

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 co-operation with GI and to publish the annual GI Award dissertation.

Broken down into

• seminars

• proceedings

• dissertations

• the matics

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 978-3-88579-618-3

This volume contains papers from the Vorgehensmodelle 2013 Conference on software development process models held in Lörrach October 9 to 10, 2013. Vorgehensmodelle 2013 is the 20th event in a conference series focusing on a broad range of topics investigating software development process models from a variety of perspectives. With its emphasis on lively discussions and cross-fertilization of academia and industry, it provides a valuable platform to advance the state of the art in topics such as tailoring, plan-driven and agile approaches, industry-sector-specific approaches, specific process modules, standards and tools.



E. Hanser, M. Mikusz, M. Fazal-Baqaie (Hrsg.): Vorgehensmodelle 2013

224

GI-Edition

Lecture Notes in Informatics

**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**

Proceedings





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.**

**9. und 10. Oktober 2013
in Lörrach**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-224

ISBN 978-3-88579-618-3

ISSN 1617-5468

Volume Editors

Prof. Dr. Eckhart Hanser

Duale Hochschule Baden-Württemberg

Email: hanser@dhbw-loerrach.de

Dr. Martin Mikusz

Universität Stuttgart, GSaME

Email: martin.mikusz@gsame.uni-stuttgart.de

FOM Hochschule für Oekonomie und Management Stuttgart

Masud Fazal-Baqaie

Universität Paderborn

s-lab - Software Quality Lab

Email: mfazal-baqaie@s-lab.upb.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 2013

printed by Köllen Druck+Verlag GmbH, Bonn

Geleitwort

Die Fachgruppe Vorgehensmodelle (WI-VM) ist eine der drei Fachgruppen des Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik der GI (<http://fa-wi-maw.gi.de/>). Die anderen beiden Fachgruppen sind Projektmanagement (WI-PM) und Produktmanagement (WI-PrdM). Der Fachausschuss hat das Ziel, die Managementaspekte von Anwendungssystemen hervorzuheben und fokussiert auf deren organisatorische Einbindung zu achten. Hierzu integriert er die drei selbständig agierenden Fachgruppen.

Ich freue mich, dass wieder ein so attraktiver Workshop der Fachgruppe WI-VM stattfinden kann, wie es das Programm zeigt. Es handelt sich um ein Jubiläum, denn der erste Workshop fand 1994 in München statt, also vor 20 Jahren. Es ist schön, dass Sie einen der Organisatoren von 1994, Herrn Günther Müller-Luschnat, für eine Keynote gewinnen konnten.

Man sollte meinen, dass in 20 Jahren ein Thema wie Vorgehensmodelle „abgegrast“ sein sollte, weil nicht relevantes Neues mehr gefunden wird. Wie jedoch schon das Inhaltsverzeichnis dieses Bandes zeigt, ist dem keineswegs so. Vorgehensmodelle leben nicht von theoretischer Forschung, sondern von der Reflexion praktischer Projekterfahrungen, die sachlich und institutionell vielfältig ist. Die Autorenprofile zeigen, dass die Beiträge des Jubiläums-Workshops überwiegend von erfahrenen Verfassern mit breitem Hintergrund stammen. Dies ist für die Teilnehmer der Tagung und die Leser des Tagungsbandes ein sehr positiver Aspekt, der von der Fachgruppe in den vergangenen 19 Jahren sorgfältig gepflegt und ausgebaut wurde. Allen Mitgliedern, die sich engagiert haben und weiter engagieren, sei hiermit im Namen der GI gedankt.

Zwei inhaltliche Schwerpunkte des diesjährigen Programms sind erkennbar, die auch für die Zukunft Hoffnung machen. Erstens rückt (endlich!) das Requirements Engineering in den Vordergrund und zweitens wird über agile Vorgehensmodelle auch mit etwas Abstand reflektiert, deren Betrachtung gelegentlich in Dogmen oder Mythen abglitt. Es macht z. B. einen fundamentalen Unterschied, ob ein System für zehn oder für zehntausend Benutzer entwickelt wird oder ob eine evolutionäre Einführung überhaupt möglich ist, wenn etwa die nötige Infrastruktur fehlt. Diese Schwerpunkte machen Hoffnung, dass hier erkennbare verhaltenswissenschaftliche Sichtweisen weiter an Boden gewinnen, ohne dass eine solide Beachtung der Technik unterbleibt.

Ich wünsche der Tagung viel Erfolg, insbesondere der neuartigen Session Future Track und dem Leser dieses Bandes hilfreiche Anregungen.

Münster, im Oktober 2013

Thorsten Spitta

Sprecher des FA WI-MAW der Gesellschaft für Informatik e.V.

Vorwort

Liebe Leser,

der vorliegende Tagungsband enthält die Beiträge der 20. Tagung „Vorgehensmodelle 2013“ der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V. Die Veranstaltungsreihe verfolgt das Ziel, fundierte Ansätze zu Vorgehensmodellen gemeinsam mit Erfahrungen zu deren Anwendung in der Praxis einem Fachpublikum aus Wissenschaftlern und Praktikern vorzustellen. Der intensiven Diskussion der vorgestellten Ansätze sowie dem Erfahrungsaustausch wird dabei traditionell eine besonders hohe Bedeutung zugemessen.

Die Fachgruppe nimmt diese Jubiläumstagung zum Anlass, um unter dem Motto „Vorgehensmodelle – Anspruch und Wirklichkeit“ auf das bisher Erreichte, auf bisher vernachlässigte Themen und Entwicklungen, sowie auf die Herausforderungen für die Zukunft zu blicken. Die Beiträge beleuchten in diesem Sinn Diskrepanzen zwischen Anspruch und Wirklichkeit bei Vorgehensmodellen sowohl aus verschiedenen Perspektiven, als auch mit einem bestimmten thematischen Schwerpunkt.

Die diesjährigen Keynotes beschäftigen sich insbesondere mit Vorgehensmodellen in großen Unternehmen. Herr Christian Schmidkonz, Chief Process Office, LEAN Transformation & Continuous Improvement Services, SAP AG, beleuchtet die spannende Frage, wie Software mit agilen Ansätzen in Großprojekten entwickelt werden kann. Die Keynote von Herrn Peter Dettelmann, Werksleiter Evonik Rheinfelden, erlaubt uns den „Blick über den Tellerrand“: Auch Teams in der chemischen Produktion brauchen ein Vorgehensmodell. Wie unterscheidet es sich von denen in der IT? Wo sind Gemeinsamkeiten? Besonders freut uns, dass wir im Jubiläumsjahr mit Herrn Müller-Luschnat den ersten Sprecher unserer Fachgruppe dafür gewinnen konnten, uns über die Geschichte der Fachgruppe und der Tagung zu berichten.

Das Hauptprogramm der Tagung umfasst neun ausgewählte Beiträge aus Praxis und Wissenschaft, die einen Review-Prozess durchlaufen haben. Wir möchten uns an dieser Stelle ausdrücklich bei den Mitgliedern des Programmkomitees bedanken, die durch ihre Gutachten der eingereichten Beiträge (Annahmequote 47%) erst einen objektiven Bewertungsprozess möglich machten.

Eine GI-Fachgruppe wie unsere Fachgruppe Vorgehensmodelle hat die Aufgabe, sich mit der Zukunft des Fachgebiets zu beschäftigen: Wie geht es weiter mit Vorgehensmodellen? Brauchen wir detaillierte Vorgehensmodelle oder ist es nicht besser, gut ausgebildete Fachleute mit einem gut gefüllten "Vorgehens-Werkzeugkasten" auszustatten? Was nützt ein detailliertes Modell, das keiner im Unternehmen richtig kennt oder gar anwendet? Auf der anderen Seite: Braucht man nicht alleine schon für die Behörden den Nachweis des praktizierten „Good Engineerings“? Hilft vielleicht ein systemischer Ansatz weiter? Welchen Weg soll man einschlagen? Solche und ähnliche Fragen sollen in

der erstmals eingeführten Session „Future Track“ diskutiert werden. Fünf eingeladene Referenten erzählen von ihren Erfahrungen mit Vorgehensmodellen und formulieren jeweils ihre These, welche sie in einem kurzen Vortrag vorstellen. Danach soll die These ausgiebig im Auditorium diskutiert werden. Dieses Feedback ist nicht nur für die Referenten solcher Zukunftsthemen und für unsere Fachgruppe Vorgehensmodelle sehr wichtig, sondern soll auch den Anwesenden neue Denkanstöße liefern. Wir freuen uns auf eine angeregte und kontrovers geführte Diskussion.

Ein großer Dank gilt den Sponsoren Evonik Industries und Endress+Hauser, ohne deren finanzielle Unterstützung eine solche Veranstaltung heute kaum noch möglich ist. Unser Dank gilt auch der Dualen Hochschule Baden-Württemberg (DHBW) Lörrach, die uns ihre modernen Räumlichkeiten für unsere Tagung zur Verfügung stellt. Gerade eine Hochschule, die ihre Studierenden zusammen mit ihren dualen Partnern ausbildet, passt sehr gut zum Gedanken unserer Tagung, im Bereich der Vorgehensmodelle Wissenschaft und Praxis zusammenzubringen. Nicht zuletzt danken wir unseren Kooperationspartnern s-lab - Software Quality Lab, Graduate School of Excellence advanced Manufacturing Engineering (GSaME Univ. Stuttgart) und Softwareforen Leipzig für ihre Unterstützung.

Wir hoffen, dass der vorliegende Tagungsband für Sie neue Erkenntnisse und Anregungen enthält. Auf unserer Website www.vorgehensmodelle.de finden Sie neben den Vortragsfolien der Tagung und Informationen über die Aktivitäten und die Organisation unserer Fachgruppe auch unsere Kontaktadressen – wir freuen uns über Ihre Anregungen und auch über neue Mitglieder. Wir sehen die Fachgruppe als eine Plattform für den Austausch mit anderen Interessierten zum Thema Vorgehensmodelle.

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

Lörrach, Stuttgart, Paderborn, im Oktober 2013

Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie

Herzlich willkommen an der DHBW Lörrach

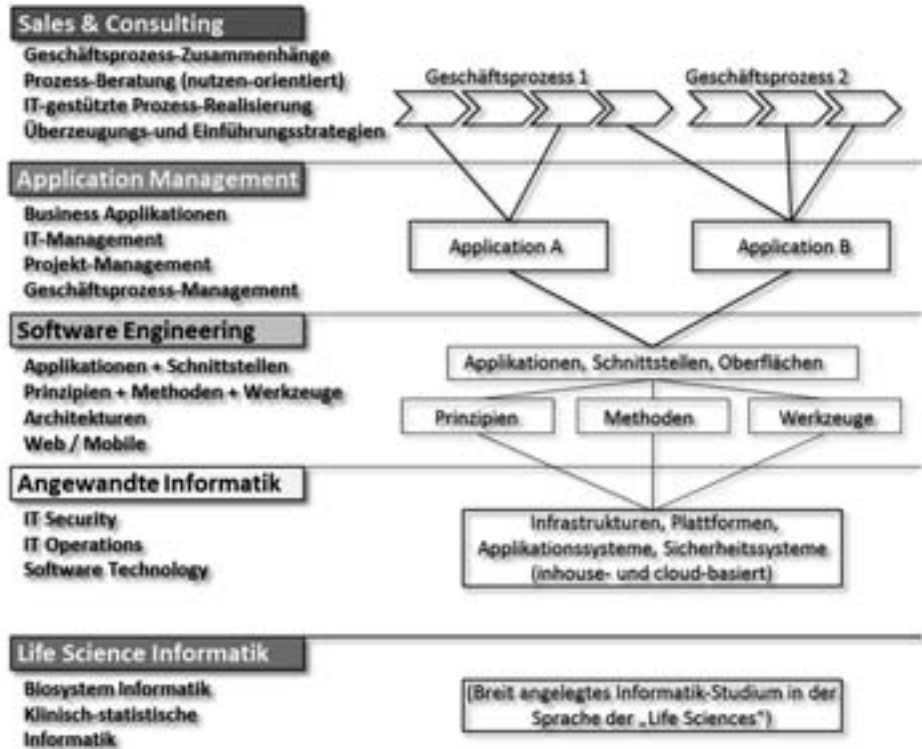
Die Duale Hochschule Baden-Württemberg ist die erste duale, praxisintegrierende Hochschule in Deutschland. Gegründet am 1. März 2009 führt sie das seit knapp 40 Jahren erfolgreiche duale Prinzip der früheren Berufsakademie Baden-Württemberg fort. Bundesweit einzigartig ist die am US-amerikanischen State University-System orientierte Organisationsstruktur der DHBW mit zentraler und dezentraler Ebene. An ihren acht Standorten und vier Außenstellen (Campus) bietet die DHBW in Kooperation mit rund 9.000 ausgewählten Unternehmen und sozialen Einrichtungen eine Vielzahl von national und international akkreditierten Bachelor-Studiengängen in den Bereichen Wirtschaft, Technik und Sozialwesen an. Vor Kurzem wurde das Angebot um ausgewählte berufsintegrierende und berufsbegleitende Masterstudiengänge erweitert. Mit derzeit rund 31.000 Studierenden und über 125.000 Alumni ist die DHBW die größte Hochschule des Landes.

Zentrales Merkmal der DHBW ist das duale Studienkonzept mit wechselnden Theorie- und Praxisphasen sowie die enge Kooperation zwischen der Hochschule und ihren „Dualen Partnern“. Die Unternehmen und sozialen Einrichtungen wählen ihre Studierenden selbst aus, schließen mit ihnen einen dreijährigen Vertrag und bieten während des Studiums eine fortlaufende Vergütung. Den theoretischen Teil des Studiums absolvieren die Studierenden an einem der Standorte der DHBW. Durch den kontinuierlichen Wechsel zwischen Theorie- und Praxisphasen im dreimonatigen Rhythmus erwerben die Studierenden neben fachlichem und methodischem Wissen die im Berufsalltag erforderliche Handlungs- und Sozialkompetenz. Theorie- und Praxisinhalte sind dabei eng aufeinander abgestimmt und beziehen aktuelle Entwicklungen in Wirtschaft, Technik und Gesellschaft in die Lehrpläne mit ein. Die in den Praxisphasen erbrachten Leistungen sind integrativer Bestandteil des Studiums, weshalb sämtliche Studiengänge an der DHBW als Intensivstudiengänge anerkannt und mit 210 ECTS-Punkten bewertet sind.

