Arbeitsanweisung aufgefasst: Ein Planer erstellt den Plan und verlässt sich dann auf eine Workflow-Maschine, die das Projekt für ihn steuert und die Mitarbeiter „antriggert“. Umplanungen sind vorgesehen; Soll/Ist-Vergleiche um Planabweichungen zu bestimmen, kommen in dem Konzept aber nicht vor.

Insgesamt ist in der Domäne ein Hang zur Prozess-/Planautomatisierung zu beobachten, die die Anwendbarkeit in KPM-Szenarien erschwert: Das angestrebte Ideal ist oft ein Unterstützungssystem, das die Mitarbeiterkoordination (z.B. die Zuweisung von ToDos) vornimmt. Die im PM so wichtigen Faktoren wie gemeinsame Planung, Vertrauen auf die Mitarbeiter (die ggf. auch bewusst vom Plan abweichen, weil ihnen eine bessere Lösung einfällt), Soll/Ist-Vergleiche, etc. werden meist nicht einbezogen.

2.3 Adaption formaler Systemmodelle (These 3)

Prinzipiell ist die Idee Osterweils dennoch richtig: Die Übertragung informatischer Konzepte auf Entwicklungsprojekte kann einen wichtigen Beitrag leisten, diese besser zu verstehen und mit Werkzeugen zu unterstützen. Betrachtet man KPM-Szenarien, dann bieten sich insbesondere komponentenbasierte Systemmodelle (z.B. [BS01]) als Basis für eine Adaption im Rahmen einer KPM-Theorie an, weil sie Prinzipien wie Abstraktion, Modularisierung, (De-)Komposition, Skalierbarkeit, Information Hiding und Schnittstellen von Haus aus bieten. Aus der Einführung dieses Beitrags ist ersichtlich, dass

diese Prinzipien auch in KPM-Szenarien anzutreffen sind, beispielsweise beim Schnittstellenmanagement oder bei Informationssicherheitsanforderungen.

Gemäß [Sta73] ist bei jedweder Modellierung das pragmatische Merkmal zu beachten. Hier sollte die Erkenntnis stehen, dass kein Plan perfekt sein kann und Planabweichungen oft unvermeidlich und manchmal sogar sinnvoll sind. Tritt man deshalb vom Automatisierungsanspruch zurück und betrachtet einen Plan insbesondere als Ergebnis der Planung und als Basis für den Soll/Ist-Vergleich, so hat dies Konsequenzen: Ein Plan muss nicht mehr (zwingend) ausführbar sein, sondern sollte insbesondere geeignet sein, Planungsschwächen zu identifizieren und den Soll/Ist-Vergleich zu ermöglichen. Im Sinne der Metapher des „Process Programmings“ müssen Pläne damit nicht prozedural „ausformuliert“ werden, sondern sie können deklarativ auf beliebiger Abstraktionsebene spezifiziert werden – Unter- und Überspezifikation eingeschlossen. Die Umsetzung des Plans geht von intelligenten Mitarbeitern aus, die mit ihrer Intelligenz und Erfahrung selbst dafür sorgen, sich möglichst im Rahmen der Deklaration zu bewegen. Sie können auch bewusst und eigenverantwortlich vom Plan abweichen, wenn es die Projektumstände erfordern, da sie keine Maschine zur Einhaltung des Plans „zwingt“.

Die Vorteile einer solch deklarativen Betrachtungsweise sind anhand des „Feedback-Loop-Problems“ [Mau96] erkennbar: Wie Abbildung 1 zeigt, seien die Tasks A und B wechselseitig voneinander abhängig: Es sollen zwei Dokumente entstehen, die sich gegenseitig referenzieren. In der herkömmlichen Workflow-Betrachtungsweise mit Eingabe- und Ausgabe-Artefakten und einem Taskbeginn erst dann, wenn alle Eingaben vorliegen (z.B. [MD+00]) ist dafür keine Lösung zu finden (vgl. auch [Mau96]). Wenn Alice ihr Artefakt fertig stellt und es als Input für Bobs Task dient, dann kann Bob zwar Alice' Dokument referenzieren, aber nicht mehr umgekehrt. Wenn die Tasks hingegen wechselseitig aufeinander warten, entsteht ein Deadlock.

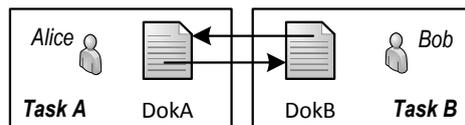


Abbildung 1: Feedback-Loop-Problem

Mit der oben beschriebenen deklarativen Herangehensweise kann das Problem sinnvoll als Plan modelliert werden (siehe Abbildung 2). Die Semantik der Ports und Konnektoren ist dabei kein Artefaktfluss, sondern eine Sichtbarkeit auf Artefakt-Repositorys: Alice macht Bob alle als „DokA“ gekennzeichneten Artefakte sichtbar – dasselbe gilt umgekehrt für Bob und die Kennzeichnung mit „DokB“. Alice sichert außerdem zu, dass sie für folgendes Sorge trägt: *Falls bei Bob mindestens ein „DokB“ für sie sichtbar ist, dann existiert in ihrem Repository genau ein mit „DokA“ beschriftetes Artefakt und dieses Artefakt referenziert alle von Bob mit „DokB“ gekennzeichneten Artefakte.* Bob garantiert das Gleiche in umgekehrter Richtung.

Wenn zu einem beliebigen Zeitpunkt im Projekt nun überprüft wird, ob der Plan erfüllt ist, so existieren genau zwei Möglichkeiten, in denen sowohl Bob als auch Alice *gleichzeitig* ihre Zusicherungen erfüllen: Entweder keiner von beiden hat ein Dokument pro-

duziert, oder sie haben gemeinsam genau den erwünschten Zustand erreicht. *Wie* die beiden nun im Detail kooperieren, um ihre Ziele gemeinsam zu erreichen, ist bewusst offengehalten, um hier die Intelligenz und Verantwortung der Projektmitarbeiter zu nutzen. Der Ansatz ist damit vergleichbar mit deklarativen Sprachen wie PROLOG oder SQL bei denen der Programmierer nur die Eigenschaften der Lösung spezifiziert, ohne den Lösungsweg vorzugeben.

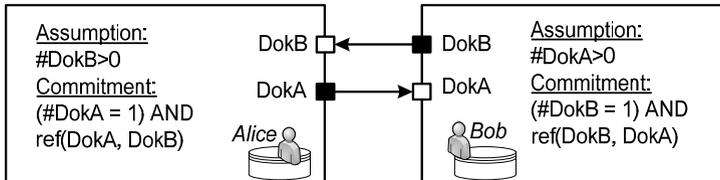


Abbildung 2: Deklarative Lösungsbeschreibung des Feedback-Loop-Problems

Auf Basis dieser Sichtweise können nun komplexere Planungsprobleme angegangen werden. Durch Übertragung bekannter Konzepte wie Verfeinerung, Korrektheit, oder Realisierbarkeit (vgl. [BS01]) können beispielsweise Pläne hinsichtlich ihrer Qualitätseigenschaften charakterisiert werden, um damit u.a. die in [AH07] genannten Forschungslücken zu adressieren.

3 Zusammenfassung und Diskussion

KPM-Szenarien sind herausfordernd und werden bisher nicht ausreichend durch Werkzeuge und Vorgehensmodelle unterstützt. Für eine bessere Unterstützung ist zunächst eine bessere theoretische Durchdringung der PM-Domäne und insbesondere der Planung notwendig. Die Informatik hat in der Software-Prozess-Forschung bewiesen, dass eine Übertragung ihrer formal-mathematischen Prinzipien von Computersystemen auf Softwareprozesse enorme Fortschritte und Einsichten bringen kann. Allerdings lag der Forschungsschwerpunkt meist auf der Prozessautomatisierung, weniger auf der Unterstützung von Planung und Kontrolle, sodass insbesondere Geschäftsprozesse profitieren konnten. Zukünftige Forschungsaktivitäten sollten die positiven Erfahrungen der SW-Prozessforschung aufgreifen und ganz gezielt auf Projektplanung übertragen, um die Domäne theoretisch und mit Hilfe der Mathematik zu durchdringen. Schwerpunkte sollten die Frage nach Qualitätseigenschaften von Plänen (z.B. Korrektheit, Realisierbarkeit etc.) sowie die Beachtung von Informationssicherheitsaspekten (z.B. zur Konsistenthaltung von Plänen ohne die Pläne zu enthüllen) sein. Wenn die formalen und theoretischen Grundlagen geklärt sind, können darauf aufbauend einerseits Werkzeuge gebaut werden. Andererseits ergeben sich ganz neue Fragestellungen im Kontext von Vorgehensmodellen, beispielsweise wie ein Vorgehensmodell dabei unterstützen kann, stets „gute Pläne“ abzuleiten.

Literaturverzeichnis

- [AH07] Alberts, D. S.; Hayes, R. E.: Planning: complex endeavors. Assistant Secretary of Defense (C3I/Command Control Research Program) Washington DC, 2007.
- [BC+07] Bendraou, R.; Combemale, B.; Crégut, X.; Gervais, M. P.: Definition of an Executable SPEM 2.0. In: Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific. IEEE, 2007; S. 390-397
- [BE+10] Buschermöhle, R.; Eekhoff, H.; Frommhold, H.; Josko, B.; Schiller, M.: SUCCESS 2010. Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Softwareentwicklungsjahren in Deutschland (V. 2.0). 2010.
- [BL+95] Bröckers, A.; Lott, C. M.; Verlage, M.; Rombach, H. D.: MVP-L language report version 2. 1995.
- [BS01] Broy, M.; Stölen, K.: Specification and development of interactive systems. FOCUS on streams, interfaces, and refinement. New York, NY: Springer (Monographs in computer science). 2001.
- [CN+03] Chen, F.; Nunamaker Jr, J. F.; Romano, N. C.; Briggs, R. O.: A collaborative project management architecture. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, IEEE, 2003; S. 12ff.
- [DL04] Dvir, D.; Lechler, T.: Plans are nothing, changing plans is everything: the impact of changes on project success. In: Research Policy 33 (1), 2004; S. 1-15
- [DS02] Damm, D.; Schindler, M.: Security issues of a knowledge medium for distributed project work. International Journal of Project Management, 2002; 20. Jg., Nr. 1, S. 37-47
- [EG91] Emmerich, W.; Gruhn, V.: FUNSOFT nets: a Petri-net based software process modeling language. In: Proceedings of the 6th international workshop on Software specification and design. IEEE Computer Society Press, 1991; S. 175-184
- [FB11] Friedrich, J.; Bergner, K.: Formally founded, plan-based enactment of software development processes. Proceedings of the 2011 International Conference on Software and Systems Process. ACM, 2011.
- [Fis10] Fischer, E.: Operationalisierung des Projektcontrollings. Verlag Dr. Hut, 2010.
- [Gna05] Gnatz, M. A. J.: Vom Vorgehensmodell zum Projektplan. VDM Publishing, 2005.
- [KH02] Koskela, L. J.; Howell, G.: The underlying theory of project management is obsolete. In: The PMI Research Conference: PMI, 2002; S. 293-302
- [Mau96] Maurer, F.: Working Group Report on Computer Support in Project Coordination. In: WETICE. 1996. S. 200-206
- [MD+00] Maurer, F.; Dellen, B.; Bendeck, F.; Goldmann, S.; Holz, H.; Kötting, B.; Schaaf, M.: Merging Project Planning and Web-Enabled Dynamic Workflow Technologies. In: IEEE Internet Computing 4 (3), 2000; S. 65-74
- [MH04] Münch, J.; Heidrich, J.: Software project control centers: concepts and approaches. In: Journal of Systems and Software, 2004; 70 (1-2), S. 3-19
- [Ost87] Osterweil, L.: Software processes are software too. Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, 1987.
- [RB+10] Ramler, R.; Beer, W.; Klammer, C.; Wolfmaier, K.; Larndorfer, S.: Concept, Implementation and Evaluation of a Web-Based Software Cockpit. In: Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on. IEEE, 2010; S. 385-392
- [Sta73] Stachowiak, H. Allgemeine Modelltheorie. Springer. 1973.
- [WC+11] Wise, A.; Cass, A. G.; Lerner, B. S.; McCall, E. K.; Osterweil, L. J.; Sutton Jr, S. M.: Using Little-JIL to coordinate agents in software engineering. In Engineering of Software. Springer Berlin Heidelberg, 2011; S. 383-397