In Lörrach studieren über 2000 Studierende in insgesamt 19 akkreditierten Bachelor- und vier Masterstudiengängen der Fakultäten Wirtschaft und Technik. Gemeinsam mit der DHBW Lörrach bilden rund 650 Unternehmen aus der Region, dem gesamten Bundesgebiet sowie den Nachbarländern Frankreich und Schweiz aus.

Die informatik-orientierten Studienrichtungen werden an der DHBW Lörrach im „Studienzentrum Informatik / Wirtschaftsinformatik“ gebündelt (siehe <http://inf-wi.dhbw-loerrach.de>). Das Studienzentrum bietet die Studienrichtungen „Sales & Consulting“, „Application Management“, „Software Engineering“, „Angewandte Informatik“ und „Life Science Informatik“ an. Die speziellen Lörracher Inhaltsprofile der einzelnen Studienrichtungen sind nachfolgender Abbildung zu entnehmen.

Die im Zentrum der Tagung liegenden Vorgehensmodelle spielen in allen genannten Studienrichtungen eine bedeutende Rolle und werden ständig in studentischen Projekten weiterentwickelt.



Wir freuen uns sehr, dass die 20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V. an der DHBW Lörrach stattfindet.

Den Teilnehmerinnen und Teilnehmern wünschen wir erfolgreiche Tage mit vielen guten Vorträgen und Gesprächen.

Ein herzliches Dankeschön an alle (Referenten, Organisatoren, Kooperationspartner, Sponsoren, sonstige Unterstützer), die zum Gelingen dieser Jubiläumstagung beigetragen haben.

Jürgen Treffert

Prorektor und Dekan der Fakultät Wirtschaft

Programmkomitee

Vorsitz

Martin Mikusz, Univ. Stuttgart

Masud Fazal-Baqaie, Univ. Paderborn

Mitglieder

Volker Arendt, Bergische Universität Wuppertal

Hubert Biskup, IBM Deutschland GmbH

Jens Calamé, SQS

Gerhard Chroust, J. Kepler University Linz

Martin Engstler, Hochschule der Medien Stuttgart

Detlev Frick, Hochschule Niederrhein

Jan Friedrich, 4Soft GmbH

Andreas Gadatsch, Hochschule Bonn-Rhein Sieg

Thomas Greb, Thomas Greb Consulting

Reinhard Höhn, X-PER.NET Consulting

Patrick Keil, Technische Universität München

Ralf Kneuper, Beratung für Softwarequalitätsmanagement und Prozessverbesserung

Marco Kuhrmann, Technische Universität München

Oliver Linssen, Liantis GmbH & Co. KG

Frank Marschall, T-Systems International GmbH

Alexander Miskiw, GPM e.V. Region Düsseldorf

Günther Müller-Luschnat, Pharmatechnik GmbH

Andreas Rausch, Technische Universität Clausthal

André Schnackenburg, Bundesverwaltungsamt, BVA/BIT

Klaus Stephan, PMI Köln Chapter e.V.

Reinhard Wagner, GPM Deutsche Gesellschaft für Projektmanagement e.V.

Christa Weßel, Organisationsentwicklung & Informationstechnologie

Doris Weßels, Fachhochschule Kiel

Organisatorische Leitung

Eckhart Hanser, Duale Hochschule Baden-Württemberg

Sponsoren



Kooperationspartner



Inhaltsverzeichnis

Teil I – Die Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.

Ralf Kneuper

Zwanzig Jahre GI-Fachgruppe „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“	17
---	----

Teil II – Hauptprogramm

Joachim Schramm, Thomas Ternité, Marco Kuhrmann

Organisationsspezifische Anpassungen von Vorgehensmodellen – Nutzen und Herausforderungen	33
---	----

Thomas Diefenbach, Michael Tonndorf

Ein Konzept zur Harmonisierung der Vorgehensmodelle für die IT-Steuerung Bund	47
---	----

Stefan Hilmer

Verantwortung übernehmen: Das Project Management Office als Process Owner für das unternehmensweite Projektmanagement	65
---	----

Andrea Herrmann, Rüdiger Weißbach

Wer macht eigentlich Requirements Engineering & Management?	77
---	----

Feyyat Kaymaz

Prioritisation and selection of the right business and IT requirements in the software engineering process	87
--	----

Alexander Rachmann, Sven Eselgrimm, Frank Engel

Ausgewählte Problemfelder und Lösungsansatz des Requirements Engineering in der Telekommunikationsbranche	99
---	----

Hans-Peter Korn

Das „agile“ Vorgehen: Neuer Wein in alte Schläuche - oder ein „Déjà-vu“?	109
--	-----

Jörg Hofstetter, Martin Jud

Was fehlt Scrum? – ein beispielhafter Lösungsansatz aus der Hochschulpraxis	133
---	-----

Roland Petrasch, Torben Franzke, Songsak Rongviriyapanich

Einführung von Scrum in einem Software-Entwicklungsprojekt der ContiTech AG	147
---	-----

Teil III – Eingeladene Beiträge der Session „Future Track“

Uwe Henker
Human Factors and SCRUM159

Jane Trümner
Was ist dran an Agilen Mythen?163

Rolf Voller
Warum Projekte scheitern167

Jan Wehinger
Jenseits der Vorgehensmodelle - Projektteams wirksam in Organisationen
verankern, das Viable System Model und MAP171

Sven Niemand, Sven Feja, Sören Witt, Andreas Speck
Vorgehensmodellentwurf für Forschungsprototypen175

Teil I

Die Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.

Zwanzig Jahre GI-Fachgruppe „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“

Ralf Kneuper

Beratung für Softwarequalitätsmanagement und Prozessverbesserung
64295 Darmstadt
ralf@kneuper.de

Abstract: Im Jahr 2013 wird die Fachgruppe „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“ der Gesellschaft für Informatik (GI) 20 Jahre alt. Aus diesem Anlass gibt der vorliegende Beitrag einen Überblick über ihre Entwicklung von der Gründung 1993 bis heute (August 2013). Als Mitglied der Fachgruppenleitung in verschiedenen Funktionen seit 1997 bis heute hat der Autor diese Entwicklung zum großen Teil direkt begleitet. Quellen für diesen Beitrag, wenn auch nicht einzeln aufgeführt, sind die Protokolle und sonstigen Informationen, die auf der Webseite der Fachgruppe (www.vorgehensmodelle.de) zu finden sind sowie eigene Erinnerungen des Autors.¹

Eine Übersicht über die Arbeitskreise der Fachgruppe, der durchgeführten Workshops sowie die Fachgruppenleitungen seit Gründung der Fachgruppe ist im Anhang enthalten.

1993: Gründung der Fachgruppe

Anfang der 1990er Jahre begannen Unternehmen sich intensiver für die Nutzung von Vorgehensmodellen für die Softwareentwicklung zu interessieren. Dafür gab es mehrere Auslöser, u. a. die zu dieser Zeit laufenden Diskussionen über die „Softwarekrise“ (siehe z.B. [We92]), die auch in Unternehmen aufkommende objektorientierte Entwicklung (über die Nutzung objektorientierter Programmiersprachen hinaus, siehe z.B. [Sc94]), sowie die mit dem Aufkommen der Qualitätsmanagementnorm ISO 9001 verbundene wachsende Betrachtung von (Entwicklungs-)Prozessen. 1992 erschien dann eines der ersten Bücher, das sich explizit mit „Modellen der Software-Entwicklung“ beschäftigte [Ch92], Autor war der später auch in der Fachgruppe aktive Gerhard Chroust. Ein Jahr vorher war der „Software-Entwicklungsstandard der Bundeswehr“, später besser bekannt als V-Modell², erschienen, der in der Folge auch für die zivile Bundesverwaltung übernommen wurde.

¹ Eine frühere Version dieses Beitrags ist erschienen unter [Kn12].

² Die Bezeichnung „V-Modell“ wurde damals noch in erster Linie als Abkürzung für „Vorgehensmodelle“ verwendet (siehe z.B. [Ch92, S. 37]). Erst später wurde die Bezeichnung eingeschränkt auf V-förmige Vorgehensmodelle genutzt, mittlerweile ist damit meist das V-Modell des Bundes (heute V-Modell XT, siehe [HH08]) gemeint.

Im Juni 1993 trafen sich daher einige Interessierte in München zur Gründung einer neuen GI-Fachgruppe, die sich mit diesem Thema „Vorgehensmodelle“ beschäftigen sollte. Den Anstoß dafür gab die Umorganisation des GI-Fachbereiches 5 „Wirtschaftsinformatik“ im März 1993 mit Gründung des Fachausschusses 5.1 „Management der Anwendungsentwicklung und -wartung“. Dieser Fachausschuss sollte drei Fachgruppen umfassen, darunter eine zum Thema „Vorgehensmodelle“.³

Teilnehmer der Gründungssitzung dieser Fachgruppe waren Georg Bremer (ICR), Helmut Burckhardt (SNI), Günther Müller-Luschnat (ALLDATA), Andreas Oberweis (Universität Karlsruhe), Georg v.d. Ropp (Hypo Bank), Siegfried Stein (IBM) und Peter M. Urban (Sapiens).

Wie schon dieser Teilnehmerkreis andeutet, ist die Fachgruppe seit der Gründung bis heute eine der wenigen Fachgruppen in der GI, deren Mitglieder zum weit überwiegenden Teil aus der Wirtschaft kommen, während die Mitglieder aus Hochschule und Forschung deutlich in der Minderheit sind.

Hauptthemen der Gründungssitzung waren eine Sammlung der Themen, mit denen die Fachgruppe sich beschäftigen sollte sowie die Abstimmung der geplanten Aktivitäten. Grundsätzlich lässt sich das Thema Vorgehensmodelle ja aus der Sicht der Wirtschaftsinformatik oder aus der Sicht der Softwaretechnik betrachten. Um den eigenen Aufgabenbereich klar abzugrenzen und Konflikte mit dem Fachbereich „Softwaretechnik“ zu vermeiden, vereinbarten die Teilnehmer, Vorgehensmodelle schwerpunktmäßig aus betriebswirtschaftlicher Sicht zu betrachten. Diese Schwerpunktsetzung führte auch zu dem Namen „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“ der Fachgruppe.

Als wichtigste Themen wurden Methoden, Grundlagen sowie Werkzeugunterstützung definiert. Zur Bearbeitung dieser Themen wurde vereinbart, im nächsten Jahr einen Workshop zu organisieren sowie (nach Möglichkeit gemeinsam mit dem Fachausschuss) eine Zeitschrift herauszugeben.

Eine wichtige Aufgabe bei der Gründung einer solchen Fachgruppe ist die Festlegung der organisatorischen Rahmenbedingungen wie Wahl und Größe des Leitungsgremiums oder Mitgliedsbeitrag. Diese Rahmenbedingungen ebenso wie die Aufgaben, Ziele und Aktivitäten der Fachgruppe wurden bei dem Gründungstreffen vereinbart und anschließend in Form einer Fachgruppenordnung dokumentiert. Unter anderem wurde festgelegt, dass die Fachgruppe von einem aus fünf Personen bestehenden Leitungsgremium geleitet werden soll, das aus seiner Mitte den Sprecher und dessen Stellvertreter wählt.

Zum Sprecher der Fachgruppe in Gründung wurde Hr. Müller-Luschnat gewählt, Stellvertreter wurde Herr Stein.

³ Informationen zur Arbeit des Fachausschusses sind zu finden auf dessen Webseite unter <http://fa-wi-maw.gi.de/>

1994 – 1997: Die Fachgruppe nimmt die Arbeit auf

Auf Grund der in der GI zu klärenden organisatorischen Fragen existierte die Fachgruppe noch eine ganze Weile mit dem Zusatz i. G. (in Gründung). Erst im Oktober 1995 wurde die Fachgruppe offiziell gegründet als FG 5.1.1 „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“.

Bereits im März 1994 fand aber der erste Workshop der Fachgruppe in München statt. Seitdem wird der Workshop der Fachgruppe jährlich durchgeführt, und die Organisation dieses Workshops als Medium zum Erfahrungsaustausch und zur Information über aktuelle Entwicklungen ist die zentrale Aktivität der Fachgruppe. Hauptziele der Workshops sind der Erfahrungsaustausch, die Gelegenheit zur informellen Diskussion in den Pausen sowie die Information über aktuelle Entwicklungen.

Wie bereits bei der Gründungsversammlung vereinbart begann der Fachausschuss mit den beteiligten Fachgruppen 1995, einen „Rundbrief“ als gemeinsame Mitgliederzeitschrift herauszugeben. Dieser Rundbrief erscheint seither jährlich zweimal.⁴

1996 wurde auch eine Webpräsenz der Fachgruppe aufgebaut, wofür der seinerzeitige Arbeitgeber des Sprechers Günther Müller-Luschnat, FAST e.V., die Infrastruktur zur Verfügung stellte.

1997 – 2000: Die Fachgruppe etabliert sich

Im März 1997 fand bei der GMD (Gesellschaft für Mathematik und Datenverarbeitung, mittlerweile in die Fraunhofer-Gesellschaft integriert) in Berlin-Adlershof der 4. Workshop der Fachgruppe statt, diesmal zum Thema „Vorgehensmodelle – Einführung, betrieblicher Einsatz, Werkzeug-Unterstützung und Migration“. Dort wurde u.a. das neue V-Modell 97 vorgestellt, das später unter Mitarbeit der in der Fachgruppe aktiven Manuela Wiemers auch als Buch veröffentlicht wurde, siehe [DW00]. Wesentliche Änderungen im V-Modell 97 gegenüber der Vorversion von 1992 waren die größere Flexibilität in der Anwendung, dank der auch Entwicklungsprojekte mit einer objektorientierten Vorgehensweise unterstützt werden, sollten, sowie die Ausweitung des Anwendungsbereiches von der reinen Softwareentwicklung auf die Systementwicklung (siehe [DW00, S. 3]).

Ein weiteres wesentliches Thema beim Workshop 1997 war die Werkzeugunterstützung für Vorgehensmodelle, und die kleine Ausstellung von entsprechenden Anbietern stieß auf großes Interesse der Teilnehmer. Werkzeugunterstützung blieb daher, neben der sich stärker etablierenden Objektorientierung, auch in den nächsten Jahren einer der Schwerpunkte der Fachgruppenarbeit.

Beim Workshop 1997 gab es auch erstmals einen Tagungsband, der bei der GMD erschien. 1998 gab es leider keinen Tagungsband, aber seit 1999 wurde zu jedem der jähr-

⁴ Einen Überblick über die erschienenen Rundbriefe gibt die Webseite <http://fa-wi-maw.gi.de/rundbrief.html>

lichen Workshops ein Tagungsband herausgegeben, seit 2000 im Shaker-Verlag im einheitlichen Layout.⁵

Beim Workshop 1997 stand die erste Neuwahl der Fachgruppenleitung an. Hier wurde Günther Müller-Luschnat wiedergewählt, neuer Stellvertreter wurde Ralf Kneuper, der auch das Programmkomitee für den Workshop 1997 geleitet hatte.

Ebenfalls in 1997 begann die Arbeit an einem Sammelband zur Arbeit der Fachgruppe, der 1998 im Teubner-Verlag unter dem Titel „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“ erschien, herausgegeben von Ralf Kneuper, Andreas Oberweis und Günther Müller-Luschnat (siehe [KMO98]).⁶

Im Juni 1998 wurde der Workshop der Fachgruppe zusammen mit einer anderen Gruppierung der GI, dem Arbeitskreis GROOM (Grundsätze objektorientierter Modellierung), durchgeführt zum Thema „Anwendung von objektorientierten Entwicklungsstrategien und deren Unterstützung durch Vorgehensmodelle“.

Ebenfalls in 1998 wurde die Tagungsreihe „Modellierung“ ins Leben gerufen, die mittlerweile in einen von derzeit 12 verschiedenen Fachgliederungen der GI getragenen GI-Querschnittsfachausschuss übergegangen ist. Hauptaufgabe dieses Ausschusses ist die regelmäßige Durchführung von Workshops und Tagungen zu übergreifenden Fragen der Modellierung. Von Anfang an war auch die Fachgruppe „Vorgehensmodelle“ daran beteiligt, vertreten durch den damaligen Sprecher Günther Müller-Luschnat, der die Fachgruppe über viele Jahre in diesem Fachausschuss vertrat.

Aufgrund einer Neustrukturierung des Fachbereichs 5 der GI wurde die Nummerierung der Fachgruppe im Sommer 1999 geändert. Bisher: 5.1.1, jetzt: 5.11. Name und Zweck der Fachgruppe blieben unverändert.

In diesem Zeitraum (genauer Zeitpunkt nicht mehr nachvollziehbar) wurde auch der Domainname `vorgehensmodelle.de` auf die Fachgruppe registriert.

2000 – 2003: Agile Vorgehensmodelle kommen in den Fokus

Nachdem jahrelang die „schwergewichtigen“ Vorgehensmodelle und deren Werkzeugunterstützung zentrale Themen bei den Workshops waren, verlagerte sich der Schwerpunkt in diesen Jahren auf die agilen Vorgehensmodelle und Methoden. So waren „Leichte Vorgehensmodelle“ das Thema des Workshops 2001 in Glashütten und es gab heftige Diskussionen über diese neuen Methoden, wie sie in dem wenige Wochen vorher verabschiedeten *Agilen Manifest* beschrieben sind. (Bis dahin war auch die Bezeichnung „leichte“ oder „leichtgewichtige“ Vorgehensmodelle üblich und wurde erst danach durch die heute übliche Bezeichnung „agile“ Vorgehensmodelle abgelöst.) Beim Workshop

⁵ Für 2013 ist geplant, den Tagungsband nicht mehr im Shaker-Verlag herauszugeben, sondern in der LNI-Reihe, die von der GI herausgegeben und für Tagungsbände der Fachgruppen empfohlen wird.

⁶ Nachdem dieser Sammelband einige Jahre lang vergriffen war, soll er in Kürze elektronisch verfügbar werden, siehe <http://www.springer.com/engineering/book/978-3-8154-2605-0>

2002 wurden diese Diskussionen dann unter dem Stichwort „Angepasste Vorgehensmodelle“ weitergeführt.

Im Rahmen des 7. Workshops der Fachgruppe im April 2000 in Bonn wurde eine neue Fachgruppenleitung gewählt mit dem neuen Fachgruppensprecher Ralf Kneuper und seiner Stellvertreterin Manuela Wiemers. Als weitere Mitglieder der Fachgruppenleitung wurden Gerhard Chroust, Thomas Fischer und Christian Steinmann gewählt. Der bisherige Sprecher Günther Müller-Luschnat wurde zum Fachexperten im Leitungsgremium ernannt.

In den Jahren davor war schon regelmäßig eine Bewertung der Vorträge durch die Teilnehmer erfolgt. Um diese Bewertung stärker zu betonen, wurden ab 2001 bis 2006 Preise für die beste Präsentation sowie den besten Inhalt vergeben. Dabei zeigte sich, dass Manuela Wiemers nicht nur als stellvertretende Sprecherin der Fachgruppe agierte, sondern auch ihre Vorträge immer wieder gut ankamen, so dass sie 2001, 2003 und 2005 ausgezeichnet wurde.

Um die Organisation der jährlichen Workshops zu erleichtern, insbesondere auch die Arbeit auf mehrere Schultern zu verteilen, erarbeitete Ralf Kneuper ein „Vorgehensmodell für die Durchführung von Vorgehensmodell-Workshops“, das er beim Workshop 2001 vorstellte. Aktualisierte Versionen dieses Vorgehensmodells dienten jahrelang als Hilfsmittel für die Organisation der Workshops.⁷

Ebenfalls 2001 wurde auch der Web-Auftritt der Fachgruppe komplett überarbeitet und auf ein einheitliches, zu den Webseiten der GI passendes, Design umgestellt.

2003 – 2006: Der Rahmen ändert sich, die Fachgruppenarbeit bleibt

In der Arbeit mit Vorgehensmodellen gab es schon in den Vorjahren immer große Diskussionen über deren praktische Anwendbarkeit und Akzeptanz, und durch das Aufkommen der agilen Vorgehensweisen wurden diese Diskussionen verschärft. Dies spiegelte sich in den Vorträgen bei den Workshops wieder, und die Erfahrungen vieler Vortragender und Teilnehmer zeigten die Schwierigkeiten, Vorgehensmodelle nicht nur als „Schrankware“ zu erstellen, sondern sie auch im Unternehmen als akzeptierte Hilfestellung für alle Beteiligten einzuführen. Parallel dazu gab es immer wieder heftige Diskussionen zwischen den Verfechtern der agilen Methoden, die Vorgehensmodelle wie das V-Modell als enges Korsett betrachten, das die Kreativität der Entwickler einschränkt, und den Verfechtern der „alten Schule“, die agile Methoden als Freibrief für undiszipliniertes Hacken ansehen.

„Hinter den Kulissen“ gab es 2003 auch einige organisatorische Änderungen für die Fachgruppe, denn die GI änderte die Struktur ihrer Fachgliederungen: Statt der bis-

⁷ Unter <http://www.kneuper.de/Publikationen/index.html#TT06> ist die Originalversion dieses Vorgehensmodells verfügbar. Die aktualisierte Fassung auf der Webseite der Fachgruppe ist nicht mehr verfügbar, da das Vorgehen mittlerweile geändert wurde. Oliver Linssen und Marco Kuhmann planen eine neue Version dieses Modells.

herigen dreigliedrigen Struktur aus Fachbereich (hier „Wirtschaftsinformatik“), Fachausschuss (hier „Management der Anwendungsentwicklung und -wartung“) und Fachgruppe gibt es seither nur noch eine zweigliedrige Struktur, die Ebene der Fachausschüsse ist entfallen. Außerdem haben die Fachgruppen statt Nummern jetzt mehr oder weniger sprechende Kürzel, aus der Fachgruppe 5.11 wurde die heutige Fachgruppe WI-VM (Fachgruppe Vorgehensmodelle im Fachbereich Wirtschaftsinformatik).

In diesem Fall war der Fachausschuss aber nicht nur eine organisatorische Zwischenebene, sondern hatte mit der Organisation der Tagungsreihe „Software Management“ und der Herausgabe des Rundbriefes als Kommunikationsmedium der beteiligten Fachgruppen eigene Aufgaben. Daher beschlossen die drei Fachgruppen, den Fachausschuss als gemeinsamen Ausschuss der Fachgruppen weiterzuführen, der diese Aufgaben seither im Auftrag der Fachgruppen übernimmt.

Ebenfalls 2003 gab es turnusgemäß eine Neuwahl der Fachgruppenleitung, bei der die bisherige Leitung weitgehend wiedergewählt wurde. Als Sprecher wurde Ralf Kneuper wiedergewählt, als Stellvertreterin Manuela Wiemers. Auch Gerhard Chroust und Thomas Fischer wurden wiedergewählt; einzig neues Mitglied der Fachgruppenleitung war Roland Petrasch, der Organisator des Workshops 2003.

Der Workshop der Fachgruppe in 2005 fand in Berlin statt, diesmal in den Räumen der Volkswagen Coaching ganz zentral „Unter den Linden“. Mit Reinhard Höhn (Wien) als Leiter des Programmkomitees sowie den Stellvertretern Stephan Höppner und Herbert Wetzel (St. Gallen) gab es drei neue Aktive in der Fachgruppe, die daher als Fachexperten in die Fachgruppenleitung aufgenommen wurden.

Gegen Ende dieser Periode wurden die Webseiten der Fachgruppe vollständig überarbeitet und umgezogen. Während sie bisher auf dem angesprochenen FAST-Server lagen und als reine HTML-Anwendung erstellt waren, gab es jetzt das Angebot der GI, die Webseiten in einem CMS auf Servern der GI bereit zu stellen. Zwar gab es bei diesem neuen Angebot der GI noch einige Kinderkrankheiten zu überstehen, und auch die Einarbeitung in Typo3, das von der GI bereit gestellte CMS, stellte sich als nicht ganz einfach heraus, aber der Umzug konnte dank Unterstützung durch eine Werkstudentin gerade noch vor dem Workshop 2006 abgeschlossen werden

2006 – 2009: Die Fachgruppe im Umbruch

Inhaltlicher Schwerpunkt der Fachgruppenarbeit war in dieser Zeit neben der weiterhin zentralen Frage der Akzeptanz von Vorgehensmodellen vor allem der auch in einer neuen Arbeitsgruppe behandelte Überblick über die verschiedenen Vorgehensmodelle und verwandte Modelle. So ging es beim Workshop 2007 beispielsweise in erster Linie um das V-Modell XT sowie um Reifegradmodelle wie CMMI und SPICE, und 2008 standen erneut das V-Modell XT sowie ITIL und OPM3 im Fokus.

In 2006 fand der 13. Workshop wieder im Süden, nämlich in München statt, organisiert von Ralf Kneuper, der nach neun Jahren als stellvertretender Sprecher bzw. Sprecher der

Fachgruppe dieses Amt abgab, und Hubert Biskup, der schon seit vielen Jahren in der Fachgruppe aktiv ist. Bei der Neuwahl der Fachgruppenleitung wurden Reinhardt Höhn als Sprecher und Stephan Höppner als Stellvertreter gewählt, mit Hubert Biskup, Marco Kuhrmann und Roland Petrasch als weiteren Mitgliedern.

2006 gab es erstmals eine weitere von der Fachgruppe organisierte Veranstaltung, nämlich den Workshop „Vorgehensmodelle in der Praxis“ auf der GI-Jahrestagung, der von 2006 bis 2011 jährlich von Marco Kuhrmann organisiert wurde.

2009 – 2012: Vorgehensmodelle und Projektmanagement

Im Rahmen der Fachgruppensitzung 2009 wurde Oliver Linssen als neuer Sprecher der Fachgruppe gewählt. Reinhard Höhn – bisher Sprecher der Fachgruppe – wurde zum stellvertretenden Fachgruppensprecher gewählt.

Oliver Linssen übernahm es auch gleich, den Webauftritt der Fachgruppe zu aktualisieren, die fehlenden Inhalt der letzten Jahre zu vervollständigen, ihn auf eine neue Version des CMS Typo3 zu migrieren sowie an das neue Design der GI anzupassen. Bei dieser Gelegenheit wurde auch die Fachgruppenordnung zurückgezogen, da eigene Fachgruppenordnungen in der GI nicht mehr üblich sind.

Der inhaltliche Schwerpunkt in diesem Zeitraum lag vor allem auf dem Bezug zwischen Vorgehensmodellen und verwandten Fachgebieten; beispielsweise wurde der Workshop 2010 unter dem Titel „Integration von Vorgehensmodellen und Projektmanagement“ in Zusammenarbeit mit der GPM Deutsche Gesellschaft für Projektmanagement e.V., Region Stuttgart, veranstaltet. 2011 ging es in Kiel um das „Zusammenspiel von Vorgehensmodellen und Organisationsformen“, und 2012 in Düsseldorf um „Qualitätsmanagement und Vorgehensmodelle“.

Als zusätzliche Kommunikationsmöglichkeit der Fachgruppenmitglieder wurde bei Xing eine Diskussionsgruppe eingerichtet. Außerdem wurde ein eigener Email-Verteiler der Fachgruppe eingerichtet, nachdem in der Vergangenheit der Verteiler des Fachbereichs „Wirtschaftsinformatik“ mitbenutzt worden war.

2012 bis heute: Die Fachgruppe heute

Nachdem die Amtszeit der vorigen Fachgruppenleitung ausgelaufen war, wurde bei der Sitzung am Rande des Workshops 2012 ein neues Leitungsgremium gewählt. Erstmals seit Bestehen der Fachgruppe kommt der Fachgruppensprecher mit Prof. Eckhart Hanser von der DHBW in Lörrach nicht aus der Wirtschaft, sondern von einer Hochschule – allerdings einer dualen Hochschule, so dass zu hoffen ist, dass der schon immer angestrebte Austausch von Wirtschaft einerseits und Forschung und Lehre andererseits damit noch verstärkt werden kann.