Elche fangen - Von der Kunst in IT-Projekten Beobachtungen und Interviews durchzuführen

Christa Weßel

Dr. Christa Weßel
Frankfurt am Main
mail@christa-wessel.de

Abstract: IT-Projekte leben von der engen Zusammenarbeit zwischen Entwicklern, Nutzern und anderen Stakeholdern. Beobachtungen und Interviews sind starke Instrumente, um Bedürfnisse, Wissen, Ideen, Befürchtungen und Erwartungen kennenzulernen. Sie dienen vor allem der Exploration, dem Entdecken, Verstehen und Erklären. Sie finden außerdem Einsatz in der formativen und summativen Evaluation. Im "Walk and Talk" können Akteure in IT-Projekten Beobachtungen und Interviews in Unternehmens-, Prozess-, Projekt- und Anforderungsanalysen anwenden und die Ergebnisse für die Arbeit in ihren IT-Projekten verwenden. Ein Walk and Talk ist eine Kombination aus Beobachtung, Begehung und Gesprächen, die als leitfaden-gestützte Interviews geführt werden. Er erstreckt sich über einige Stunden bis zu zwei Tagen. Außerdem können noch Fokusgruppen-Interviews oder Workshops hinzukommen. "Agil" ist dieser Ansatz, weil er sich mit ein wenig Übung strukturiert, flexibel, ergebnisoffen und mit Fokus auf die Beforschten einsetzen lässt. Die Beforschten sind in der Regel die Nutzer und andere Kunden. Dabei können die Durchführenden das Konzept des "Elchs auf dem Tisch" nutzen. Vier Elche stehen für Tabus in der Arbeitswelt, wie Macht, Karriere, Beziehungen und Fehler. Diese Tabus können sowohl negative als auch positive Energie in einem Unternehmen entwickeln.

1 Soziale Aspekte und Standardisierung in IT-Projekten: Die Rolle der Sozioinformatik

"Brauchen wir mehr Detaillierung der Verfahren oder ist es nicht besser, gut ausgebildete Fachleute mit einem gut gefüllten 'Vorgehens-' und 'Projektmanagement-Werkzeugkasten' auszustatten?" Diese Frage des Programmkomitees zum Workshop 2014 der GI-Gruppe Vorgehensmodelle möchte ich mit einem eindeutigen "das kommt darauf an" beantworten.

In IT-Projekten brauchen die Akteure vielfältige methodische, fachliche und soziale Kompetenzen. Schritte des Projektmanagement, der Software-Entwicklung oder Maintenance lassen sich sicher gut in detaillierten Vorgehensmodellen beschreiben.

Wie sieht es jedoch mit dem Kennenlernen von Bedürfnissen, Wissen, Ideen, Befürchtungen und Erwartungen von Nutzern, Auftraggebern und anderen Stakeholdern in IT-Projekten aus? Das ist ein wichtiges Thema in der Sozioinformatik.

Kling hat Sozioinformatik so beschrieben [K100]: "Social informatics is the body of research that examines the design, uses, and consequences of information and communication technologies in ways that take into account their interaction with institutional and cultural contexts." In der Sozioinformatik geht es um die Zusammenhänge von Informations- und Kommunikationstechnologie und sozialen Veränderungen.

Sozioinformatik nutzt Methoden der Soziologie und der Informatik. Dabei geht es nicht nur um Forschung sondern auch um Entwicklung und um die praktische Anwendung.

Im Vokabular der Soziologie ausgedrückt bedeutet "praktische Anwendung", ins "Feld" zu gehen und mit Stakeholdern zu sprechen, sie zu beobachten und sie in Interviews und mit Fragebögen zu befragen. Den Begriff der Feldstudie hat Kurt Lewin Mitte des vergangenen Jahrhunderts geprägt [Le63]. Stark wird das Instrumentarium der Soziologie, wenn die "Forschenden" triangulieren, also mehrere Methoden, Konzepte, Theorien und / oder Untersucher miteinander kombinieren [AIM03].

Qualitative Methoden, wie Beobachtungen und leitfaden-gestützte Interviews dienen vor allem dem Entdecken und Erklären und der Hypothesengenerierung. Zur täglichen Arbeit von IT-Experten gehört genau dies: entdecken, verstehen, erklären.

Dabei sollten sie sich auch mit Unausgesprochenem auseinandersetzen. Denn oftmals werden die, die einen Bereich analysieren, als Bedrohung gesehen: Was haben wir falsch gemacht? Sollen wir etwas verlieren, Zuständigkeiten, Mitarbeiter, Projekte, Budgets, oder gar ganz aufgelöst werden? Dieses Unausgesprochene oder Unbesprechbare beschreibt Dennis Perkins als die vier "Elche" [Pe88]. Die Metapher des Elches steht für Unausgesprochenes, insbesondere für Tabus wie Macht, Karriere, Beziehungen und Fehler.

Sollen IT-Experten Soziologie studieren? Nein, aber sie können einige Grundwerkzeuge der soziologischen Forschung erlernen und anwenden [We13], und sie können von Unternehmensberatung, Coaching und Organisationsentwicklung etwas über den Umgang mit "Elchen" lernen [Cu08, Pe88, We13]. Lassen Sie uns zunächst die Anwendung von Beobachtungen und Interviews innerhalb einer Organisationsanalyse betrachten. Es geht um einen "Walk and Talk".

2 Der Walk and Talk

2.1 Eine Methode der Organisationsanalyse

Ein Walk and Talk ist eine Methode der Organisationsanalyse. Er ist zum Beispiel im Rahmen von Unternehmensanalysen, Machbarkeitsstudien, Prozessverbesserungen oder Produktentwicklungen einsetzbar. Ziel ist es, eine entspannte und realitätsnahe Erhebung durchzuführen. Die Menschen sollen von ihrer Arbeit erzählen, Einblick in ihr Wissen, ihre Befürchtungen, Erwartungen und Ideen geben. Damit soll der sogenannte "Hawthorne-Effekt" klein gehalten werden. Dieser von Roethlisberger und Dickson beschriebene Effekt besagt, dass Menschen ihr Verhalten ändern, sobald sie sich beobachtet fühlen [RD05].

Ein Walk and Talk erstreckt sich über einige Stunden bis zu zwei Tagen. Ein längerer Zeitraum ist für alle Beteiligten zu anstrengend. Er ist eine Kombination aus Beobachtung, Begehung und Gesprächen, die als leitfaden-gestützte Interviews geführt werden. Außerdem können noch Fokusgruppen-Interviews oder Workshops hinzukommen. Spaziergehen und Plaudern ist für die Durchführenden alles andere als harmlos, sondern ein sehr intensiver Einsatz ihrer mentalen, psychischen und auch körperlichen Kräfte.

2.2 Vorbereitung

Zum Instrumentarium für Gespräche und Beobachtungen im Walk and Talk gehören von den Untersuchern vorbereitete Checklisten und Leitfäden. Typische Fragestellungen sind:

- Was sind Kernmerkmale unserer Unternehmenskultur?
- Wie können wir die Mitwirkung und das Commitment unserer Mitarbeiter für die Neuorientierung in unserer Geschäftsstrategie stärken?
- Welche Vorbehalte und Zusprüche gibt es für Projekt X?
- Wie sind die Arbeits- und Kommunikationsabläufe in Abteilung Y?
- Welche Werkzeuge benötigt der Bereich Z für Dienstleistung A?
- ...

Als Externe sollten die Untersucher einen internen Ansprechpartner im Unternehmen haben. Mit diesem klären die Untersucher, wann, wo und mit wem sie wie lange sprechen wollen.

Gemeinsam entwickeln sie Leitfäden, zum einen für den gesamten Zeitraum des Walk and Talks mit einer leitenden Frage und weiteren Fragen sowie einem zeitlichen Ablauf „Wann sind Sie wo?“ Für besondere Gespräche, beispielsweise mit Führungskräften sollten sie eigene Leitfäden entwickeln.

Der interne Ansprechpartner organisiert vor Ort den Ablauf. Dazu gehören Verabredungen zu den Gesprächen und auch Gruppentreffen, wenn beispielsweise ein Workshop mit dazu gehören soll. Für einen Workshop und die Freistellung der Mitarbeiter für diese Stunden muss die Einwilligung des Auftraggebers vorliegen.

2.3 Durchführung

Wichtig ist es, die Führungskraft des zu untersuchenden Bereichs so früh wie möglich im Verlauf des Walk and Talks zu treffen. Dies gilt für den Fall, dass diese Führungskraft der Auftraggeber ist, und im Besonderen dann, wenn die Unternehmensleitung den Auftrag erteilt hat.

Nach jedem Gespräch fertigen die Untersucher "Feldnotizen" an. Diese enthalten, wann

sie wie lange mit wem gesprochen haben und die Hauptpunkte ihres Gesprächs. Am Abend einer zweitägigen Analyse werten die Untersucher ihre Notizen aus und verfeinern den bis dahin vorläufigen Plan für Tag 2. Dazu sind folgende Fragen hilfreich:

- Wollen die Untersucher noch mal mit jemandem von Tag 1 sprechen?
- Wollen die Untersucher mit jemandem sprechen, den sie bislang nicht vorgesehen hatten?
- Wollen die Untersucher eine Gruppe interviewen?
- Falls für Tag 2 ein Workshop vorgesehen ist: Wollen die Untersucher etwas an den Fragestellungen und / oder den Aufgaben ändern?

"Warum wollen Sie zwei Tage zu uns kommen?" Eine typische Frage von Auftraggebern, wenn ich mit ihnen den Ablauf eines Walk and Talks abstimme. - "Weil Sie sich am zweiten Tag nicht mehr verstellen." Das antworte ich natürlich nicht. Es ist aber ein Körnchen Wahrheit darin. Nach dem ersten Tag haben sich die Menschen ein wenig an die Untersucher gewöhnt.

Für einen Workshop am zweiten Tag spricht, dass die Untersucher und die Teilnehmer gemeinsam über die Situation, Ziele und Maßnahmen nachdenken. Sollten die Untersucher oder andere im Workshop Notizen und Skizzen auf einem Flipchart oder ähnlichem angefertigt haben, sollten sie diese – natürlich nur mit Erlaubnis der Autoren – fotografieren und in die Feldnotizen mit aufnehmen. Diese Reflexionen können und sollten neben der Analyse der eigenen Beobachtungen und Gespräche in den Bericht der Untersucher einfließen.

Der zweite Tag schließt mit der Verabschiedung von der Leitung des Unternehmens, Bereichs oder Projekts. Ein kurzes Gespräch von einer viertel Stunde genügt.