Ähnlich dem schon in den Jahren zuvor zu beobachteten Trend zu einer wachsenden Anzahl von Mitgliedern im Programmkomitee des Workshops gab es auch bei der Größe des Leitungsgremiums eine Änderung. Während das Gremium bisher entsprechend den bei Gründung der Fachgruppe vereinbarten Festlegungen aus jeweils fünf Personen inkl. Sprecher und stellvertretendem Sprecher bestand, wurde es jetzt auf sieben Personen erweitert, und auch der Kreis der ernannten Fachexperten, der bisher aus maximal drei Personen bestand, wurde jetzt auf sechs Personen ausgedehnt (siehe Tabelle 2 für eine vollständige Aufstellung dieser Personen). Es bleibt abzuwarten, ob das damit verfolgte Ziel, den Kreis der Aktiven in der Fachgruppe zu erweitern, tatsächlich erreicht wird oder im Gegenteil sich niemand für anstehende Aufgaben zuständig fühlt, da es ja noch genügend andere Zuständige gibt, wie das gelegentlich schon bei den Gutachten der Programmkomitee-Mitglieder zu beobachten ist.

2012 wurde der Workshop der Fachgruppe auch erstmals im Herbst statt im Frühjahr durchgeführt, nachdem der ursprünglich im Frühjahr 2012 geplante Termin aus organisatorischen Gründen abgesagt werden musste. Bei dieser neuen Regelung wird es wohl auch in den kommenden Jahren bleiben und der nächste Workshop ist für Oktober 2013 geplant. Anlässlich des zwanzigsten Jahrestages seit Gründung der Fachgruppe sowie des zwanzigsten Workshops ist hier ein Rückblick auf das Erreichte sowie ein Ausblick auf die Herausforderungen der Zukunft geplant.

Herausforderungen für die Zukunft

Entsprechend dem Thema des 20. Workshops soll hier noch ein kurzer Ausblick auf die anstehenden Herausforderungen für die Fachgruppe gegeben werden, sowohl auf inhaltlicher Ebene als auch auf organisatorischer Ebene.

Die inhaltlich größte Herausforderung zum Thema Vorgehensmodelle erscheint, wie schon seit vielen Jahren, die Gestaltung der Gratwanderung zwischen einerseits der genauen Festlegung der Entwicklungsprozesse als Basis für organisatorisches Lernen, Qualitätssicherung und schließlich auch wirtschaftliches Arbeiten, und andererseits der notwendigen kreativen Freiheit der Entwickler als Basis für innovative Lösungen. Derzeit läuft diese Diskussion häufig unter dem Stichwort der Plan-getriebenen gegenüber der agilen Entwicklung (siehe z.B. [BT03]). Je nach Art der zu entwickelnden Lösungen muss zwischen diesen Polen immer wieder ein neuer Mittelweg gefunden werden. Die Beteiligten, insbesondere die Entwickler, müssen überzeugt werden, den gefundenen Mittelweg auch mitzugehen, aber gleichzeitig muss die Organisation offen dafür sein, Verbesserungsideen der Beteiligten aufzugreifen und umzusetzen, am besten, indem die Beteiligten schon bei der Festlegung des Vorgehensmodells eingebunden sind.

Die bei der Definition und Nutzung von Vorgehensmodellen zu berücksichtigenden Beziehungen zu anderen Fachgebieten wurden in den Workshops der letzten Jahre bereits aufgegriffen, bleiben aber sicher auch weiterhin im Fokus. Das gilt insbesondere für die Beziehung zwischen der Softwareentwicklung und dem Betrieb dieser Software, die bislang eher oberflächlich behandelt wurde, in den Unternehmen aber immer wieder zu Spannungen führt. Zwar gibt es unter dem Stichwort „DevOps“ derzeit Ansätze, beide

Themenbereiche stärker zusammen zu führen, aber hier gibt es sicher noch Handlungsbedarf.

Wesentliche organisatorische Herausforderungen für die Fachgruppe sieht der Autor darin, weiterhin ausreichend viele Teilnehmer und Vortragende für den Workshop zu gewinnen. Viele Unternehmen sparen an den Kosten für Weiterbildung, was sich natürlich auf die Teilnehmerzahlen an derartigen Veranstaltungen ebenso wie die Anzahl der Einreichungen aus den Unternehmen auswirkt. Auch sind Vorgehensmodelle in der Forschung derzeit kein Modethema, so dass auch hier die Anzahl der potentiellen Vortragenden eher gering ist.

Eine immer wieder aufgebrachte Frage ist dabei auch die nach dem angestrebten wissenschaftlichen Stellenwert des Workshops. Bisher lag der Schwerpunkt des Workshops immer auf dem Erfahrungsaustausch zwischen Unternehmen, ergänzt durch Vorträge über neue Entwicklungen aus der Forschung. Der Nutzen für die Unternehmen war dadurch relativ groß, teilweise aber auf Kosten des wissenschaftlichen Niveaus. Hier gab es daher wiederholt den Wunsch, das wissenschaftliche Niveau zu steigern, eventuell sogar den Workshop zu einer wissenschaftlichen Tagung weiterzuentwickeln. Neben der Schwierigkeit, entsprechende Vortragende zu gewinnen, wird die zentrale Herausforderung bei diesem Ansatz darin bestehen, den Unternehmen auch weiterhin den entsprechenden Anwendungsnutzen zu bieten, damit man sie nicht als Teilnehmer verliert.

Anhang

Arbeitskreise

Im Laufe der Jahre gab es eine Reihe von Arbeitskreisen der Fachgruppe, die meist über begrenzte Zeit ein Thema bearbeiteten und dann auch wieder eingestellt wurden.

Eine Ausnahme ist der Arbeitskreis „Begriffe“, der kurz nach Gründung der Fachgruppe seine Arbeit aufnahm und immer noch aktiv ist. Dieser 1994 gegründete Arbeitskreis arbeitet gemeinsam mit ähnlichen Arbeitskreisen der GI-Fachgruppen „Software Engineering“ (SWT-SE) und „Test, Analyse und Verifikation von Software“ (SWT-TAV) an einem Informatik-Begriffsnetz, insbesondere an einer Begriffssammlung zum Thema „Vorgehensmodelle“.

Weitere Arbeitskreise, die mittlerweile wieder eingestellt wurden, waren:

- Berufsbild Softwaretester (gemeinsam mit SWT-TAV) (gegründet 2001)
- Anwendbarkeit von Vorgehensmodellen (gegründet 2000)
- SPIN Rhein-Main (gegründet 2003)

Derzeit hat die Fachgruppe die folgenden aktiven Arbeitskreise:

- Begriffe
- Vorgehensmodelle – Übersicht und Vergleich
- Vorgehensmodell-Katalog
- Operational Excellence

Workshops der Fachgruppe

Nr.	Jahr	Ort	Leitung PK, OK	Thema
	1993			Gründungstreffen
1	1994	München	Müller-Luschnat, Oberweis, v.d. Ropp	
2	1995	Karlsruhe	Müller-Luschnat, Oberweis, Rohloff	
3	1996	München	Müller-Luschnat, Hummel	
4	1997	Berlin	Kneuper, Mon- tenegro	
5	1998	Frankfurt	(nicht mehr nachvollziehbar)	
6	1999	Kaiserslautern	Kneuper, Verlage	Vorgehensmodelle, Prozess- verbesserung und Quali- tätsmanagement
7	2000	Bonn	Andelfinger, Herzwurm, Mellis, Müller-Luschnat	Vorgehensmodelle, Wirt- schaftlichkeit, Werkzeugun- terstützung und Wissensma- nagement
8	2001	Glashütten/Ts.	Kneuper, Wiemers	Leichte Vorgehensmodelle
9	2002	Glashütten/Ts.	Kneuper, Petrasch, Wiemers	Angepasste Vorgehensmo- delle
10	2003	Berlin	Petrasch, Wie- mers, Kneuper	Praxistauglichkeit von Vor- gehensmodellen
11	2004	Berlin	Petrasch, Wie- mers, Kneuper	Akzeptanz von Vorgehens- modellen
12	2005	Berlin	Höhn, Höppner, Wetzel, Petrasch, Wiemers	Entscheidungsfall Vorge- hensmodelle

Nr.	Jahr	Ort	Leitung PK, OK	Thema
13	2006	München	Biskup, Kneuper	Nutzen und Nutzung von Vorgehensmodellen
14	2007	München	Höhn, Höppner, Wetzel	Vorgehensmodelle und Projektmanagement – Assessment, Zertifizierung, Akkreditierung
15	2008	Berlin	Höhn, Höppner, Wetzel, Petrasch	Vorgehensmodelle und der Product Life Cycle – Projekt und Betrieb von IT-Lösungen
16	2009	Düsseldorf	Höhn, Höppner, Wetzel, Linssen	Vorgehensmodelle und Implementierungsfragen Akquisition – Lokalisierung – soziale Maßnahmen – Werkzeuge
17	2010	Stuttgart	Greb, Lange, Linssen	Integration von Vorgehensmodellen und Projektmanagement
18	2011	Kiel	Kuhrmann, Linssen, Weißels	Zusammenspiel von Vorgehensmodellen und Organisationsformen
19	2012	Düsseldorf	Kuhrmann, Linssen	Qualitätsmanagement und Vorgehensmodelle
20	2013	Lörrach	Mikusz, Fazal-Baqaie, Hanser	Vorgehensmodelle 2013 – Anspruch und Wirklichkeit

Tabelle 1 Übersicht über die Workshops der Fachgruppe

Fachgruppenleitungen

Zeitraum	Sprecher	Stellvertretender Sprecher	Fachgruppenleitung	Fachexperten
1993 – 1997	Günther Müller-Luschnat	Siegfried Stein		
1997 – 2000	Günther Müller-Luschnat	Ralf Kneuper	Gerhard Chroust, Thomas Fischer, Rainer Midderhoff	
2000 – 2003	Ralf Kneuper	Manuela Wiemers	Gerhard Chroust, Thomas Fischer, Christian Steinmann	Günther Müller-Luschnat
2003 – 2006	Ralf Kneuper	Manuela Wiemers	Gerhard Chroust, Thomas Fischer, Roland Petrasch	Reinhard Höhn, Stephan Höppner, Roland Petrasch
2006 – 2009	Reinhard Höhn	Stephan Höppner	Hubert Biskup, Marco Kuhrmann, Roland Petrasch	Gerhard Chroust, Ralf Kneuper, Herbert Wetzel
2009 – 2012	Oliver Linssen	Reinhard Höhn	Thomas Greb, Ralf Kneuper, Marco Kuhrmann	Gerhard Chroust, Eckhard Hanser, Stephan Höppner
2012 – 2015	Eckhart Hanser	Oliver Linssen	Doris Weßels, Volker Arendt, Hubert Biskup, Masud Fazal-Baqaie, Martin Mikusz	Gerhard Chroust, Marco Kuhrmann, Ralf Kneuper, Reinhard Höhn, Thomas Greb, Stephan Höppner

Tabelle 2 Übersicht der Fachgruppenleitungen

Literaturverzeichnis

- [BT03] Boehm, B.; Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley, 2003.
- [Ch92] Chroust, G.: Modelle der Software-Entwicklung. Oldenbourg Verlag, 1992.
- [DW00] Dröschel, W.; Wiemers, M.: Das V-Modell 97. Oldenbourg-Verlag, 2000.
- [HH08] Höhn, R.; Höppner, S.: Das V-Modell XT. Springer-Verlag, 2008.
- [KMO98] Kneuper, R.; Müller-Luschnat, G.; Oberweis, A.: Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Teubner Verlagsgesellschaft, 1998.
- [Kn12] Kneuper, R.: Geschichte der GI-Fachgruppe “Vorgehensmodelle für die betriebliche Anwendungsentwicklung”. 33. Rundbrief des Fachausschusses WI-MAW, März 2012.
- [Sc94] Schäfer, S.: Objektorientierte Entwurfsmethoden. Verfahren zum objektorientierten Softwareentwurf im Überblick. Addison-Wesley, 1994.
- [We92] Weber, H.: Die Software-Krise und ihre Macher. Springer-Verlag, 1992.

Teil II

Hauptprogramm

Organisationsspezifische Anpassungen von Vorgehensmodellen – Nutzen und Herausforderungen

Joachim Schramm¹, Thomas Ternité¹, Marco Kuhrmann²

¹Technische Universität Clausthal, Institut für Informatik
Julius-Albert-Straße 4
38678 Clausthal-Zellerfeld
joachim.schramm@tu-clausthal.de
thomas.ternite@tu-clausthal.de

²Technische Universität München, Fakultät für Informatik
Boltzmannstr. 3
85748 Garching
kuhrmann@in.tum.de

Abstract: Vorgehensmodelle stellen eine wichtige Quelle für bewährte Vorgehensweisen und die Einführung von Methoden dar. Allerdings wird häufig übersehen, dass Vorteile eines Vorgehensmodells stark von der tatsächlichen Nutzbarkeit abhängen. Vorgehensmodelle sind mitunter so abstrakt, dass ihre Vorteile nur dann vollumfänglich zur Geltung kommen, wenn die angebotenen Inhalte auf die jeweiligen Rahmenbedingungen angepasst und konkretisiert werden. Eine geplante und strukturierte organisationsspezifische Anpassung kann die Nutzbarkeit signifikant erhöhen und die Projektteams von der Notwendigkeit entlasten, Anpassungen des Vorgehens ad hoc vornehmen zu müssen. In diesem Artikel wird der Nutzen organisationsspezifischer Anpassungen von Vorgehensmodellen diskutiert. Dazu werden eine Charakterisierung organisationsspezifischer Anpassungen vorgenommen und Herausforderungen aufgezeigt, die mit solchen Anpassungen einhergehen.

1 Einleitung

Vorgehensmodelle werden in der Literatur vielfach als Vorlage für Projekte bezeichnet oder als Muster ingenieurmäßigen Vorgehens in Softwareprojekten. Tatsächlich und auf das Wesentliche reduziert, sind Vorgehensmodelle eine Anleitung, ein Spielplan, der die Zusammenarbeit von Menschen beschreibt, die kooperativ eine Aufgabe bearbeiten und eine Lösung erstellen sollen. Unter diesem Gesichtspunkt kommt der organisationsspezifischen Anpassung eines Vorgehensmodells eine besondere Rolle zu. Bei der Einführung eines Vorgehensmodells kommt es insbesondere darauf an, organisatorische Besonderheiten zu erfassen, herauszuarbeiten, welche Terminologie verwendet wird, welche Ergebnisse in Projekten üblicherweise erarbeitet werden und mit welchen Methoden. Die Erfahrung zeigt hierbei, dass diese Aufgaben zwingend erforderlich sind, da die einfache Anwendung eines generischen Standardvorgehensmodells viele Spezialitäten einzelner

Organisationen nicht erfasst. Werden diese Spezialitäten nicht erfasst, können Widersprüche zwischen dem Vorgehensmodell auf der einen Seite und gelebten Prozessen auf der anderen Seite entstehen, z.B. im Bereich Projektmanagement, Aufbauorganisation, Unternehmensstrategie und sonstigen Rahmenbedingungen – eine Diskrepanz zwischen Anspruch und Wirklichkeit. Aus Sicht eines Unternehmens kann es beispielsweise eines der strategischen Unternehmensziele sein, eine hoch-qualitative Softwareentwicklung zu betreiben und gleichzeitig nach diversen Normen zertifiziert zu sein. Ein organisations-spezifisches Vorgehensmodell kann im Rahmen einer solchen Strategie eine wesentliche Rolle spielen. Oftmals wird dabei auf ein Standardmodell als Vorlage zurückgegriffen, um nicht mehr bei „Null“ anfangen zu müssen und um den Bezug zu etablierten Standards zu betonen. Trotzdem sind auch bei Verwendung eines Standardvorgehensmodells als Vorlage Anpassungsaufwand, Trainings- und Einführungsmaßnahmen, sowie ggf. Maßnahmen und Vorgaben für die langfristige Pflege einzuplanen. Dieser Aufwand und viele Herausforderungen und sogar Fallen werden oft unterschätzt. In Konsequenz ist oft ein „Verteufeln“ von Anpassungsprojekten oder Vorgehensmodellen in Gänge zu beobachten. Vorteile, die Anpassungsprojekte und die resultierenden organisationsspezifischen Vorgehensmodelle haben, werden dann oft nicht mehr zur Kenntnis genommen.

In diesem Beitrag wird in Kapitel 2 zunächst Bezug zum aktuellen Stand von Wissenschaft und Praxis hergestellt. Dabei wird aufgezeigt, welche Möglichkeiten der Anpassung es gibt und mit welchen Vor- und Nachteilen sie einhergehen und in wie fern praktisch von der Möglichkeit der organisationsspezifischen Anpassung Gebrauch gemacht wird. In Kapitel 3 wird der Mehrwert von organisationsspezifischen Anpassungen dargestellt, bevor in Kapitel 4 auf die Herausforderungen von Anpassungen eingegangen wird. Kapitel 5 bildet den Abschluss mit einem kurzen Fazit.

2 Hintergrund, Stand der Technik und verwandte Arbeiten

Die Durchführung einer organisationsspezifischen Anpassung eines Vorgehensmodells resultiert im Aufbau von Varianten und Familien von Vorgehensmodellen (auch als Prozesslinien bezeichnet), was zunehmend eine koordinierte und strukturierte Vorgehensweise erfordert. Der strukturierte und geplante Aufbau von Prozesslinien wird von Rombach [Ro05] als Lösung vorgeschlagen, um komplexe integrierte Prozesse effizient zu entwickeln und zu warten. Eine Prozesslinie wird hierbei in Analogie zu einer Software Produktlinie [SEI05] verstanden, in der ein „stabiler“ Kernprozess den Ausgangspunkt der Entwicklung abgeleiteter organisations- bzw. projektspezifischer Prozesse bildet. In der Terminologie der Produktlinien werden die Bestandteile des Kernprozesses als „Commonalities“ bezeichnet, also als allen abgeleiteten Prozessen gemeinsame Anteile. Änderungen, Erweiterungen usw. werden über sogenannte „Variabilities“ realisiert, welche entweder echte Erweiterungen zum Kernprozess darstellen oder Änderungsoperationen enthalten, die Strukturen und Inhalte des Kernprozesses anpassen [NZ12].

Neben den Strukturen, welche erforderlich sind, um eine Prozesslinie aufzubauen, ist ein umfassendes methodisches Vorgehen erforderlich, das die Erstellung/den Aufbau sowie die Verwaltung und Weiterentwicklung einer Prozesslinie unterstützt. Bereits seit Mitte der 1990’er Jahre wird im Bereich des (Situational) Method Engineering (vgl. [Br96],

[He06] oder [Br99]) an solchen Verfahren gearbeitet, wobei in [KMT13] gezeigt wurde, dass diese Forschungsdisziplin noch keine ausreichende Reife erreicht hat, um eine umfassende methodische Unterstützung zur Verfügung zu stellen.

Der Bedarf eines konkreten, methodischen Vorgehens wurde bei der Wartung und Weiterentwicklung des V-Modell XT notwendig, auf das wir uns im Folgenden auch als Beispiel für eine sich in der praktischen Anwendung befindliche Prozesslinie beziehen. Im Kontext des V-Modell XT war es 2007 erforderlich, die (technische) Prozessarchitektur auf ein neues, langfristig tragfähiges Fundament zu stellen. In Abbildung 1 ist zur Illustration der „Familienstammbaum“ des V-Modell XT nach [SKM12] gezeigt. Die Zunahme an V-Modell-Varianten über die Zeit zog aufwendige Arbeiten hinsichtlich der Analyse der Änderungen nach sich (vgl. [OC07], [Joc09a] und [Joc09b]). Für die weitere Entwicklung des V-Modell XT war es daher erforderlich, das Metamodell sowie die Methoden für Anpassung und Variabilität neu zu entwickeln, wobei neben den konzeptuellen Vorschlägen aus dem Bereich der Prozesslinien auch neue Konzepte, etwa eine umfassende Anpassungsmethode [KTF11] entwickelt wurden.

In [SKM12] wurde eine umfassende Untersuchung hinsichtlich der Reife aktueller Prozessframeworks bezüglich der Fähigkeiten zur Unterstützung methodischen Vorgehens bei der Prozessentwicklung und beim Management komplexer Vorgehensmodelle durchgeführt. In [KMS13] wurde darauf aufbauend geschlossen, dass lediglich SPEM und das V-Modell XT als gleichwertige Prozessframeworks für solche Aufgaben in Frage kommen, wobei lediglich das V-Modell XT einer nachvollziehbaren Evolution unterliegt. In Bezug auf die methodische Unterstützung der Anpassung von Vorgehensmodellen auf Organisationen zeichnet sich ein ähnliches Bild. In [KMS13] wurden die Prozessframeworks V-Modell XT, SPEM, OPEN und ISO 24744 auch hinsichtlich ihrer methodischen Unterstützung für Prozessingenieure untersucht. Lediglich für das V-Modell XT und SPEM sind Anleitungen verfügbar, die Prozessingenieure bei der Anpassung eines Vorgehensmodells unterstützen. Eine konkrete und umfassende Anpassungsmethode definiert hierbei lediglich das V-Modell XT [KTF11], während für SPEM lediglich eine Anleitung zur Nutzung des Eclipse-basierten Editors verfügbar ist.

2.1 Organisationsspezifische Anpassungen von Vorgehensmodellen

In Abbildung 1 sind alle V-Modell-XT-Varianten (Standardmodell, organisationsspezifische Anpassungen und deren Versionen) zu sehen, welche zum Zeitpunkt der Studie in [KMS13] bekannt waren. Anpassungen des V-Modell XT durch öffentliche Träger gibt es für die Behörden im Allgemeinen (V-Modell XT Bund; gesteuert vom Bundesverwaltungsamt), die Bundesnetzagentur (BNetzA), das Zentrum für Informationsverarbeitung und Informationstechnik (ZIVIT), die bayerische Finanzverwaltung und die Bundeswehr. Im Rahmen des WEIT e.V. (Verein zur Weiterentwicklung des V-Modell XT) sind Anpassungen in der Industrie durch die Firmen Witt Weiden und EADS (FlyXT) bekannt.

Auch für SPEM-basierte Vorgehensmodelle listen Ruiz-Rube et al. [Ru12] bekannte und dokumentierte Anpassungen auf, wobei es sich in der Regel und nur mit wenigen Aus-

nahmen (etwa der Rational Unified Process selbst oder Hermes) um sehr spezifische und individuelle Prozesse überschaubaren Umfangs oder rein akademische Arbeiten handelt.

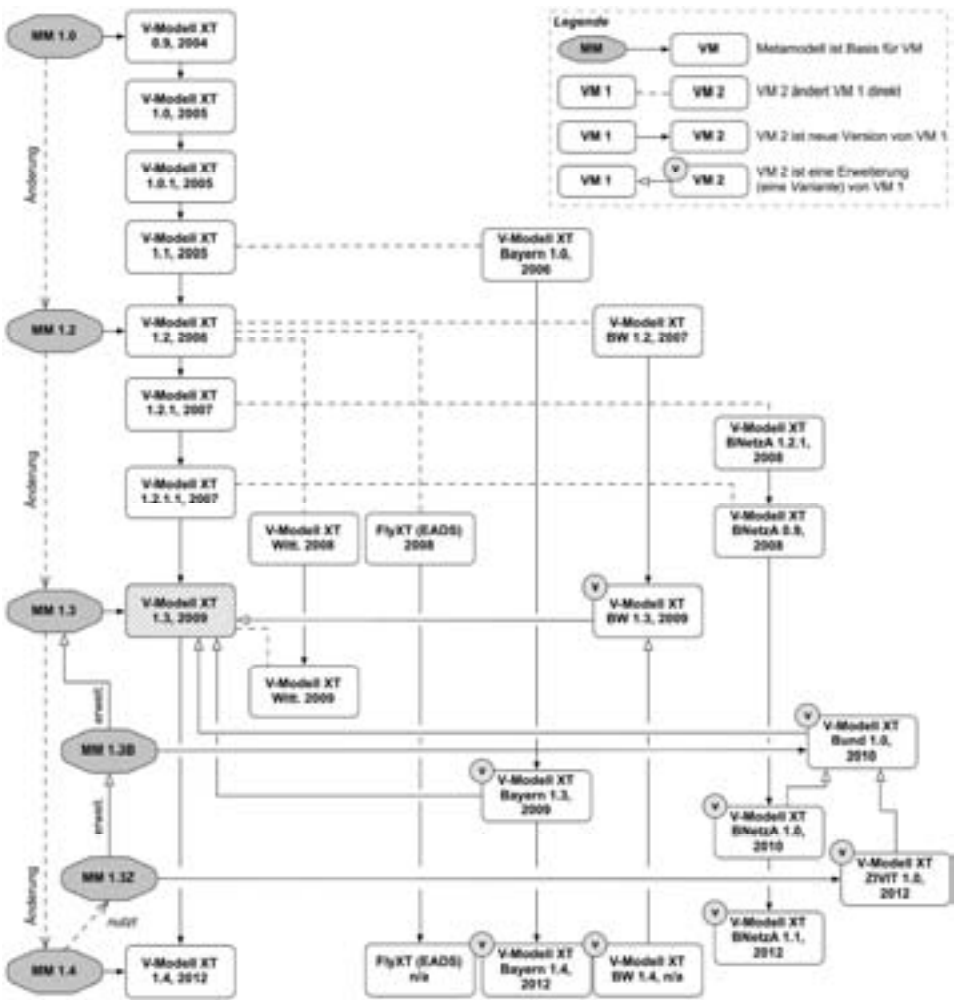


Abbildung 1: Familienstammbaum des V-Modell XT

Eine mit Abbildung 1 vergleichbare Darstellung der SPEM-basierten Prozesse ist [SKM12] zu entnehmen. Der grundsätzliche Bedarf an organisationspezifischen Anpassungen von Vorgehensmodellen erscheint somit offensichtlich. Über den allgemeinen Grad des Bedarfs an Anpassungen kann jedoch im Rahmen dieses Artikels keine Aussage getroffen werden. Es ist weder bekannt, wie viele Unternehmen Bedarf an spezifischen Anpassungen haben, noch in welchem Ausmaß dieser Bedarf bereits durch z.B. nicht öffentlich bekannt gemachte unternehmensinterne Vorgehensmodelle gedeckt wurde.

2.2 Möglichkeiten der Anpassung

Die Möglichkeiten zur Anpassung eines Vorgehensmodells werden im Wesentlichen durch zwei Aspekte getrieben und gesteuert: (1) die methodischen Anforderungen und Möglichkeiten zu ihrer Umsetzung und (2) die technischen Möglichkeiten, die Prozessframeworks zur Verfügung stellen, um Anpassungen auch tatsächlich durchführen zu können.

Der methodische Aspekt wird maßgeblich davon bestimmt, ob es entweder eine dokumentierte und nachvollziehbare Anpassungsmethode gibt, oder ob ein Prozessingenieur über ausreichend übertragbares Wissen verfügt, um auch mit einem nicht dokumentierten Prozessframework zu arbeiten. Hinzu kommt, dass im Rahmen der Prozessanpassung nicht nur das „Umsetzen“ erforderlich ist. Auch Analyse-, Abstimmungs- und Prüftätigkeiten gehören in diesen Bereich, weshalb Anpassungsprojekte in ihrem Umfang und in ihrer Komplexität mit Softwareentwicklungsprojekten vergleichbar sind. Auf der technischen Ebene ist das Vorhandensein einer adäquaten Infrastruktur essenziell.