2.4 Analyse und Bericht

Die ausführliche Analyse der Feldnotizen und Dokumentationen fließt in einen Bericht ein, den die Untersucher mit dem Auftraggeber und möglichst auch mit den Menschen besprechen, die sie "beforscht" haben. Je nach Auftrag - Unternehmensanalysen, Machbarkeitsstudien, Prozessverbesserungen oder Produktentwicklungen - gehören auch Empfehlungen und eine Skizzierung der weiteren Schritte in den Bericht.

2.5 Anwendungen

Ein Walk and Talk hat seine Wurzeln in der explorativen empirischen Sozialforschung. Neben seiner Anwendung in der Organisationsberatung und -Entwicklung und im Change Management ist er im Projektmanagement und im Software-Engineering vor allem in Machbarkeitsstudien, Prozessverbesserungen, Anforderungsanalysen und Evaluationen einsetzbar. Der Ablauf ist definiert und kann an das jeweilige Projekt adaptiert werden (Tabelle 1).

Tag	Schritt
Tag/e X	Vorbereitungen
Tag 1	Treffen mit dem internen Ansprechpartner
	Treffen mit Bereichs-/ Abteilungs-/ Projekt-Leitung
	"spazieren gehen und plaudern"
	Reflexion mit internem Ansprechpartner
	Nachbereitung Tag 1 und Vorbereitung Tag 2
Tag 2	Treffen mit dem internen Ansprechpartner
	"spazieren gehen und plaudern"
	Workshop
	"spazieren gehen und plaudern"
	Abschied von Bereichs-/ Abteilungs-/ Projekt-Leitung
	Reflexion mit internem Ansprechpartner
	Nachbereitung Tag 2
Tag/e Y	Analyse und Bericht
Tag/e Z	Besprechung und Reflexion mit Auftraggeber und Beforschten

Tabelle 1: Schritte im "Walk and Talk"

Die Feldnotizen aus den Beobachtungen und Gesprächen und die Dokumentationen aus den Workshops bieten Datenmaterial, das Eingang in die Beschreibung von Ziel, Ressourcen, Risikofaktoren, Aufgaben und Meilensteinen findet. Die Untersucher können Persona Models und Szenarien entwickeln [We10]. Damit kommen sie einer wichtigen Forderung in qualitativen Untersuchungen nach, der Wahrung der Vertraulichkeit und des Datenschutzes [AIM03].

Für die Einordnung des Walk and Talk in Vorgehensmodelle schlage ich in Anlehnung an die Gliederung bei [Ho07] folgendes vor:

- Wasserfallmodell: Zielerforschung, Anforderungsanalyse, Verifikation
- Scrum: am ehesten Projektstart
- XP: Entwicklung von User Stories
- RUP: Konzeptionalisierung, Anforderungsanalyse, Konfigurations- und Änderungsmanagement
- V-Modell [XT]: Anforderungsdefinition, eventuell Tests
- OEP (Object Engineering Process) [Oe06]: vor allem folgende Schritte
 - Systemidee und Zielsetzung entwickeln

- Interessenhalter identifizieren
- Interessen der Interessenhalter identifizieren
- Geschäftsanwendungsfälle (Business Use Case) identifizieren
- Anwendungsfälle essenziell beschreiben
- Systemanwendungsfälle identifizieren
- Anforderungen (Requirements) beschreiben

3 Strukturiert, flexibel, offen und agil

Forscher, Entwickler und Dienstleister brauchen ein Portfolio von soliden, methodisch fundierten Instrumenten, um Unternehmensanalysen, Machbarkeitsstudien, Prozessverbesserungen oder Produktentwicklungen durchführen zu können [Cu08, Kl00, Le63].

Dazu gehören Beobachtungen, Interviews und Workshops. Der "Walk and Talk" ist eine strukturierte und gleichzeitig flexible Kombination aus Beobachtung, Gesprächen, Interviews und Workshops. Mit ihm können die Untersucher neue Erkenntnisse gewinnen, Zusammenhänge verstehen und Hypothesen generieren.

Prozessmodelle, wie der Rational Unified Process (RUP), der Object Engineering Process und das Agile Programming, berücksichtigen Aspekte wie evolutionäre und inkrementelle Entwicklung, Flexibilität und multidisziplinäre Teams. Ein Walk and Talk weist diese Eigenschaften auf. Daher könnte für eine Organisation (kommerziell oder forschend) die Integration von Walk and Talks in ihre Projekte interessant sein. Zu berücksichtigen ist jedoch, dass manche Prozessmodelle explizit das Vorgehen in Software-Projekten beschreiben, beispielsweise das V-Modell und auch – so [LL13] – das RUP. Dies hat eine Organisation bei entsprechender vertraglicher Verpflichtung in ihrer Projektdurchführung zu berücksichtigen.

4 Elche fangen

Oftmals müssen sich Untersucher auch mit Unausgesprochenem und für die Beforschten Bedrohlichem auseinandersetzen. Perkins' vier "Elche" Macht, Karriere, Beziehungen und Fehler lassen sich durch einen bewussten, offenen, respektvollen und nicht wertenden Umgang damit "bändigen" [Pe88, We13]. Sie können auch positive Energie entwickeln (vgl. dazu [Br14]). Macht ist nötig, um Entscheidungen umsetzen (lassen) zu können, Karriere ist nichts anderes als der (Berufs)Weg. Menschen sind soziale Wesen und haben Beziehungen, die sie gestalten können. Fehler werden nur dann schlimm, wenn sie verdrängt werden. Der positive Umgang mit ihnen lässt sich im Alltag umsetzen und im Risiko-, Chancen- und Qualitätsmanagement verankern.

Literatur

- [AIM03] Ammenwerth, E.; Iller, C.; Mansmann, U.: Can evaluation studies benefit from triangulation? A case study. *Int J Med Inform* 2003; 70: 237-248. <http://elske-ammenwerth.de/Publikationen/z18.pdf>
- [Br14] Bruch, H. et.al.: *Energy Factory. Organisationale Energie*. St. Gallen, CH, 2014. <http://www.energyfactory.com/>
- [Cu08] Cummings TG: (Ed.) *Handbook of Organization Development*. Los Angeles, Sage Publications 2008.
- [Ho07] Horn, T.: *Vorgehensmodelle zum Softwareentwicklungsprozess*. Aachen, 2001-2007. <http://www.torsten-horn.de/techdocs/sw-dev-process.htm> (und Unterseiten)
- [KL10] Kling, R.: *Learning About Information Technologies and Social Change: The Contribution of Social Informatics*. *The Information Society*, 2000; 16: 217–232. <http://www.indiana.edu/~tisj/readers/full-text/16-3%20kling.pdf>
- [Le63] Lewin, K.: *Feldtheorie in den Sozialwissenschaften: Ausgewählte theoretische Schriften*. Hrsg. von Dorwin Cartwright. Ins Dt. übertr. u. mit e. Verz. d. Schriften Lewins vers. von A. Lang u. W. Lohr. Bern; Stuttgart: Huber 1963 (en.: *Field Theory in Social Science*. 1947)
- [LL13] Ludewig, J.; Lichter, H.: *Software Engineering - Grundlagen, Menschen, Prozesse, Techniken*. 3., korrigierte Auflage. Heidelberg, dpunkt.verlag 2013.
- [Oe06] Oestereich, B.: *Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.1*. 8. Auflage. München, Oldenbourg Wissenschaftsverlag 2006.
- [Pe88] Perkins DNT. *Ghosts in the executive suite: Every business is a family business*. Branford, CT, Syncretics Group 1988.
- [RD05] Roethlisberger, FJ.; Dickson, WJ.: *Management and the Worker*. Volume V of Thompson K (Ed). *The Early Sociology of Management and Organizations*. London, Routledge 2005 (first published 1939).
- [We10] Weßel, C.: *Semi-strukturierte Interviews im Software-Engineering: Indikationsstellung, Vorbereitung, Durchführung und Auswertung – Ein Fall-basiertes Tutorium*. In: Fähnrich KP, Franczyk B (Hg.). *INFORMATIK 2010. Service Science – Neue Perspektiven für die Informatik*. Band 2. Volume P-176. Gesellschaft für Informatik, Bonn 2010: 1097-1107. http://www.christawessel.de/app/download/9502849021/wessel2010_inf10_int_paper.pdf?t=1399987491
- [We13] Weßel, C.: *Basiswissen Consulting. Der Elch auf dem Tisch und andere Beratungskonzepte*. Heidelberg (und andere), mitp Verlag 2013.

Praktische Umsetzung eines internationalen Web-Relaunchs mit Hilfe einer hybriden Projektmethodik

Hans Christian Liebig

HTWSaar
Waldhausweg 14
66123 Saarbrücken
christian.liebig@htwsaar.de

Abstract: Internationale Unternehmen stehen gerade in exportorientierten Branchen vor der Herausforderung international ihre Produkte zu bewerben und in zahlreichen Zielländern präsent zu sein. Neben der Landessprache sind kulturelle, technische und auch juristische Herausforderungen zu lösen. Ausgehend von einem realen internationalen Internetprojekt unter der Beteiligung verschiedener Kulturkreise (u. a. US-amerikanisch, Chinesisch, Indisch, Europäisch) stellt die Arbeit Herausforderungen und Lösungsmöglichkeiten durch eine hybride Projektmethodik anhand einer Fallstudie dar. Ein besonderer Fokus liegt in der Integration externer Dienstleister

1 These

Komplexe, internationale IT-Projekte, wie ein Web-Relaunch, können von einer hybriden Projektmethodik profitieren, soweit einige Voraussetzungen beachtet werden.

2 Einleitung

Mittelständische Unternehmen in Deutschland sind vielfach international ausgerichtet und exportorientiert. Aufgrund ihrer Größe sind sie jedoch nicht in der Lage in jedem ihrer Zielmärkte mit einer vollständigen unternehmerischen Infrastruktur präsent zu sein. In Webprojekten werden in aller Regel zahlreiche Informationen aus unterschiedlichen Bereichen den Besuchern angeboten. Dies führt insbesondere bei IT-Projekten mit einem starken Kundenbezug zu vielfältigen Herausforderungen, die insbesondere bei Webprojekten gelöst werden müssen. Wie in [K111] dargestellt ist, sind insbesondere auch technische Fragestellungen zu lösen, um in den Zielmärkten entsprechende Wettbewerbsvorteile zu haben.

3 Fallstudie

Die vorliegende Fallgestaltung basiert im Wesentlichen auf einem realen Projekt, welches sowohl mit internen, wie auch externen Projektteilnehmern realisiert wurde:

Ein mittelständisches Unternehmen, in der Größenordnung von 3.000 Mitarbeitern und einem konsolidierten Jahresumsatz von ca. 1 Mrd. Euro hat als Zielmarkt 150 Länder identifiziert und die Webpräsenz soll in 30 Sprachen realisiert werden. Neben den Mitgliedsländern der EU, sollen insbesondere die sogenannten BRIC-Staaten berücksichtigt werden. Das Unternehmen ist in der Elektrotechnik im Investitionsgüterbereich tätig. Die Produkte seien erklärungsbedürftig und unterliegen regulatorischen Anforderungen (s. z. B. Maschinenrichtlinie der EU [EU06]).

3.1 Projekt Anforderungen

Die Reaktionszeit jedes Seitenabrufs im geplanten Zielgebiet soll deutlich unter 3 Sekunden liegen, um ein abwandern potentieller Kunden zu verhindern. Daneben sollen folgende Altsysteme angebunden werden: ERP, PIM, MAM und CRM-System. Hierzu ist insbesondere eine geeignete Infrastruktur zu entwerfen, die dynamisch Geschäftsprozesse in Form von technischen Workflows umsetzt. Zudem ist davon auszugehen, dass die Produkte variantenreich sind, so dass hier neben der reinen Suchfunktionalität auch Produktkonfiguratoren verwendet werden sollen. Daneben ist die Ausgabe in unterschiedlichen Sprachen (inkl. der jeweiligen Zeichensätze und Übersetzungsworkflows) zu gewährleisten. Die jeweiligen rechtlichen Rahmenbedingungen in den unterschiedlichen Zielländern sind einzuhalten.

3.2 Kulturelle Anforderungen

Wie in [VBW12] ausführlich dargestellt wurde, sind insbesondere bei internationalen IT-Projekten die kulturellen Unterschiede zu berücksichtigen (s. auch [Ho80] [Ho93]). Diese sind in zwei Perspektiven zu berücksichtigen: Zum einen im Projektmanagement und zum anderen bezüglich des Projektergebnisses.

Gerade ein Web-Relaunch trifft aufgrund seiner gewünschten Außenwirkung in den Kernbereich von Kulturen. Im Rahmen der Projektabwicklung zeigte sich, dass im US-amerikanischen Umfeld neueste Technologien (insbesondere Integration von Social Media Elementen) erwartet werden, während gerade diese Technologien und deren Anwendung im europäischen Kulturkreis teilweise unerwünscht bzw. verboten sind (Verstoß gegen Datenschutzgesetze). Auch bezüglich der Darstellung von Menschen ist auf die Einhaltung einer „Diversity“-Repräsentation zu achten.

4 Auswahl der Projektmethodik

Es gibt eine Vielzahl von Projektmethoden. Im Folgenden wird zunächst das klassische Wasserfallmodell den agilen Methoden gegenübergestellt und eine hybride Projektmethodik dargestellt, welche in der Fallstudie zum Einsatz kam.

4.1 Das klassische Wasserfallmodell und dessen Implikationen

Im Rahmen des klassischen Wasserfallmodells werden durch Lasten- und Pflichtenhefte sehr genau und umfassend alle wesentlichen Anforderungen sowie deren Realisierung beschrieben. Das Projekt wird in starren Phasen durchlaufen. Gerade in der Anforderungs- und Projektplanungsphase müssen alle wesentlichen Arbeitspakete erkannt, beschrieben und vom Aufwand zuverlässig abgeschätzt werden. Dies ist bei hoch standardisierten Projekten, wie z. B. im Baubereich gut möglich, bei innovativen, komplexen IT-Projekten jedoch sehr schwierig, da es häufig Sachverhaltsänderungen (z. B. neue Softwareversionen) gibt, die die Anforderungen an das Projekt wesentlich verändern. Sollte es Abweichungen zu den festgelegten Spezifikationen geben, so lässt sich bei einem ordnungsgemäßen Vorgehen die Verantwortung klären und damit die Haftung abgrenzen.

4.2 Agile Methoden

Agile Methoden, wie z. B. Scrum [Sc97] verzichten demgegenüber auf eine initiale Detailplanung, sondern durchleben das Projekt in sehr kurzen Phasen (Sprints) wieder und wieder. Von Sprint zu Sprint werden dann einerseits die Anforderungen angepasst, wie auch auf einer Metaebene die Zusammenarbeit verbessert.

Innerhalb eines Unternehmens, in denen der Leistungsaustausch im Unternehmen selbst stattfindet sind solche Methoden auch gut anwendbar. Behält doch das Unternehmen das direkte Direktionsrecht gegenüber allen Akteuren. Schwierig wird es allerdings, wenn innerhalb eines Projektes externe Unternehmen in der Weise eingebunden werden sollen, wie es agile Methoden erfordern: Die externen Berater etwa sollen integrierte Projektmitglieder werden. Die Verantwortung zwischen den Unternehmen schwimmt damit und ist im Falle einer etwaigen juristischen Auseinandersetzung nur noch schwer zuzuordnen. Rechtsabteilungen neigen daher dazu gerne in typisierten Verträgen die Projektabwicklung zu strukturieren. Das ist gerade bei agilen Methoden nur schwer möglich. Folgendes Beispiel soll dies erläutern: Der klassische Werkvertrag setzt voraus, dass das abzuliefernde Werk hinreichend spezifiziert ist, dies ist jedoch bei agilen Methoden nicht möglich. Ein Dienstvertrag greift in aller Regel auch nicht, da für die einzelnen Sprints der Erfolg, d.h. die Erstellung funktionsfähiger Artefakte, geschuldet werden soll.

4.3 Hybride Methoden

Gerade im Zusammenspiel von internen und externen Projektmitarbeitern besteht das Bedürfnis nach Rechtssicherheit auf der einen Seite und Projektdynamik auf der anderen

ren Seite. Dies lässt sich durch hybride Projektmethoden realisieren. Die Ausprägung dieser Mischform ist in verschiedenen Konstellationen denkbar:

1. Die Grundmethodik ist ein klassisches (z. B. Wasserfall-) Modell. Innerhalb eines oder mehrerer Teilprojekte wird komplett die Projektmethodik zu einem agilen Verfahren gewechselt. Das Risiko bei diesem Wechsel ist eine Verantwortungsdiffusion und finanzielle Risiken. Zum Beispiel: Die einzelnen Webseiten müssen bezüglich ihrer Darstellung und Funktionalität spezifiziert werden. Für dieses abgegrenzte Teilprojekt wird die Methodik gewechselt. Es wird ein Dienstleistungsvertrag geschlossen. Die externen Berater werden in das Projekt vollständig, wie eigene Mitarbeiter, integriert, und erstellen direkt in Sprints zusammen mit Vertretern der Fachabteilung die jeweiligen Webseiten als Click-Prototypen.
2. Die Grundmethodik ist ein agiles Verfahren. Innerhalb der Projektabwicklung entscheidet man sich zum Wechsel der Methodik: Anstelle eines oder mehrerer Sprints wird nun ein Wasserfallmodell eingesetzt. Bei dieser Art des Wechsels besteht das Risiko im Verlust der Anpassungsfähigkeit und Dynamik. Das lässt sich jedoch begrenzen, wenn die Umsetzung zeitnah möglich ist, wie es regelmäßig bei reinen Beschaffungen der Fall ist, die Anpassungen entweder nur eingeschränkt notwendig bzw. möglich sind oder wenn diese zu einem späteren Zeitpunkt nachzuholen sind. Ein Beispiel ist die Auswahl und Beschaffung eines Content Management Systems (CMS).
3. Eklektische Verfahren: Die Grundidee ist hierbei die jeweils geeigneten Methoden im jeweiligen Projektkontext frei miteinander zu kombinieren, ohne sich für eine führenden Methode zu entscheiden. Hier besteht das Risiko darin, dass sich die verantwortlichen „Projektleiter“ verzetteln und das Projekt unkontrolliert voranschreitet. Zudem besteht die Gefahr der Überforderung der Projektmitarbeiter.

In der vorliegenden Fallstudie wurde als Grundmethode die Wasserfall-Methode mit agilen Teilprojekten gewählt. Dies hat seinen Grund in der bis dahin mangelnden Erfahrung im betroffenen Unternehmen mit agilen Methoden und andererseits mit dem frühen Scheitern der reinen Wasserfallmethode im Rahmen und Planungsphase.

5 Erfahrungen in der Anwendung einer hybriden Projektmethodik im Rahmen der Fallstudie

Das im Rahmen der Fallstudie geschilderte Projekt wurde erfolgreich umgesetzt, ein SOA-basiertes PIM wurde als technische Lösung vorgeschlagen [Li12]. Die Budgetüberschreitung lag bei ca. 30%. Schwerer wog jedoch die erhebliche Verzögerung in der Umsetzung. Neben den rein technischen Gründen, führten auch Schwierigkeiten in der hybriden Projektmethodik zu Verzögerungen. Folgende Erfahrungswerte sind festzuhalten:

- Das zunächst angedachte Wasserfallmodell scheiterte mit der Erstellung des Pflichtenheftes. Dieses war im Verhältnis zur Projektgröße einerseits zu komplex, zu unvollständig und wurde letzten Endes durch die schnellen technischen Entwicklungen kurz vor der Fertigstellung überholt.
- Innerhalb des eigenen Unternehmens funktionierten agile Methoden sehr gut und verkürzten geplante Projektphasen deutlich (z. B. Webseitenanpassungen). Insbesondere verbesserte sich die Kommunikation zwischen den Projektteilnehmern (s. hierzu auch [A197]) durch die enge räumliche Nähe.
- Sobald unternehmensübergreifend mit agilen Methoden gearbeitet wurde, führten Moral Hazards zu vielfältigen Problemen: Ressourcen wurden nicht vereinbarungsgemäß zur Verfügung gestellt, Arbeiten je nach Auftragslage verschleppt bzw. nicht in der notwendigen Qualität abgeliefert.
- Rahmenverträge in Kombination mit „Sprintverträge“ zeigten sich als angemessene Lösung zur Lösung der Moral Hazard-Problematik auf der einen Seite und erhielten auf der anderen Seite die Dynamik im Projekt. Hierzu wurden in den Rahmenverträgen, neben den allgemeinen Regeln, wie sie bei vielen Unternehmen üblich sind, auch Regeln festgelegt, die insbesondere auch für agile Methoden sinnvoll sind: Klare Verantwortungszuordnung und definierte Reaktionszeiten, Schweigen als Annahme in bestimmten Fällen, Eskalationswege, Festlegung von Fehlerbehebungsroutinen, festschreiben der Vertragsart bei Sprints als Werkvertrag mit geschuldeten Erfolg. Abgrenzung zu Dienstleistungsverträgen, die nur im engen Rahmen zugelassen wurden. In den Sprintverträgen wurden die Arbeitspakete beschrieben, die im nächsten Durchlauf realisiert werden sollten.

6 Zusammenfassung und Ausblick

Die Realisierung eines Web-Relaunch stellt höchste Anforderungen sowohl in fachlicher Hinsicht, wie auch im Bereich des Projektmanagements. Der Autor hat die Erfahrung gemacht, dass die technischen Fragestellungen beherrschbar sind, jedoch die Entscheidungsfindung im interkulturellen Projektumfeld die eigentliche Herausforderung darstellt. Die Grundidee agiler Projektmethodik, mit schnellen kleinen Schritten sich dem Ziel zu nähern, findet gerade bei solchen Projekten ein gutes Anwendungsfeld. Andererseits ist in der unternehmensübergreifenden Zusammenarbeit auf die Beherrschung der Moral Hazard-Problematik zu achten. Diese ist in aller Regel nur durch ein angepasstes, striktes Vertragswerk sicherzustellen.