Die methodischen und technischen Erfordernisse müssen auf die grundsätzliche Vorgehensweise abgestimmt werden. Die Erstellung eines Vorgehensmodells kann somit im Wesentlichen in folgende Klassen einordnet werden:

- *Neuaufbau eines Vorgehensmodells*: Dies ist die vollständige Neuentwicklung eines Vorgehensmodells und aller erforderlichen Komponenten. Dies kann unter Zuhilfenahme eines vorhandenen Metamodells wie SPEM oder dem V-Modell XT Metamodell, aber auch losgelöst davon unter Entwicklung eines spezifischen Metamodells geschehen.
- *Änderung eines Vorgehensmodells*: Die Inhalte eines vorhandenen Vorgehensmodells werden auf die aktuellen Erfordernisse hin angepasst. Inhaltlich werden Umbenennungen und Textergänzungen an vorhandenen Elementen (Artefakte, Rollen, etc.) vorgenommen, aber auch neue Elemente ergänzt oder nicht mehr benötigte Elemente entfernt. Dabei wird das zugrunde liegende Metamodell i.d.R. übernommen oder ein Teilbereichen angepasst.
- *Variantenbildung in einer Prozesslinie*: Basierend auf einem gemeinsamen Kernprozess wird in einem Erweiterungsmodell eine neue Variante erstellt. Die vorgenommenen Änderungen entsprechen im Wesentlichen denen aus dem Punkt „Änderung eines Vorgehensmodells“, allerdings geschieht die Änderung nicht direkt, sondern unter Zuhilfenahme von speziellen Techniken wie maschinell interpretierten Änderungsoperationen oder Tailoring-Mechanismen (siehe [KTF11], [Ku08], [Te10]).

Jede der oben stehenden Vorgehensweisen weist Besonderheiten auf, auf die im Rahmen der Prozessanpassung eingegangen werden muss. So ist beispielsweise die Änderung eines gegebenen Vorgehensmodells auf den ersten Blick das einfachste und schnellste Vorgehen; jedoch weist dieses gerade im Hinblick auf die langfristige Pflege und Evolution erhebliche Mängel auf (vgl. [OC07], [Oc09a] oder [Oc09b]). Die Variantenbildung ist mit Blick auf die langfristige Entwicklung das geschicktere Vorgehen; jedoch erfor-

dert dieses Vorgehen profunde methodische sowie technische Kenntnisse und ein informiertes Abwägen zwischen zwei entgegengesetzten Prinzipien: (1) Einhaltung der durch die Prozesslinie vorgesehenen Anpassungsmechanismen¹ und (2) Verzicht auf die vorgesehenen Mechanismen zugunsten einer umfassenden Anpassungsfreiheit.

3 Mehrwert organisationsspezifischer Vorgehensmodelle

Organisationsspezifische Vorgehensmodelle besitzen gegenüber einem Vorgehensmodellstandard eine Reihe von Vorzügen, die es attraktiv machen eine organisationsspezifische Anpassung in Angriff zu nehmen. Im Einzelnen sind dies:

1. Berücksichtigung spezifischer Prozesse und Prozessartefakte
2. Organisationsspezifische Terminologie (als Spezialfall von 1.)
3. Konkretisierung generischer Vorgaben
4. Chance auf Integration bereits vorhandener isolierter Prozessfragmente
5. Kompatibilität mit vorhandenen oder zu schaffenden Organisationsstrukturen

3.1 Berücksichtigung spezifischer Prozesse und Prozessartefakte

Um einer Ablehnung entgegenzuwirken und einen organisationsspezifischen Begriffssaparat einzuführen, werden bei organisationsspezifischen Vorgehensmodelladaptierungen spezifische Prozesse und Prozessartefakte ergänzt, angepasst und nicht (mehr) benötigte Prozessartefakte weggelassen.

Das V-Modell XT bietet beispielsweise die Möglichkeit, Prozessartefakte wie Vorgehensbausteine, Produkte, Rollen, Aktivitäten oder Entscheidungspunkte zu modifizieren hinzuzufügen oder zu entfernen. Bestehende Prozessdokumente wie die „Betriebliche Freigabeerklärung“ wurden beispielsweise ergänzend in das V-Modell XT Bund integriert.

3.2 Organisationsspezifische Terminologie

Die Terminologie ist eine kritische Komponente eines Vorgehensmodells, beschreibt sie doch die alltäglichen Arbeitsgegenstände. Streng genommen handelt es sich dabei um einen Spezialfall von Absatz 3.1, der hier allerdings aufgrund seines durch Anwender wahrgenommenen hohen Stellenwertes noch einmal gesondert betrachtet wird.

¹ Dadurch entsteht u.U. der Vorteil, dass ein etwaiger Konformitätsnachweis leichter zu erbringen und die zukünftige Weiterentwicklung des Vorgehensmodells (sowohl des Standards, als auch der organisationsspezifischen Anpassung) weniger Aufwand mit sich bringt. Der Nachteil ist eine mögliche Einschränkung der Handlungsfreiheit bei der Anpassung.

Eine neue, fremde oder abweichende Terminologie eines Standardvorgehensmodells kann zu Irritationen und im schlimmsten Fall zur (strikten) Ablehnung des Vorgehensmodells führen. Besonders kritisch hinsichtlich der Terminologie sind üblicherweise die Namen von Rollen, da diese mit der Position und dem Status der jeweiligen Rolleninhaber assoziiert sind. Auch die Bezeichnung von Ergebnissen ist wichtig, insbesondere wenn Ergebnisse zwischen Projekten oder zwischen Auftraggeber und Auftragnehmer ausgetauscht werden. Eine organisationsspezifische Terminologie hilft, den aktuellen Begriffsapparat zu strukturieren und an eine etablierte Terminologie anzupassen. Die Nicht-Berücksichtigung führt in der Regel zur Ablehnung des Vorgehensmodells.

Beispielsweise wurde im V-Modell XT Bund 1.1 dieser Diskrepanz zwischen den Terminologien Projektleiter und Projektmanager entgegengewirkt. Der Projektmanager wurde in Projekteigner umbenannt. Zweck der Umbenennung war die Klarstellung der Benennung der Rolle, welche in der praktischen Anwendung immer wieder zu Konflikten geführt hat, da die dieser Rolle zugeordneten Aufgaben nicht denen des Projekteigners entsprachen.

3.3 Konkretisierung generischer Vorgaben

Standardvorgehensmodelle sind in der Regel in weiten Teilen generisch verfasst [Ro05], sodass eine Anwendung ohne Übertragung in den jeweiligen Projekt- und Organisationskontext schwierig und in Einzelfällen auch unproduktiv ist. Beispielsweise schreibt das V-Modell XT die inhaltlichen Anforderungen an eine Umfangs- und Aufwandschätzung vor, belässt es jedoch dem Anwender, welche Schätzverfahren, Methoden und Werkzeuge zum Einsatz kommen [Ku07]. Diese Abstraktion ist erforderlich, damit ein generischer Standard in der Breite einsatzfähig bleibt. Dennoch ist klar, dass spätestens bei der Durchführung der vorgeschriebenen Aktivitäten eine Entscheidung getroffen werden muss, wie in der vorliegenden Situation genau vorgegangen wird. Im Idealfall wendet der Anwender dann ein geeignetes Verfahren an und erfüllt somit umfänglich die Anforderungen des Vorgehensmodells. Jedoch besteht das Risiko, dass aufgrund von Ressourcenengpässen oder mangelnder Erfahrung ein höchstens annähernd geeignetes Verfahren zum Einsatz kommt. Selbst im besten Fall muss entweder im laufenden Projekt ein Verfahren entwickelt oder als Baustein aus früheren Projekten übernommen werden.

Die Durchführung einer organisationsspezifischen Anpassung bietet in diesem Zusammenhang die Gelegenheit, die notwendige Konkretisierung vorwegzunehmen. Dadurch werden einerseits die Projektbeteiligten von dieser Pflicht entbunden. Dies macht die Arbeit für den Einzelnen einfacher und vermeidet Mehraufwand. Zum anderen lässt sich ein organisationsspezifisches Verfahren in die vorhandenen Strukturen der Organisation besser einbetten als die im Projektgeschäft entstandenen ad hoc Lösungen. Die Integration mit bereits vorhandenen Verfahren und Werkzeugen ist dann zwar ggf. aufwendiger, allerdings lassen sich durch die potenzielle Wiederverwendung von Wissen und Verfahren auch zusätzliche positive Nebeneffekte erzielen.

3.4 Chance auf Integration bereits vorhandener isolierter Prozessfragmente

Im Allgemeinen gibt es in Organisationen Prozesse und dazu passende Vorlagen und Werkzeuge. Ein organisationsspezifisches Vorgehensmodell muss so beschaffen sein, dass solche „Assets“ aufgegriffen, in den bereitgestellten Verfahrensschatz integriert und für den Anwender geeignet präsentiert werden. Beispielsweise bietet das V-Modell XT Bund seinen Anwendern als Hilfestellung für die Vertragsgestaltung die Vorlage „EVB-IT Systemvertrag“ direkt als Bestandteil der Prozessdokumentation an.

Auf diese Weise werden fragmentarisch und ggf. nicht in der Breite eingeführte Lösungen und Prozesse konsolidiert und in einen Gesamtzusammenhang gebracht. Ein Vorgehensmodell kann so als „Einstiegsportal“ dienen und für die betrachteten Prozessbereiche in übersichtlicher und steuerbarer Weise Werkzeuge, Vorlagen und Erklärungen anbieten.

3.5 Kompatibilität mit vorhandenen oder zu schaffenden Organisationsstrukturen

Vorgehensmodelle stehen nicht für sich allein. Sie betten sich in ein organisatorisches Umfeld ein, in dem Projekte durchgeführt werden. Dem entsprechend sind Projekte zwar temporäre, jedoch nicht isolierte Organisationsstrukturen. Projekte müssen sich in eine Organisation einfügen. Konsequenterweise muss damit auch das Vorgehensmodell, nach dem ein Projekt durchgeführt wird, zur Organisation passen. Insbesondere gehört dazu, dass Vorgehensmodelle nicht nur das Projekt betrachten, sondern auch mindestens Schnittstellen zu den umgebenden Organisationsprozessen, etwa Buchhaltung, Personalwesen oder Einkauf, herstellen. Im Rahmen einer organisationsspezifischen Anpassung eines Vorgehensmodells sollten daher auch solche Schnittstellen berücksichtigt werden. Ansonsten besteht die Gefahr, dass ein Vorgehensmodell für die Softwareentwicklung als Fremdkörper eine organisatorische Immunreaktion hervorruft.

Im V-Modell XT Bund hat man einen wichtigen Schritt getan, um die Einbettung in die Organisation deutlicher darzustellen, als das V-Modell XT dies vermochte. Dies hat sich auf die Weiterentwicklung des Standards ausgewirkt. Seit Version 1.4 unterscheidet das V-Modell XT daher zwischen „Projektrollen“ und „Organisationsrollen“, sodass direkt ersichtlich wird, ob beispielsweise datenschutzbezogene Tätigkeiten von einer projektspezifischen Rolle (Datenschutzverantwortlicher), oder von einer organisationsweit definierten Rolle (Datenschutzbeauftragter) durchgeführt werden.

4 Herausforderungen

Allgemein betrachtet lassen sich die folgenden Herausforderungen bei der Durchführung (und anschließenden Einführung) von organisationsspezifischen Anpassungen aufzählen:

1. Weiterentwicklung des Standardvorgehensmodells
2. Fortentwicklung des zugrunde liegenden Metamodells
3. Klärung der Konformität zum Standardvorgehensmodell
4. Abhängigkeiten von der Werkzeuglandschaft
5. Rückfluss zum Vorgehensmodellstandard

4.1 Weiterentwicklung des Standardvorgehensmodells

Eine erste Herausforderung ist, dass ein Standardvorgehensmodell in der Regel unabhängig von der organisationsspezifischen Anpassung weiterentwickelt wird. Einerseits ist es keine triviale Aufgabe, alle dabei gemachten Änderungen aufzuspüren [Ar08], da ein simpler Textvergleich selten ausreicht: binär kodierte Begleitdokumentationen (Bilder, Werkzeuge, etc.) erfordern in der Regel sehr spezialisierte Vergleichssoftware, sofern ein Vergleich überhaupt möglich ist. Andererseits ist zu berücksichtigen, dass nicht nur das Standardvorgehensmodell gegenüber seiner Vorversion geändert wurde, sondern auch das organisationsspezifische Modell gegenüber dem ursprünglichen Standardmodell. Wenn die Neuerungen einer überarbeiteten Fassung des Standardvorgehensmodells auch in der organisationsspezifischen Anpassung einfließen sollen, muss sichergestellt sein, dass man in der Lage ist, sowohl die vorgenommenen Änderungen im organisationsspezifischen Vorgehensmodell aufzählen zu können (z.B. durch Release Notes) als auch die des Standardmodells. Zudem müssen die Gründe für etwaig umgesetzte Änderungen dokumentiert vorliegen. Das betrifft nicht nur den Konfliktfall bei „parallel“ veränderten Prozesselementen. Auch technisch völlig unabhängige Änderungen (z.B. in verschiedenen Kapiteln der Prozessdokumentation) können inhaltlich so voneinander abhängen, dass eine Konsistenzprüfung erforderlich ist.

Eine Analogie kann hierbei zum Customizing oder zur Erweiterungsprogrammierung von Standardsoftware gezogen werden. In der Regel wird eine neu eingeführte Standardsoftware durch Anpassung an Prozesse und Strukturen in der Organisation angepasst [Lw12], etwa durch Konfiguration oder Parametrisierung. Wird nach einiger Zeit ein neues Release der Standardsoftware veröffentlicht und die Organisation entscheidet sich, dieses Release zu verwenden, müssen die Neuerungen geprüft werden und je nach Komplexität der Anpassung – besonders bei Veränderungen – erneut durchgeführt werden [HPW86].

4.2 Fortentwicklung des zugrunde liegenden Metamodells

Das V-Modell XT basiert auf einem formalen Metamodell, wodurch u.a. eine organisationspezifische Anpassung möglich ist. Dadurch entstehen aber auch Herausforderungen. Im V-Modell XT Metamodell wird beispielsweise festgelegt, dass es das Element „Rolle“ und das Element „Produkt“ gibt, und das genau eine Rolle für ein Produkt „verantwortlich“ sein muss.