Literaturverzeichnis

- [Al97] Allen, T. J.: Architecture and Communication: Among Product Development Engineers, Cambridge, 1997, S. 165-197.
- [Eu06] E. R. Europäisches Parlament, Richtlinie 2006/42/EG; Maschinenrichtlinie, 2006.
- [Ho80] Hofstede, G.: Culture's Consequences, London, 1980.
- [Ho93] Hofstede, G.: Interkulturelle Zusammenarbeit, Wiesbaden, 1993.
- [KI11] North, K.: Wissensorientierte Unternehmensführung, 5. Auflage, Hrsg., Springer Gabler Verlag, Heidelberg, 2011.
- [Li12] Liebig, H. C.: Referenzmodell eines internationalen Produktinformationsmanagements, PIK, S. 183-186, 2012.
- [Sc97] Schwaber, K.: Scrum Development Process, OOPSLA Business Object Design and Implementation Workshop, London, 1997.
- [VBW12] von Stetten, A.; Beimborn, D.; Weitzel, T.: „Auswirkungen kulturspezifischer Verhaltensmuster auf das Sozialkapital in multinationalen IT-Projektteams - Ein Fallstudienansatz.“, Wirtschaftsinformatik, Bd. 54, S. 135-151, 2012.

Dimensionen von Projekten

Differenzierung von Standards und Methoden?

Klaus Schopka

Projektmanagement
Schopka GmbH
Blumenstr. 28.
85774 Unterföhring
klaus.schopka@schopka.com

Abstract: Projekte werden hinsichtlich des Einsatzes von Methoden, Standards, Prozessen, Controlling und Governance in der Praxis oft undifferenziert behandelt. Der Beitrag will sinnvolle und notwendige Differenzierungen aus der praktischen Erfahrung aufzeigen.

1 Einleitung

Die Genehmigung und Durchführung von Projekten in Unternehmen oder Organisationen erfolgt unter Berücksichtigung unterschiedlicher Standards wie Genehmigungsprozessen, Projektmanagementmethoden, Qualitätsmanagement, Beschaffungsmanagement, Vorgaben des Controllings, des Risikomanagements und der generellen Unternehmensgovernance. Hinter diesen Standards verbirgt sich eine Vielzahl von Einzelaktivitäten und Entscheidungen, die erheblichen Aufwand verursachen, Zeit beanspruchen und somit knappe Ressourcen im Projektumfeld binden. Der entstehende Aufwand wird gerne unterschätzt, da er in der Regel auf unterschiedlichste Stellen und Abteilungen verteilt ist.

Persönliche Erfahrungen des Autors, in unterschiedlichsten Positionen und Projekten, führen zu der Aussage: „**Standards und Methoden bieten Spielräume für Differenzierungen bei deren Einsatz. Diese müssen aber auch genutzt werden!**“ Diese Aussage wird in Folge näher betrachtet.

Der Einsatz von Standards an sich führt zu positiven Ergebnissen, wie auch aktuelle Studien betonen [Pm14]. So kann schon über die Genehmigung eines Projektes sichergestellt werden, dass es strategiekonform ist bzw. einen positiven Beitrag zur Zielerreichung leistet. Wenn die richtigen Projekte gemacht werden (Effektivität), kann im nächsten Schritt die Effizienz der Abwicklung erhöht werden. Hauptziel ist dabei die Erreichung des vereinbarten Projektergebnisses. Termine und Kosten sind flankierende Ziele auf dem Weg dorthin.

Einige Fragen bleiben aber offen:

- Ist ein konkretes Vorhaben diesen Aufwand wert? Liegt tatsächlich ein Projekt vor?
- Gibt es Dimensionen oder Merkmale von Projekten, die ein differenziertes Vorgehen erlauben?
- Welche Differenzierungen verträgt eine wirkungsvolle Projektsteuerung?

2 Dimensionen von Projekten

Projekte sind nach PMBOK „zeitlich begrenzte Vorhaben zur Schaffung eines einmaligen Produktes, Dienstleistung oder Ergebnisses“ [Pm13].

Gängige Beispiele für Dimensionen von Projekten und exemplarischen Hinweisen zu Anforderungen an Standards sind:

- **Projektgröße:** Als Maßstab gilt meist das Budget. „Kleine“ Projekte sind oft auch einfach und flach strukturiert. Hier sollte minimaler Aufwand betrieben werden.
- **Projektrisiken:** Hohe Risiken verlangen nach sorgfältigem Risikomanagement.
- **Komplexität:** Hohe Komplexität stellt Anforderungen an die Dokumentation, Planung und Kontrolle. Man sollte prüfen, ob eine Aufteilung des Projektes möglich ist.
- **Projektdauer:** „Langläufer“ erfordern gute Dokumentation und eine laufende Überprüfung des zugrunde liegenden „Business Case“.
- **Bedeutung für das Unternehmen:** Eine hohe Bedeutung ist verbunden mit besonderen Informationspflichten in das Management und hohen Erwartungen bzgl. der Termineinhaltung.
- **Auftraggeber (intern / extern):** Externe Kundenprojekte haben hohe Anforderungen an die Leistungsdokumentation, ohne die z.B. keine Abrechnung oder Leistungsabnahme möglich ist.
- **Innovationsgrad:** „Standardprojekte“ sind etabliert und werden nach einfachen Standards abgewickelt.
- **Projektphasen:** Eine solide Projektdefinition vor dem Projektstart erspart hohen Aufwand im Projektablauf. Je nach Projektphase sind unterschiedliche Abteilungen und Mitarbeiter in dem Projekt aktiv. Kritisch sind dabei die Übergaben.

Für eine realistische Beurteilung von Projekten ist die Kombination von mehreren Dimensionen erforderlich. Ein Projekt kann klein sein, gemessen an den Kosten; bedingt durch hohe Risiken und hohe Bedeutung für das Unternehmen wird es dennoch hohe Aufmerksamkeit erfordern.

Der Einsatz von Standards und Methoden muss diese Merkmale berücksichtigen. In Praxis geschieht dies durch Bildung von Klassifizierungen. Gängiges Beispiel ist eine Einteilung in A/B/C – Projekte unter Nutzung von Zuordnungstabellen.

Eine derartige Klassifizierung ist nur eine Hilfe zur ersten Sichtung von Projekten. Entscheidend ist die daraus resultierende Ausgestaltung der relevanten Standards. Auf wie viel Kontrolle bin ich als Unternehmen bereit, zu verzichten? Positiv formuliert: Wie viel Vertrauen in die Kompetenz und Selbstkontrolle meiner Mitarbeiter besitze ich? Die Festlegung der Merkmale und deren Ausprägung müssen unternehmensspezifisch erfolgen.

Unabhängig von der Ausgestaltung der Standards verbleiben Grundanforderungen mit entsprechendem Basisaufwand für jedes Projekt.

3 Projektwürdigkeit

Vor der Betrachtung eines differenzierten Einsatzes von Standards und Methoden muss die Grundsatzfrage geklärt sein: Soll ein konkretes, geplantes Vorhaben tatsächlich als Projekt abgewickelt werden? Ist das Vorhaben den damit verbundenen Aufwand wert? Hiermit soll einer Projektinflation vorgebeugt werden.

Die allgemeine Definition von Projekten hilft hier nicht weiter. Entscheidende Kriterien sind das angestrebte Ergebnis des Vorhabens und seine Bedeutung für das Unternehmen.

Damit werden auch Vorhaben projektwürdig, die prinzipiell im alltäglichen Liniengeschäft abgewickelt werden können. Grund hierfür kann die Komplexität des Vorhabens mit entsprechendem Koordinationsaufwand sein. Ein einfaches Beispiel aus der IT ist der Aufbau neuer Serversysteme. Noch vor 10 bis 15 Jahren wurden hierfür teilweise eigene Projekte gestartet. Hauptgrund war die organisatorische Komplexität der Zuarbeit vieler Stellen mit zumeist kleinen Teilaufgaben. Durch technologische Entwicklungen (Virtualisierung, Cloud) und organisatorische Neuordnungen von Einzelaufgaben ist dies mittlerweile eine betriebliche Standardaufgabe, die in Projekten abgerufen werden kann.

Möglich ist auch eine Abgrenzung von Vorhaben nach wirtschaftlichen Grenzen, die unternehmensspezifisch gesetzt werden können.

Aber auch die Abwicklung von Vorhaben, z.B. durch den IT-Betrieb, hat Grenzen. ITIL wird hier deutlich: „...there is a tendency not to use project management processes when they would in fact be appropriate.” Und benennt eine Reihe konkreter Vorteile der Projektabwicklung [Bm11].

Die Entscheidung über die Projektwürdigkeit eines konkreten Vorhabens muss vor Beginn des Genehmigungsverfahrens fallen.

4 Differenzierter Einsatz von Methoden und Standards

Wenn geklärt ist, dass tatsächlich ein Projekt abzuwickeln ist, muss entschieden werden, welche Anforderungen an diese Abwicklung zu stellen sind. In aller Regel wird hierfür ein einzelnes Merkmal nicht ausreichend sein. Ein wertmäßig „kleines“ Projekt kann mit hohen Risiken für das Unternehmen verbunden sein und dadurch hohe Aufmerksamkeit erfordern.

Die Merkmale und ihre relevanten Ausprägungen müssen transparent, verständlich und nachvollziehbar definiert sein. Hohe Finanzmathematik hilft hier nicht weiter.

4.1 Genehmigungsprozess

Projekte zielen auf Veränderungen im Unternehmen und binden begrenzte Ressourcen von Mitarbeitern und finanziellen Mittel. Eine Genehmigung, die sicherstellt, dass die für das Unternehmen wichtigen Projekte mit Vorrang durchgeführt werden können, ist notwendig. Für eine Genehmigung sind Informationen über das geplante Projekt zu erstellen und zu dokumentieren. Die Entscheidung selbst kann hierauf aufbauend in einem Portfoliomanagementgremium fallen. Hilfreich ist ein vorgegebener Standard an Dokumenten und klar definierten Inhalten, die in jedem Fall vorzulegen sind. Die Informationstiefe kann bei Projekten mit geringer Auswirkung auf das Unternehmen flach gehalten werden. In jedem Fall muss der Nutzen für das Unternehmen transparent sein. Gibt es bereits genehmigte, übergeordnete Projekte oder Programme, denen das Projekt zugeordnet werden kann? Grundsatzdiskussionen können entfallen, entschieden werden muss aber immer über die notwendige Ressourcenfreigabe. Diese könnte in Engpassituationen wichtige Projekte gefährden und muss daher bewusst und verbindlich getroffen werden.

Die Entscheidungshierarchien müssen flach gehalten werden. Hier hilft die Unterscheidung in „Entscheider“ und „zu informieren“.

Fazit: Nur die wirklich notwendigen Informationen einfordern. Wichtig ist der Gesamtüberblick über die geplanten und genehmigten Projekte im Unternehmen. Widerstände bei der Umsetzung sind hier zu erwarten und müssen in offener Kommunikation ausgeglichen werden, um zu gemeinsam getragenen Vereinfachungen zu gelangen.

4.2 Projektmethodik

Eine Projektmethodik legt allgemein fest, wie Projekte zu organisieren und durchzuführen sind. Definiert wird **was** zu tun ist. Nicht definiert wird in der Regel **wie** etwas im Detail zu tun ist. Dies verbleibt in der Verantwortung des Anwenders. So weist z.B. PRINCE2 konkret auf Möglichkeiten des „Tailorings“ hin [Og09]. Beispiele sind das

Zusammenlegen von Rollen in einer Person, Zusammenlegen und Vereinfachen von Berichten, vereinfachte Lenkungsprozesse, Vereinfachungen von Berichtswegen und Dokumentationen oder weniger formale Kommunikation.

Für Projektmanagement nach PMI gilt ebenfalls die Möglichkeit einer differenzierten Anwendung z.B. bei der Auswahl und dem Einsatz von Projekt Management Prozessen [Pm13].

Ein eher komplexes Tailoring nach Projektstrukturmerkmalen ist Bestandteil des V-Modell XT [Vm12].