Sollen zusätzliche Informationen strukturiert aufgenommen werden (z.B. eine „Rolle“ soll einer „Kategorie“ angehören), muss das Metamodell angepasst werden. Werden solche Metamodelländerungen gemacht, müssen konkrete Anleitungen erstellt werden, wie organisationspezifische Anpassungen des V-Modell XT mit diesen Metamodelländerungen umgehen müssen und welche Schritte notwendig sind.

4.3 Klärung der Konformität zum Standardvorgehensmodell

Konformität ist gemäß ISO/IEC 9126 [Bh09] (mittlerweile aufgegangen in der ISO/IEC 25000 [SE05]) ein Qualitätsmerkmal um Softwarequalität zu messen und damit die *„Fähigkeit des Softwareprodukts, Standards, Konventionen oder gesetzliche Bestimmungen und ähnliche Vorschriften [...] einzuhalten.“* Die Sicherstellung der Konformität eines organisationsspezifischen Vorgehensmodells zum Standardvorgehensmodell ist hierbei die Herausforderung. Schwierigkeiten bereiten hierbei nicht nur etwaig erforderliche Anpassungen zur Herstellung der Konformität. Denn die Grenze zwischen konform und nicht konform ist fließend. Die Begutachtung muss von erfahrenen Prozessingenieuren durchgeführt werden.

Um als V-Modell-XT-konformes Vorgehensmodell zu gelten, müssen Vorgehensmodelle laut Konformitätskonzept einen Abdeckungsgrad von 85% in Bezug auf die Inhalte des V-Modell XT aufweisen [VM05]. Eine Gewichtung ist hierbei nicht vorgesehen. Als Ansatz zur Objektivierung sind die Fragenkataloge des Konformitätskonzepts unbestritten. Dennoch kann ein ermittelter Abdeckungsgrad höchstens als Näherung gelten.

4.4 Abhängigkeiten von der Werkzeuglandschaft

Weitere Herausforderungen kommen hinzu, falls das Vorgehensmodell durch Werkzeuge unterstützt wird. Dabei muss man im Wesentlichen zwei Arten der Werkzeugintegration betrachten. Einerseits kann es Werkzeuge geben, die das Standardvorgehensmodell unterstützen und im Rahmen des Anpassungsprozesses an organisationspezifische Besonderheiten ebenfalls integriert bzw. angepasst werden müssen. Diese Anpassungen können sowohl das Design (Bilder, Farben etc.), technische Rahmenbedingungen (Betriebssysteme, Firewalls, Rechtevergabe etc.) und die tatsächliche Funktionalität des Werkzeuges betreffen. Andererseits kann es Werkzeuge geben, die sich bereits seit Jahren in einer Organisation etabliert haben. Hier ist die Herausforderung diese Werkzeuge in die organisationspezifische Anpassung mit einzubeziehen [KKT12].

4.5 Rückfluss zum Vorgehensmodellstandard

Eingeführte Änderungen wie die in Abschnitt 3.2 genannte Umbenennung der Rolle *Projektmanager* in *Projekteigner* müssen nicht immer als rein organisationsspezifische Anpassung verstanden werden. Je nach Zusammenhang kann es sich bei einer Änderung um eine Verbesserung handeln, die für das Standardvorgehensmodell einen Wert besitzen könnte.

Tatsächlich sind einige Anpassungen aus dem V-Modell XT Bund 1.0 wieder bei der Weiterentwicklung des V-Modell XT 1.4 berücksichtigt und integriert worden. Die Herausforderung hierbei ist zum Einen der Kommunikationsaufwand zwischen den beteiligten Prozessingenieuren, die Beurteilung der Übertragbarkeit von Änderungen und dann im Anschluss auch die Klärung der Frage, wie etwaig übernommene Neuerungen sich auf die Beschaffenheit der nächsten Version des organisationsspezifischen Vorgehensmodells auswirken. Wird die Änderung beispielsweise nicht 1:1 sondern angepasst übernommen, dann muss für das organisationsspezifische Vorgehensmodell entschieden werden, wie mit dieser Situation umzugehen ist.

5 Zusammenfassung und Bewertung

In diesem Beitrag haben wir Herausforderungen und Mehrwerte organisationsspezifischer Anpassungen von Vorgehensmodellen diskutiert. Dabei wurden Beispiele und Möglichkeiten von Anpassungen für Vorgehensmodelle genannt. Organisationsspezifische Vorgehensmodelle besitzen gegenüber einem Standardvorgehensmodell eine Reihe von Vorzügen. Mit diesen gehen aber auch Herausforderungen einher. Für den Umgang mit diesen Herausforderungen müssen Lösungsstrategien entwickelt werden, um den vollen Nutzen aus einem organisationsspezifischen Vorgehensmodell ziehen zu können.

Für die drei erstgenannten Herausforderungen wurden im Umfeld des V-Modell XT mit Version 1.3 bereits Lösungsansätze entwickelt und in der Praxis umgesetzt.

1. Seit 2009 wurden Anpassungen des V-Modell XT entwickelt, die auf dem mit Release 1.3 eingeführten Erweiterungsmechanismus basieren. Zum Beispiel konnte das V-Modell XT Bayern nach der Veröffentlichung des Release 1.4 des V-Modell XT (Referenzmodells) mit überschaubarem Aufwand durch den WEIT e.V. auf den aktuellen Entwicklungsstand gebracht werden.
2. Nach der Veröffentlichung des V-Modell XT 1.4 wurde von einem der Autoren eine Verfahrensbeschreibung einschließlich Transformationskript für den Aufstieg von V-Modell XT Varianten in Version 1.3 auf die damals neue Version 1.4 veröffentlicht [Te12], die ein geändertes Metamodell hatte.
3. Die Standardversion des V-Modell XT bietet im Rahmen des Erweiterungsmechanismus eine Reihe von Änderungsoperationen für die organisationsspezifische Anpassung an. Die Auswahl der zur Verfügung gestellten Änderungsoperationen wurde bewusst eng gefasst, um die Anforderungen an die im Konfor-

mitätskonzept des V-Modell XT [VM05] beschriebene „konstruktive Konformität“ erreichen zu können, also die Erreichung eines konformen organisations-spezifischen Vorgehensmodells unter Zuhilfenahme ausschließlich der freigegebenen Änderungsmechanismen. In der Praxis, etwa beim V-Modell XT Bund, zeichnet sich jedoch ab, dass eine größere Auswahl an Änderungsoperationen benötigt wird. Auf eine händische Überprüfung der Inhalte wird daher so bald nicht verzichtet werden können, allerdings lässt sich der Prozess einer Konformitätsprüfung bei Anwendung einer abgesteckten Menge von Änderungsoperationen dadurch relativ leicht werkzeuggestützt durchführen.

Literaturverzeichnis

- [Ar08] Armbrust, O.: Leitfaden zur Modelleinführung im Rahmen der organisations-spezifischen Anpassung des V-Modell XT. Technical Report 013.08/D, Institut für Experimentelles Software Engineering (IESE), März 2008.
- [Bh09] Balzert, H.: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Springer, 2009.
- [Br96] Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology, 1996.
- [Br99] Brinkkemper, S.; Saeki, M.; Harmsen, F.: Meta-Modeling Based Assembly Techniques for Situational Method Engineering. Information Systems, 1999.
- [He06] Henderson-Sellers, B.: Method Engineering: Theory and Practice. Information Systems Technology and its Applications, 2006.
- [HPW86] Horváth, P.; Petsch, M.; Weihe, M.: Standard-Anwendungssoftware für das Rechnungswesen. 2. Auflage. München : Vahlen, 1986.
- [KKT12] Kuhrmann, M.; Kalus, G.; Then, M.: The Process Enactment Tool Framework - Transformation of Software Process Models to Prepare Enactment. In Science of Computer Programming, Elsevier, 2012
- [KMS13] Kuhrmann, M.; Méndez Fernández, D.; Steenweg, R.: Systematic Software Process Development: Where Do We Stand Today? In: Proceedings of the International Conference on Software and System Process, (to appear) 2013.
- [KMT13] Kuhrmann, M.; Méndez Fernández, D.; Tießler, M.: A Mapping Study on Method Engineering - First Results. In Proceedings of the 17th Evaluation and Assessment in Software Engineering (EASE 2013), ACM Press, 2013.
- [KTF11] Kuhrmann, M.; Ternité, T.; Friedrich, J.: Das V-Modell XT anpassen. Springer, 2011.
- [Ku07] Kuhrmann, M.; Hammerschall, U.; Ternité, T.; Sihling, M.: Individueller Standard - V-Modell XT an Unternehmen anpassen (Teil 1). iX-Magazin für professionelle Informationstechnik, 2007.
- [Ku08] Kuhrmann, M.: Konstruktion Modularer Vorgehensmodelle. Doktorarbeit, Technische Universität München, 2008.
- [Lw12] Lanninger, W.: Customizing von Standardsoftware. In (K. Kurbel et al. Hrsg.): Enzyklopädie der Wirtschaftsinformatik (Online-Lexikon), München: Oldenburg, 2012.
- [NZ12] Niazi, M.; Zahran, S.: Software Process Lines: A Step towards Software Industrialization. IGI Global, 2012.
- [OC07] Ocampo, A.; Soto, M.: Connecting the Rationale for Changes to the Evolution of a Process. In Proceedings of the International Conference on Product-Focused Software Process Improvement, 2007.

- [Oc09a] Ocampo, A.; Münch, J.; Riddle, W.: Incrementally Introducing Process Model Rationale Support in an Organization. In: Proceedings of the International Conference on Software Process, 2009.
- [Oc09b] Ocampo, A.; Münch, J.: Rationale Modeling for Software Process Evolution. Software Process, 2009.
- [Ro05] Rombach, D.: Integrated Software Process and Product Lines. In: Unifying the Software Process Spectrum, LNCS, Springer, 2005.
- [Ru12] Ruiz-Rube, I.; Doderer, J. M.; Palomo-Duarte, M.; Ruiz, M.; Gawn, D.: Uses and Applications of SPEM Process Models. A Systematic Mapping Study. Journal of Software Maintenance and Evolution: Research and Practice, 2012.
- [SE05] Software-Engineering - Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) - Leitfaden für SQuaRE (ISO/IEC 25000:2005)
- [SEI05] SEI: Software Product Lines. Last visit: 2013-05-05. [Online]. Available: <http://www.sei.cmu.edu/productlines/>.
- [SKM12] Steenweg, R.; Kuhrmann, M.; Méndez Fernández, D.: Software Engineering Process Metamodels: A Literature Review. Technischer Bericht, Technische Universität München, 2012.
- [Te10] Ternité, T.: Variability of Development Models - An approach for the adaptation of development models. Doktorarbeit, Technische Universität Clausthal, 2010.
- [Te12] Ternité, T.: Migration organisationsspezifisches V-Modell 1.3 nach 1.4, vmxt.blogspot.de, 2012. Last visit: 2013-07-23. [Online]. Available: <http://vmxt.blogspot.de/2012/10/migration-organisationsspezifisches-v.html>
- [VM05] V-Modell XT Konformität V1.1. Last visit: 2013-05-05. [Online]. Available: <http://www.weit-verein.de/Konformit%C3%A4tszertifikate.html>

Ein Konzept zur Harmonisierung der Vorgehensmodelle für die IT-Steuerung Bund

Thomas Diefenbach, Michael Tonndorf

CSC Deutschland Solutions GmbH
Ettore-Bugatti-Str. 6-14. 51149 Köln
thomas.diefenbach@csc.com
CSC Deutschland Solutions GmbH
Barthstr. 4, 80339 München
michael.tonndorf@csc.com

Abstract: Die effiziente Entwicklung und Pflege leistungsfähiger IT-Lösungen zur Unterstützung komplexer Geschäftsprozesse ist für die öffentliche Verwaltung ein entscheidender Erfolgsfaktor. Die Herausforderungen bestehen darin, eine moderne, effiziente und kundenfreundliche Verwaltung unter der Randbedingung des demografischen Wandels sowie knapper Haushaltsmittel weiterzuentwickeln. Das von Bundestag und Bundesrat im Juni 2013 verabschiedete E-Government-Gesetz ist am 1. August 2013 in Kraft getreten und verpflichtet die Verwaltungen, vor der Einführung von elektronischen Verfahren Signaturanforderungen zu verifizieren und ihre Geschäftsprozesse zu optimieren, bevor die IT-Unterstützung für die optimierten Verwaltungsverfahren realisiert werden kann.

Um die erforderlichen IT-Projekte erfolgreich zu steuern, wurde das V-Modell XT und seine Vorgängerversionen entwickelt, seit den 90er-Jahren als Standard propagiert und regelmäßig neuen Entwicklungen der Softwaretechnik angepasst. Parallel hierzu ist in den letzten Jahren immer deutlicher geworden, dass mit den Möglichkeiten der IT-Unterstützung auch der Umgang mit den Geschäftsprozessen weiterentwickelt werden muss, um Potenziale zur Effizienz- und Qualitätssteigerung in der Verwaltung zu nutzen. Dazu wurden, initiiert u.a. durch die Bundesverwaltung, die Methoden des Geschäftsprozessmanagements (GPM) systematisch ausgebaut und zunehmend in der Praxis verankert.

Als weiterer Baustein für Effizienzverbesserungen erfolgte mit der Einrichtung der Bundesbeauftragten für IT und des Rates der IT-Beauftragten die Etablierung ressortübergreifender Maßnahmen zur IT-Steuerung und in der Folge der Aufbau von Enterprise Architecture Management (Architekturmanagement, kurz EAM) in den Ressorts. Die EAM-Funktionen tragen zur Standardisierung und Konsolidierung der IT und der Erzielung von Skaleneffekten u.a. im IT-Betrieb und bei der Softwareentwicklung bei. Unter dem Dach der Rahmenarchitektur IT-Steuerung Bund wurden Methoden und Modelle zur Steuerung der IT entwickelt.