Es gilt immer der Grundsatz nur das Notwendige zu machen, nicht das Mögliche.

4.3 Projektcontrolling / Berichtswesen

Hier kann man sich richtig austoben, oder den Aufwand auf das wirtschaftlich sinnvolle Maß zur Steuerung eines Projektes beschränken. Nicht was an Informationen und Berichten machbar ist zählt, sondern die richtige Information zum richtigen Zeitpunkt. Wichtig ist weiterhin der Inhalt und nicht die Form der Darstellung.

Ausgangspunkt der Diskussion von Vereinfachungen sind die Anforderungen, die eine wirksame Projektsteuerung zur Erreichung des Projektergebnisses an die Bereitstellung von Informationen und Berichten stellt. Alle zusätzlichen Anforderungen müssen nachvollziehbar begründet werden. Dieser Verhandlungsprozess wird auf eine Reihe von Widerständen treffen, da gewohnte Formate und Inhalte wegfallen können.

Ein genereller Ansatzpunkt für Vereinfachungen liegt in der Genauigkeit von Projektplanungen und davon abgeleiteten Aufwandsschätzungen. Die Möglichkeiten und Notwendigkeiten von detaillierten und exakten Planungen werden gerne überschätzt. Die Detailliertheit der Projektplanung kann abhängig von Merkmalen wie Risiko, Komplexität oder Bedeutung für das Unternehmen flexibel gehandhabt werden. Bei üblichem Informationsstand zum Projektstart sind allenfalls Schätzbereiche für den geplanten Aufwand möglich. Exaktheit ist hier weitgehend eine Illusion. Die Annäherung an Exaktheit ist mit zunehmendem Mehraufwand verbunden, dem kaum echter Nutzen, speziell hinsichtlich der abzuliefernden Projektergebnisse, gegenüber steht. Ein Trend zur Nutzung von Bandbreiten an Stelle exakter Einzelwerte ist derzeit generell im Controlling zu erkennen. Hinzu kommen Effekte aus Änderungen im Projektumfang, die nicht unbedingt in den Aufwandsplanungen nachgezogen werden.

Bei kleinen Projekten werden die Verantwortung und Durchführung des Projektcontrollings überwiegend beim Projektleiter liegen. Anforderungen des Unternehmenscontrollings müssen hier klar definiert und kommuniziert sein.

5 Implementierungsschritte

Für die Einführung von Differenzierungen von Standards für Projekte bieten sich folgende Schritte an:

1. Analyse des bestehenden und geplanten Projektportfolios. Anzahl, Umfang und Struktur der bestehenden und geplanten Projekte.
2. Eine Analyse der bestehenden Anforderungen kann parallel zu (1) erfolgen. Hierzu gehört es auch nachzufassen, inwieweit bestehende Regelungen kommuniziert bzw. bekannt sind.
3. Anforderungen der Anfordernden der Standards und Methoden sind zu hinterfragen. Anforderungen müssen dabei u.a. den entstehenden Aufwand konkret rechtfertigen. Der Einsatz technischer Hilfsmittel, z.B. Genehmigungsworkflows mit hinterlegten Regeln, sollte geprüft werden.
4. Die Verantwortung für die projektbezogene Standards und Methoden muss festgelegt sein.
5. Aus den Vorarbeiten ist ein Umsetzungsvorschlag zu erarbeiten, durchzusetzen und umzusetzen.
6. Kommunikation der Regeln an alle Beteiligten und Betroffenen.

Projektorientierte Unternehmen werden versuchen, über Projektoffices, Projekt Management Offices und Portfoliomanagementgremien flankierende Organisationsformen zu etablieren, um den Fokus in den einzelnen Projekten verstärkt auf die Erstellung der Projektleistung zu richten und nicht auf die Administration.

6 Fazit

Der differenzierte Einsatz von Standards und Methoden in Projekten ist eine notwendige Maßnahme. Differenzierungen sind beispielsweise in PM-Methoden vorgegeben. Sie müssen aber auch durchgesetzt und umgesetzt werden.

Eine Reihe von Vorgehensschritten wird hier vorgeschlagen:

- Festlegen, was im Unternehmen als Projekt behandelt wird.
- Klassifizierungsstufen für Projekte festlegen.
- Einen flachen und transparenten Genehmigungsprozess einführen.
- Freiheitsgrade von Standards nutzen. Beispiel: Projektmethoden.
- Das Berichtswesen und die Kommunikationswege durchforsten.

Der Aufwand, der hier einmal betrieben werden muss, wird sich schnell auszahlen.

Schließen möchte ich mit einem meiner Lieblingszitate: *"Perfektion ist nicht dann erreicht, wenn nichts mehr hinzuzufügen ist, sondern wenn man nichts mehr wegnehmen kann"* (Antoine de Saint Exupery).

Literatur

- [Bm11] Best Management Practice: ITIL Service Operation. Second edition 2011. S. 227.
- [Og09] OGC: Managing Successful Projects with PRINCE2. Fifth edition 2009. S. 213 ff.
- [Pm13] PMI: A guide to the management body of knowledge. Fifth edition 2013.
- [Pm14] PMI: Pulse of the Profession. The High Cost of Low Performance. February 2014.
- [Vm12] V-Modell XT Gesamt 1.4, deutsch, PDF. IABG 2012. S. 81 ff.

Markup-basiertes Spezifikations- und Anforderungsmanagement in agilen Softwareprojekten

Roman Roelofsen¹, Stephan Wilczek²

¹Rodalo GmbH
Kaiserstraße 108
53721 Siegburg
rr@rodalo.com

²Hochschule der Medien
Nobelstr. 10
70569 Stuttgart
wilczek@hdm-stuttgart.de

Abstract: Agile Softwareprojekte benötigen ein spezielles Management von Anforderungen und Spezifikation. Der Artikel versucht, die notwendige Sicht darauf zu beschreiben, daraus Anforderungen an eine Werkzeugunterstützung abzuleiten und stellt mit *agileSpecs* eine experimentelle Plattform vor, die mit Hilfe einer Markup-basierten Sprache versucht, einem agilen Anforderungs- und Spezifikationsmanagement gerecht zu werden.

1 Spezifikation als kritischer Erfolgsfaktor

Im Hinblick auf das Scheitern von IT-Projekten verweisen neben dem in der Praxis immer wieder herangezogenen CHAOS-Report der Standish Group [SG94] auch andere Untersuchungen [MM11] insbesondere auf unvollständige Anforderungen und Spezifikationen, sich im Projektablauf ändernde Anforderungen und Spezifikationen sowie unzureichendes Input und fehlende Kommunikation mit den verschiedenen Stakeholdern im Projekt. Schwaber bringt es mit dem Aphorismus „garbage in, garbage out“ auf den Punkt und betrachtet kritisch die Definition von Anforderungen, den Change Management Prozess von Anforderungen und Werkzeuge sowie Techniken für das Management von Anforderungen [SLD06].

Agile Methoden und “Continuous Delivery” haben die Entwicklung, den Test und die Auslieferung von Software verbessert. Dennoch wird gerade das Management von Anforderungen und Spezifikationen weiterhin kritisch gesehen [BMW14]. Ansätze wie die Erfassung und Verwaltung von User Stories, Use Cases bzw. Softwaretests oder die Verlagerung in das UI Prototyping bzw. das Rapid Application Development (RAD) oder das Joint Application Development (JAD) bieten Alternativen in agilen Projekten.

2 Spezifikation in agilen Softwareprojekten

Agile Softwareentwicklung zeichnet sich unter anderem durch die Fokussierung auf die beteiligten Individuen, Zusammenarbeit mit dem Kunden und die Fähigkeit aus, schnell und flexibel auf Veränderungen zu reagieren. Der funktionierenden Software wird dabei ein höherer Stellenwert beigemessen als einer umfangreichen Dokumentation, die aber dennoch als wichtig¹ erachtet wird [Be01]. Die Erhebung und Verwaltung von Softwareanforderungen in agilen Projekten ist geprägt durch eine Vielzahl unterschiedlicher Techniken und Werkzeuge. Wird die initiale Definition der formalen und nicht-formalen Anforderungen an die Software zum Projektbeginn noch in einem Dokument spezifiziert und bspw. als Grundlage für Verhandlungen mit Dienstleistern genutzt, so werden die unterschiedlichen Anforderungen im Projektverlauf verfeinert, in unterschiedliche Werkzeuge übertragen und in der Regel an mehreren Stellen verwaltet.

2.1 Arbeit in Arbeitspaketen

Agile Vorgehensmodelle stellen den Entwickler in den Mittelpunkt. Dabei wird der Prozess konsequenterweise auf die üblichen technischen Tätigkeiten des Entwicklers ausgerichtet. So werden die Anforderungen mit Hilfe von Techniken wie bspw. User Stories, Use Cases oder UI Prototyping erhoben und anschließend in für die Entwickler geeignete Arbeitspakete zerlegt, die in iterativen Schritten bearbeitet werden. Die einzelnen Arbeitspakete lassen sich in der Regel einer bestimmten Kategorie zuordnen, wobei die Kategorien von der Umsetzung einer User Story über einen Change Request bis hin zu einem Softwaretest oder der Behebung eines Softwarefehlers reichen. Die einzelnen Schritte und Phasen des Vorgehensmodells unterscheiden allerdings nicht zwischen den unterschiedlichen Kategorien. Grund dafür ist, dass innerhalb einer Iteration die stetige Weiterentwicklung bzw. Verbesserung der Software im Vordergrund steht. Ob es sich dabei um die Behebung eines Fehlers (Bug Report) oder um die Implementierung einer neuen Anforderung handelt, spielt aus Sicht des Entwicklers nur eine untergeordnete Rolle – eine funktionierende Software gilt als das messbare Ergebnis jeglicher Veränderungen innerhalb einer Iteration. Der Soll-Zustand der Software ergibt sich daher implizit aus der iterativen Verwertung (Beurteilung und Umsetzung) der Arbeitspakete, die während des Entwicklungszeitraums umgesetzt werden. Dabei wirken sich die Arbeitspakete unterschiedlich stark auf den Soll-Zustand aus. So erweitern beispielsweise User Stories den Soll-Zustand, wohingegen Bug Reports den Ist-Zustand an den Soll-Zustand angleichen sollen. Im letzten Fall könnte sowohl eine falsch *formulierte* User Story als auch eine falsch *umgesetzte* User Story der Grund sein.

2.2 Werkzeugnutzung und Medienbrüche

User Stories müssen strukturiert und mit diversen Meta-Daten erfasst werden: Datum, Priorität, Zustand („Entwurf“, „Abgenommen“, ...). Dies widerspricht jedoch schnell dem Anspruch, die Spezifikation mit minimalen Aufwand pflegen zu können und sie

¹ “That is, while there is value in the items on the right, we value the items on the left more.” [Be01] - wobei die „laufende Software“ links und eine „umfassende Dokumentation“ rechts stehen.

primär als textuelles Dokument zu behandeln, in dem User Stories und fachliches Wissen gesammelt werden. Ein Entwickler würde ab einem gewissen Zustand im Prozess eine User Story in ein Ticket-System überführen, um beispielsweise Diskussionen und Verantwortlichkeiten festzuhalten. Bug Reports werden von vornherein in einem Ticket-System erfasst. Ein Bug Report kann, wie oben beschrieben, Auswirkungen nur auf die Software oder auf die Software und die Spezifikation haben.

Bei der Verwaltung von Softwareanforderungen und Spezifikationen finden sich in der Praxis so Text- oder Tabellenkalkulations-Dokumente, die mit verschiedenen Methoden versioniert werden. Gerade in kleineren Projekten ist es in der Praxis üblich, dass der Kunde User Stories und Meta-Daten in einer Tabellenkalkulation und den Text der User Stories sowie fachliches Wissen in Text-Dokumenten erfasst. Dazu kommen Architekturskizzen, die mit Grafik- oder Präsentationsprogrammen erstellt werden. Der Entwickler kopiert den Text der User Stories und arbeitet innerhalb des agilen Prozesses mit einem Ticketsystem. Jeder arbeitet in "seinem" System, was eine manuelle und fehleranfällige Synchronisation nötig macht.

Selbst wenn auf eine Spezifikation verzichtet wird, sind Medienbrüche und Systemgrenzen vorhanden: Ticket-Systeme sammeln User Stories und Bug-Reports, der Programmierer setzt diese in Source Code um, fachliches Wissen wird ergänzend vom Kunden in Textform geliefert und Change-Logs beinhalten Zusammenfassungen. Es ist daher besonders die Angst vor weiterer Redundanz (unterschiedliche Zustände, doppelte Arbeit) die das Erstellen einer Spezifikation erschwert. Die Arbeit mit verbreiteten Tabellenkalkulations- und Textverarbeitungsdokumenten lässt sich schwer mit den derzeit üblichen Techniken wie einem zentralen Ticket- und Source Code Management-System verbinden.

2.3 Agile Spezifikation

In einem agilen Projekt muss die Spezifikation nicht zwangsläufig jederzeit den Zielzustand erfassen, sondern kann dem aktuellen Entwicklungsstand der Software entsprechen. Die Spezifikation kann in drei zeitliche Kategorien eingeordnet werden:

1. Vorgelagert: Vor der Iteration werden die als nächstes zu implementierenden Anforderungen spezifiziert
2. Begleitend: Die Spezifikation wird parallel zur Softwareentwicklung weiterentwickelt
3. Nachgelagert: Nach der Iteration werden die durchgeführten Arbeiten konsolidiert und festgehalten

In der Praxis wird in allen drei Kategorien in unterschiedlichem Ausmaß gearbeitet. Besonders der erste Zustand hat bei der Kommunikation mit dem Kunden einen großen Wert. Ziel ist es, unnötige Entwicklungen zu vermeiden und fachliche Anforderungen zuerst schriftlich festzuhalten. Die dritte Kategorie dient zur Validierung und Abnahme

der umgesetzten Anforderungen. Dies geschieht durchaus unter Anwendung der agilen Methoden, da sich auch eine Spezifikation iterativ und stetig entwickeln lässt.

Im Hinblick auf das Spezifikations- und Anforderungsmanagement in agilen Softwareprojekten liegt der Gedanke daher nahe, allen Beteiligten im Prozess die Möglichkeit zu geben, Anforderungen und Spezifikationen strukturiert aufnehmen zu können, sie jederzeit einzusehen und mit minimalem Aufwand auch zu editieren bzw. zu administrieren.

2.4 Pragmatische Betrachtung der Spezifikation

Eine Spezifikation in einem agilen Softwareprojekt sollte nicht als "umfassende Dokumentation" verstanden werden, sondern als Zustandsbeschreibung der Software, Sammlung der User Stories und Dokumentation von fachlichem Wissen. In der Praxis zeigt sich jedoch, dass die Softwareentwicklung oft nur unzureichend mit der Erstellung der Spezifikation kombiniert wird. Das mag technische (siehe der Hinweis auf die Medienbrüche oben), aber auch soziale Gründe haben. Insbesondere sehen viele Entwickler keinen direktgp Mehrwert in einer ausführlichen'Dokumentation und empfinden ihre Ctbeit nach der erfolgreichen (softwaretechnischen) Implementierung als erledigt.

Das Problem lässt sich dadurch mindern, dass die Spezifikation eben nicht als "umfassende Dokumentation" betrachtet wird, sondern, wie oben beschrieben, primär als Sammlung der User Stories und als nützliches Hilfsmittel bei der Kommunikation mit dem Auftraggeber und anderen Stakeholdern. Zusätzlich sollte die Spezifikation ein integraler Bestandteil im Arbeitsprozess des Entwicklers sein. Ein „copy & paste“ zwischen den Systemen sollte auf jeden Fall verhindert werden und die Spezifikation selbst zum aktiv genutzten Werkzeug bzw. Material des Entwicklers werden.

2.5 Ableitung eines aktuellen „Status Quo“

Unabhängig vom Entwicklungsprozess benötigen Softwareprojekte eine geeignete Spezifikation des Zielsystems bzw. eine Möglichkeit, den „Status Quo“ abzufragen. Mag sich die Spezifikation über die Laufzeit des agilen Projekts zwar stetig weiterentwickeln und vervollständigen, benötigen die Beteiligten in den meisten Fällen dennoch ein aktuelles Gesamtbild der bereits implementierten Software und des angestrebten Zielsystems. Mit steigender Anzahl der Iterationen werden mehr und mehr Arbeitspakete umgesetzt. Aber auch Modifikationen, ein Hinzufügen oder ein ersatzloses Streichen von Arbeitspaketen sind im agilen Prozess jederzeit möglich. Die aktuelle Gesamtspezifikation des Zielsystems ergibt sich aus einer Analyse aller offenen und geschlossenen Arbeitspakete in ihrer aktuellen, seit dem Projektstart aber möglicherweise mehrfach modifizierten Form. Je nach Prozess muss hier auf eine Reihe unterschiedlicher Quellen zugegriffen werden. Da die Arbeitspakete, wie oben geschildert, in unterschiedlichen Systemen (Ticketsystem, Bug-Tracking-System, Test-Werkzeug etc.) verwaltet werden können, ist die Herleitung einer aktuellen sowie strukturierten Gesamtspezifikation eine komplexe und schwierige Aufgabe.

3 Anforderungen an eine Spezifikationsplattform

Eine Plattform zur Erstellung der Spezifikation muss sich in eine Umgebung bestehend aus Ticket- und Source Code-Verwaltungssystem einfügen lassen und muss als sinnvolle Ergänzung wahrgenommen werden, die Redundanz vermeidet und die Kommunikation mit den beteiligten Stakeholdern unterstützt. Die (unorganisierte) Verwendung von Textverarbeitungs- und Tabellenkalkulationsprogrammen kann diesen Anspruch nicht erfüllen.

Mit dem Fokus, die Spezifikation als Kooperationsmaterial aller beteiligten Stakeholder zu sehen, folgt man dem ersten Grundsatz des agilen Manifests. Entsprechend sollte die Plattform (a) einen technischen Fokus haben, um als dezidierte Lösung wahrgenommen werden zu können, und (b) eine rein semantische Erstellung der Spezifikation ermöglichen.

Vergleichbare Ansprüche finden sich bei Ticket- und Wiki-Systemen, wo Markup-basierte Texteingaben mittlerweile zum Standard gehören. Eine Plattform zur Spezifikationserstellung sollte diesem Ansatz folgen, um den Programmierer eine einfache und zeitgemäße Spezifikationspflege zu ermöglichen. Die Plattform muss die Markup-basierte Eingabe jedoch so aufbereiten, dass die Spezifikation professionell zur Kommunikation mit dem Kunden genutzt werden kann. Für die Typisierung und Verwaltung einzelner Anforderungen in einem größeren Dokument bieten sich natürlich XML-basierte Formate wie DocBook [Oa14] oder DITA [DPS05] an. Hier wird es allerdings selbst mit geeigneter Softwareunterstützung für die Mehrzahl der Beteiligten im Projekt schwierig, ein auf dieser Basis erstelltes Dokument zu bearbeiten. Ein hoher Schulungsaufwand steht hier zunächst der Agilität im Weg.

Eine weitere Anforderung an die Plattform besteht darin, aufbauend auf der rein textuellen und unformatierten (Markup) Eingabe der Spezifikation die Verwaltung der Meta-Daten zu ermöglichen:

- Anforderungen müssen verfolgbar sein
- Anforderungen müssen organisierbar sein
- Änderungen am Inhalt müssen nachvollziehbar sein
- Verantwortlichkeiten müssen definierbar sein

In Softwareprojekten wird, unabhängig davon, ob eine Spezifikation gepflegt wird, ein Ticket-System zur Erfassung der Anforderungen verwendet. Dies kann auf zwei Arten geschehen:

(1) Explizit: Eine Anforderung wird, bevor sie implementiert wird, als Ticket/User Story im Ticket-System erfasst. Falls eine Spezifikation oder Auflistung der User Stories existiert, werden die Beschreibungen per "copy & paste" übernommen.

(2) Implizit: Falls der Ist-Zustand der Software nicht dem Soll-Zustand entspricht, wird ein Ticket erstellt, das die nötigen Änderungen, die Gründe dafür und das notwendige Wissen erfasst.

Das Problem mit diesem Ansatz ist, dass Tickets in der Regel nur einen transitorischen Charakter haben. Fachliches Wissen wird z.B. oft nur als Kommentar zu einem Ticket ergänzt. Es ist nicht praktikabel, aus der Summe aller Tickets den zu einem bestimmten Zeitpunkt definierten Soll-Zustand einer Software abzuleiten. Dafür wären Text-Dokumente notwendig, was jedoch eine manuelle und damit als redundant empfundene Pflege voraussetzen würde.

Tickets werden primär auf den Software-Stand bezogen und nicht auf die Spezifikation. Dabei sollte sich jedes Ticket primär auf die Spezifikation beziehen:

- Eine neue User-Story sollte in der Spezifikation, sortiert nach Kapiteln und aus Kundensicht organisiert, angelegt werden.
- Bug-Tickets bedeuten letztlich eine falsch dokumentierte oder falsch implementierte User-Story, die wiederum in der Spezifikation stehen sollte.
- Tickets zur Verbesserung der Software sollten im ersten Schritt als fachliche Ergänzung zu einer User-Story verstanden werden, die zuerst in der Spezifikation erfasst werden sollte.

Es darf kein Ticket erfasst werden, welches sich nicht auf eine fachliche Anforderung zurückführen lässt. Dies bedeutet keinesfalls, dass die Spezifikation *a priori* erstellt werden muss. Neue Anforderungen können bei Bedarf stichpunktartig erfasst werden, sobald ein fehlendes oder falsches Verhalten der Software dies notwendig macht. Die Spezifikation entsteht so automatisch als nebenläufiges Artefakt.

Die Lösung muss also darin bestehen, fachliches Wissen (sei es eine neue Anforderung oder eine Korrektur) ausnahmslos in einem Text-Dokument, nämlich der Spezifikation, festzuhalten. Gleichzeitig muss aber die Pflege der Meta-Daten, wie in einem Ticket-System, möglich sein. Basierend auf den gemeinsam gepflegten Daten und Meta-Daten sollte es dann möglich sein, für die verschiedenen Stakeholder geeignete Berichte und Analysen zu generieren. Insbesondere eine jederzeit aktuelle Spezifikation für den Auftraggeber (bspw. in Form eines formatierten PDF-Dokuments) oder eine detaillierte Historie für den Projektleiter (bspw. in Form einer Liste) wären wünschenswert.

4 Experimentelle Plattform *agileSpecs*

Basierend auf dem hier beschriebenen Lösungsansatz, agile Softwareprojekte und Spezifikationen zu verbinden, wurde die Plattform *agileSpecs*² entwickelt. Die oben genannten Anforderungen an eine Spezifikationsplattform lassen sich wie folgt zusammenfassen:

- Zentrale Plattform für alle Beteiligten im Prozess
- Leicht zu erlernende Syntax
- Trennung von Text und Anforderungen/Spezifikationen
- Bereitstellung der Meta-Daten (Versionierung, Bearbeitungshistorie etc.)
- Verwaltung von Zuständen der Anforderungen, um z.B. neue Anforderungen zu markieren oder Fehler beim Ist-Zustand zu melden
- Verfolgung der Anforderungstexte und unterstützende Benachrichtigungen und Warnungen bei Änderungen
- Erzeugung spezifischer Berichte für das Gesamtprojekt
- Erzeugung einer grafisch ansprechenden Form der Spezifikation

agileSpecs bietet eine **zentrale Plattform**, die als SaaS-Lösung im Internet genutzt oder vom Projekt im Intranet betrieben werden kann.

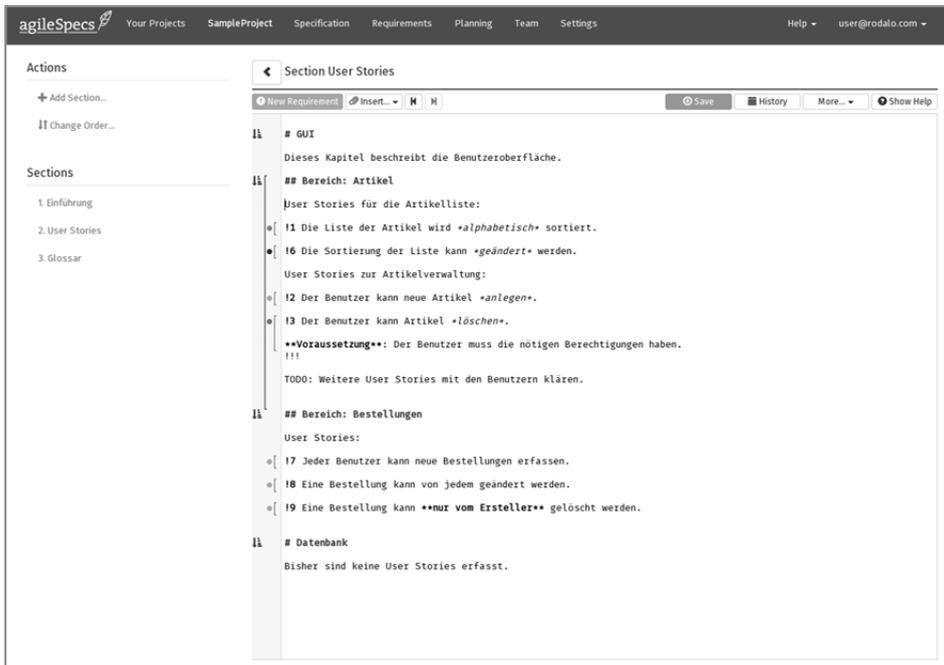


Abbildung 1: Markdown bei der Beschreibung von Anforderungen

² <http://www.agileSpecs.com>

Der Projekt-Administrator vergibt den Beteiligten passwortgeschützte Accounts, die er bei Bedarf mit anderen Single-Sign-On-Lösungen verbinden bzw. synchronisieren kann. Die Lösung benötigt lediglich einen aktuellen Browser, die Installation lokaler Software ist nicht notwendig.

Als Auszeichnungssprache wird Markdown³ verwendet. Die **einfache Syntax** kann schnell erlernt werden. Markdown ist bei Wiki-Systemen sehr verbreitet und Plattformen wie beispielsweise GitHub (github.com) setzen Markdown für Tickets, Wiki und sogar Commit-Messages ein. Markdown verfügt über keine eigene Unterstützung für User Stories/Anforderungen, so dass die Markdown-Syntax erweitert wurde, um Anforderungen markieren zu können. Das System vergibt automatisch fortlaufende und stabile IDs für die Anforderungen, die keine weitere Semantik haben (vgl. Abb. 1).

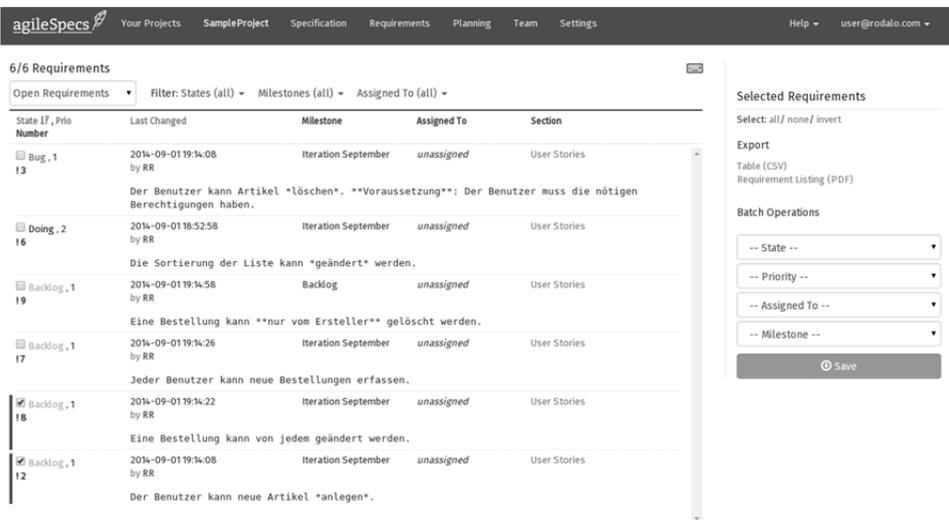


Abbildung 2: Filterung von Items aus dem Text

Der einzig notwendige Schritt, um mit *agileSpecs* eine Spezifikation zu erstellen, ist die Pflege des Textes mit Markdown. Dies kann begleitend zur Entwicklung zuerst stichpunktartig und bei Bedarf dann detailliert erfolgen. Basierend auf der Kapitelstruktur von Markdown und den Markierungen für Anforderungen werden automatisch alle Anforderungen erkannt und deren Textbeschreibung, Autor und Zustand historisiert. Weiterhin können, wie in Ticket-Systemen üblich, die Anforderungen priorisiert und Benutzern zugeordnet werden. Anforderungen können unabhängig vom Text der Spezifikation kommentiert werden. Für jedes Projekt lassen sich beliebige Zustände für Anforderungen (z.B. *Neu*, *in Arbeit*, *Fehler*, *Implementiert*) definieren. Aufbauend auf der textuellen Ansicht der Spezifikation und den darin markierten Anforderungen können alle Anforderungen aufgelistet und gefiltert werden (vgl. Abb. 2). Das Schreiben der Spezifikation kann komplett mit der Pflege der Anforderungen verwoben werden, wodurch die zusätzliche Verwendung eines Ticket-Systems obsolet wird.

³ Für eine Übersicht dazu siehe: <http://de.wikipedia.org/wiki/Markdown>

Obwohl die Spezifikation mit Markdown bereits formatiert und grafisch aufbereitet wird, sind zusätzliche Ansichten sinnvoll, die z.B. vom Auftraggeber genutzt werden bzw. den Fokus auf das Lesen und Verstehen der Spezifikation legen. *agileSpecs* stellt hier verschiedene Ansichten in Tabellenform sowie nach Bedarf formatierte PDF-Dokumente (vgl. Abb. 3) zur Verfügung.

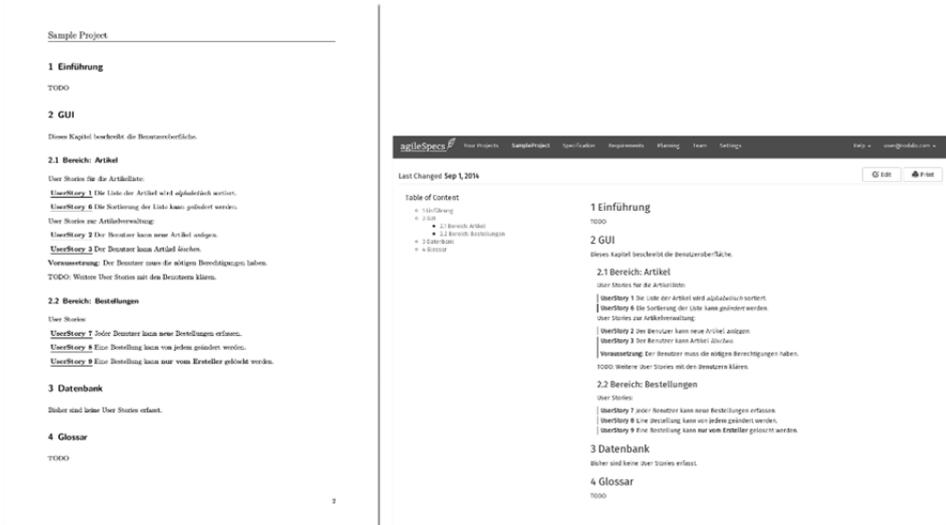


Abbildung 3: Beispiel einer Ausgabe als PDF-Dokument (links) und als elektronischer Report (rechts)

Bei der Kommunikation mit dem Kunden sind Berichte bzw. Zusammenfassungen sinnvoll, mit denen beispielsweise Iterationen besprochen werden können. Ausgewählte Anforderungen können dafür als Excel-Datei exportiert oder zusammengefasst in einem PDF aufgelistet werden.

Literaturverzeichnis

- [Be01] Beck, K. et. al.: Manifesto for Agile Software Development (2001). URL: <http://www.agilemanifesto.org>.
- [BMW14] Bittner, K.; Mines, C.; Whittaker, D.: Brief: Software Requirements Practices Are Ripe For Disruption. Forrester, Forrester Research, 25.04.2014. URL: <http://www.forrester.com/Brief+Software+Requirements+Practices+Are+Ripe+For+Disruption/fulltext/-/E-RES115852>.
- [DPS05] Day, D.; Priestley, M.; Schell, D.: Introduction to the Darwin Information Typing Architecture. URL: <http://www.ibm.com/developerworks/xml/library/x-dita1/?ca=dgr-wikiaDita1>. In: IBM Developerworks (2005) - abgerufen am 20.08.2014.
- [MM11] McLeod, L.; MacDonell, S. G.: Factors that affect Software Systems Development Project Outcomes: A Survey of Research. In: ACM Computing Surveys, Bd. 43 (2011), S. 24–55.

- [Oa14] OASIS - Organization for the Advancement of Structured Information Standards: DocBook Technical Committee Document Repository. URL: <http://www.oasis-open.org/docbook/> - abgerufen am 20.08.2014.
- [SG94] The Standish Group International: The Chaos Report (1994). The Standish Group International, Inc., West Yarmouth, MA (USA).
- [SLD06] Schwaber, C.; Leganza, G.; Daniels, M.: The Root Of The Problem: Poor Requirements. Forrester Research – Trends (2006).

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. Mühlhng, 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 1st 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 3rd 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 1st International Working Conference on Business Process and Services Computing BPSC 2007
- P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and management The 4th 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.)
9th 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.)
10th 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.)
3rd 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.)
9th International Conference on Innovative Internet Community Systems
I²CS 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 3rd 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.)
10th International Conference on Innovative Internet Community Systems (I²CS) – 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.)
4th 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.)
6th 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.) 11th International Conference on Innovative Internet Community Systems (I²CS)
- 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 11th 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.) 12th International Conference on Innovative Internet Community Systems (I²CS 2012)
- P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.) 5th 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 13th 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 Sassi,
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

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