Um das Nutzenpotenzial aus den Initiativen zum Geschäftsprozessmanagement (z.B. prozessorientierte Ressourcensteuerung, Prozess- und IT-Verzahnung) und EAM (z.B. Zielarchitekturen) systematisch in die laufenden Veränderungsprozesse

entlang von IT-Projekten einzubringen, ist es notwendig, die Methoden und Ergebnisse des GPM und EAM mit dem Vorgehen für IT-Projekte nach V-Modell XT zu verknüpfen. Durch eine Vereinheitlichung der Vorgehensmodelle aus einer verfahrensübergreifenden, fachlichen und technischen Gesamtsicht soll die auf ein einzelnes IT-Verfahren begrenzte Sicht des V-Modells XT erweitert werden.

In diesem Dokument werden Vorschläge zur Ergänzungen des V-Modells XT hergeleitet, die eine geschäftsprozessorientierte Entwicklung von IT-Lösungen unter Berücksichtigung von EAM auf der Basis der Rahmenarchitektur IT-Steuerung Bund unterstützen. Dazu gehört eine neue Projektdurchführungsstrategie mit den zusätzlichen, optionalen Vorgehensbausteinen Prozess- und Dienstemanagement sowie Lösungsintegration. Die zusätzlichen Vorgehensbausteine klammern, wie in der Struktur des V-Modells XT vorgesehen, spezifische Typen von Produkten, Aktivitäten und Rollen. Die vorgeschlagenen Erweiterungen schaffen eine Grundlage für die Harmonisierung und das Zusammenspiel der einzelnen Vorgehensmodelle und erhöhen damit gleichzeitig ihren Nutzen. So trägt dieser Ansatz zur Bewältigung wichtiger Herausforderungen der öffentlichen Verwaltung, wie der Umsetzung des E-Government-Gesetzes bei.

1 Einleitung

Die Öffentliche Verwaltung in Deutschland steht vor der Herausforderung, ihre Aufgaben als Dienstleister für Bürger, Wirtschaft und politische Entscheidungsträger unter sich schnell verändernden Rahmenbedingungen, Gesetzen sowie den Erwartungen einer modernen Informationsgesellschaft zu erfüllen. Sie muss gleichermaßen flexibel, zuverlässig und kosteneffizient sein. Die schnelle und effiziente Realisierung und Anpassung von Geschäftsprozessen sind daher entscheidende Erfolgskriterien für die Öffentliche Verwaltung. Voraussetzung dafür ist eine leistungsfähige und flexible Informationstechnik. Diese muss dabei u.a. die Öffnung der Verwaltung gegenüber der elektronischen Kommunikation (E-Government) mit ihren Kunden ermöglichen. Beide „Säulen“ – Geschäftsprozesse und IT – sind Gegenstand aktueller Initiativen der Öffentlichen Verwaltung. Es gilt, diese Säulen eng miteinander zu verzahnen, um den erwarteten Nutzen nachhaltig sicherzustellen. Auf die Schaffung der notwendigen Voraussetzungen mit Geschäftsprozessmanagement und einer übergreifenden IT-Steuerung folgt die Herausforderung der operativen Umsetzung der Ergebnisse beider Seiten innerhalb der einzelnen Projekte, mit denen die Verwaltung ihre Geschäftsprozesse mit IT-Unterstützung realisiert. Dabei wird davon ausgegangen, dass der eingeführte Entwicklungsstandard V-Modell XT zur Durchführung von IT-Projekten zugrunde gelegt wird, dessen Anpassung an diese neuen Voraussetzungen im Folgenden näher vorgestellt wird.

Die ausführliche Fassung des hier dargestellten **Erweiterungsmodells** zum V-Modell XT ist in einem internen Projektbericht für das Bundesministerium des Innern dokumentiert, der im Rahmen des Programms E-Government des Bundes 2012/2013 entstanden

ist (im Folgenden als *Referenzdokument* zitiert). Der Bericht befindet sich zum Zeitpunkt der Erstellung dieses Konferenzbeitrags in der Begutachtung durch das Bundesministerium des Innern und den Verein WEIT e.V. Für deren hilfreiche Unterstützung sei an dieser Stelle ausdrücklich gedankt.

2 Geschäftsprozessmanagement und IT-Steuerung

2.1 Regierungsprogramm und E-Government-Gesetz

Das Thema „**Optimierung der Geschäftsprozesse** für die Öffentliche Verwaltung“ ist ein Schwerpunkt des Regierungsprogramms *Vernetzte und transparente Verwaltung* 2010 – 2013 sowie einer Vielzahl von Initiativen auf Bundesebene seit über 10 Jahren. Hierzu gehören u.a. die Konzeption und Einrichtung von Kompetenz- und Dienstleistungszentren (*Shared Servicecenter*), die Entwicklung von Richtlinien, einheitlichen Standards und Methoden und die Einrichtung eines Kompetenzzentrums im Bundesverwaltungsamt zum Geschäftsprozessmanagement. Die Initiative *Nationale Prozessbibliothek* verfolgt den Aufbau einer deutschlandweiten Bibliothek an Verwaltungsprozessen.

Als zentrale Initiative des Gesetzgebers zur Verwaltungsmodernisierung ist das **E-Government Gesetz (EGovG)** zu nennen. Im September 2012 hat die deutsche Bundesregierung den Entwurf des E-Government-Gesetzes verabschiedet (EGovG; „Gesetz zur Förderung der elektronischen Verwaltung“). Die Inkraftsetzung ist am 1. August 2013 erfolgt. Zu den Regelungen des EGovG gehören u.a. Grundsätze der elektronischen Aktenführung und des ersetzenden Scannens sowie die Verpflichtung zur Dokumentation und Analyse von Geschäftsprozessen.

Mit der Einführung von **Geschäftsprozessmanagement** als wesentliches **Organisationskonzept** sind die folgenden Ziele und Erwartungen verbunden:

- Einheitlichkeit der Erhebung, Darstellung und Analyse von Prozessen,
- Ermöglichung von Prozessvergleichen (einschließlich Prozessstandardisierung),
- Transparenz bezüglich erfolgreicher Praxislösungen inkl. einer übergreifenden Plattform für Prozesse/Prozessportale,
- bessere Interoperabilität der IT-Systeme.

2.2 Konzept IT-Steuerung Bund

Der **Rolle der IT** als kritische Infrastruktur für Deutschland und die Öffentliche Verwaltung wurde mit dem im Jahr 2007 beschlossenen *Konzept IT-Steuerung Bund* (siehe [BM07]) und den daraus resultierenden Initiativen und Maßnahmen besonders Rechnung getragen. Mit der Schaffung der Rolle *Bundesbeauftragte(r) für IT*, den Beauftragten für IT in den einzelnen Ressorts sowie des Rats der IT-Beauftragten (IT-Rat) wurden die Weichen für eine behörden- und ressortübergreifende IT-Planung gestellt. Dazu gehörte auch die Bildung von Dienstleistungszentren IT (DLZ-IT) der Bundesverwaltung, um das IT-Angebot auf Dauer effizienter und zugleich effektiver bereitzustellen. Zur ressortübergreifenden IT-Planung wurden Architekturmanagementfunktionen aufgebaut und mit der Rahmenarchitektur IT-Steuerung Bund ein gemeinsamer methodisch-konzeptioneller Rahmen für die IT-Planung geschaffen [BM09]). In den einzelnen Ressorts der Bundesverwaltung und insbesondere den DLZ-IT wurden in den letzten Jahren Architekturmanagementfunktionen aufgebaut, die Methoden des Enterprise Architecture Managements zur Standardisierung und bedarfsgerechten Ausrichtung ihrer IT-Landschaft anwenden.

2.3 V-Modell XT

Für die Bundesverwaltung ist das V-Modell XT und die spezielle Variante für die Bundesverwaltung, V-Modell XT Bund, (siehe [CI12] und [CI10]), das **Standard-Vorgehensmodell** für **IT-Projekte**. Es ist in Abhängigkeit von einem spezifischen Projektprofil für ein Projekt auszuprägen (**Tailoring**). Im Kern orientiert sich ein Projekt nach V-Modell XT an den fachlichen Anforderungen einer konkreten Aufgabenstellung. Behördenweite oder gar –übergreifende Vorgaben zur Planung der künftigen IT-Landschaft werden dabei in der Praxis wenig berücksichtigt, da hier kaum explizite Vorgehensweisen zu deren Integration in das Projektvorgehen enthalten sind. Auch geht das V-Modell XT nicht auf die methodische Einbindung von Geschäftsprozessmodellen und die systematische Ableitung von Anforderungen z.B. im Sinne des **Service-Paradigmas** ein. Um hier eine bessere Verzahnung der Methoden in IT-Projekten zu erreichen, wird das V-Modell XT als Dreh- und Angelpunkt gesehen, über den die Methoden und Ergebnisse von GPM und EAM im Sinne der Rahmenarchitektur IT-Steuerung Bund wirkungsvoll integriert werden können.

Eine kritische Diskussion in Bezug auf das hier aufgezeigte **Verbesserungspotential** des V-Modells XT hat in der Öffentlichkeit bisher kaum stattgefunden. Aus der aktuellen Projektpraxis im Kontext der Bundesverwaltung wird jedoch ein zunehmender Handlungsbedarf gesehen, soll das V-Modell XT auch in Zukunft seiner Rolle als führendes

und akzeptiertes Rahmenwerk zur Entwicklung von IT-Lösungen im Öffentlichen Sektor gerecht werden.

3 Vorgehensmodelle – Ausgangslage und Handlungsbedarf

Auf Grundlage der einleitenden Betrachtungen ist festzuhalten, dass die Modellierung und das Management sowie eine effektive IT-Unterstützung von Geschäftsprozessen **Schlüsseltechniken** bei der Verwaltungsmodernisierung darstellen. Damit rückt die Frage nach den zugehörigen **methodischen Ansätzen** einschließlich einer geeigneten Arbeits-/ bzw. Werkzeugumgebung in den Vordergrund. Von den Geschäftsprozessen ist die Brücke zu einer modernen, durch klar strukturierte Anforderungen und Serviceorientierung geprägten sowie zunehmend standardisierten IT zu schlagen. Dieser Brückenschlag erfolgt zum einen bei der übergreifenden Abstimmung von Nachfrage und IT-Angebot aufeinander und zum anderen bei der Entwicklung von IT-Lösungen zur Umsetzung von Geschäftsprozessen.

3.1 Vorgehensmodelle für die Entwicklung von IT-Lösungen

Die Zielsetzung, IT-Lösungen auf der Grundlage von definierten Vorgehensmodellen zu entwickeln, in denen Fachseite und IT aufeinander abgestimmt sind, wird bereits durch die in Abbildung 1 dargestellten Konzepte und Vorgehensmodelle individuell unterstützt. In den gelben Ellipsen sind die jeweiligen Aufgaben, in den darüber stehenden roten Rechtecken sind die zugeordneten Vorgehensmodelle widergegeben.

Die **Prozessmodellierung** nach dem BMI¹-Rahmenwerk dient zur initialen Modellierung von Geschäftsprozessen bis hin zur Erstellung von Sollprozessen mit dem Ziel einer möglichst vollständigen Erfassung der Geschäftsprozesse einer Organisation, strukturiert nach dem Ebenenkonzept und übersichtlich dargestellt in einer Prozesslandkarte.

¹ Bundesministerium des Innern

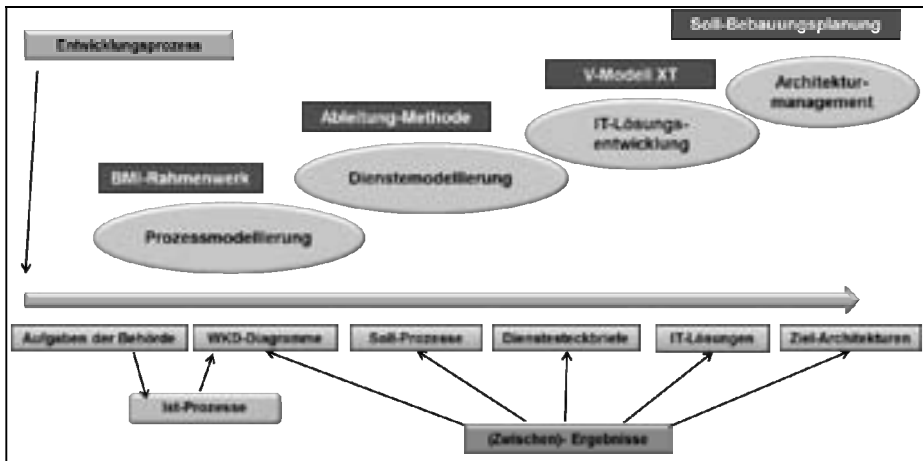


Abbildung 1: Vorgehensmodelle mit Einfluss auf IT-unterstützte Geschäftsprozesse sowie die IT-Landschaft der betrachteten Organisation

Das Vorgehensmodell zur **Ableitung** von **Diensten** aus **Geschäftsprozessen** schlägt eine erste Brücke zwischen den Geschäftsprozessmodellen und der Strukturierung von Anforderungen in Form von logischen Diensten ([BM10]). Die Methodik stammt aus dem Konzept **IT-Steuerung Bund**, für die das Bundesministerium des Innern die Verantwortung trägt. Weiterhin wurden in der IT-Steuerung Bund Modelle zur Beschreibung der IT-Landschaft inkl. einer Zielarchitektur entwickelt – die **Rahmenarchitektur IT-Steuerung Bund**. Diese berücksichtigt bereits neben den fachlichen Anforderungen, die in Form von Diensten strukturiert werden, auch die technischen und architektonischen Anforderungen des IT-Betreibers (Rechenzentrum).

Das **V-Modell XT** dient dazu, eine IT-Lösung z.B. zur Unterstützung bestimmter Geschäftsprozesse zu erstellen. Zum Kontext des V-Modells XT gehören zunächst die zu modellierenden **Geschäftsprozesse** und fachlichen **Anforderungen**. So nennt **SAGA² 5** (siehe [CI11], „Verbindliche IT-Spezifikationen der Bundesverwaltung“) als Methoden zur Geschäftsprozessmodellierung *UML 2.x*, *Flussdiagramme*, *BPMN 1.1/1.2*, *EPK (empfohlen)*, *BPMN 2.0 (in Beobachtung)*. Diese Methoden sind sehr stark an einer bestimmten Notation für Geschäftsprozessmodelle ausgerichtet.

Das **Architekturmanagement** (Enterprise Architecture Management, EAM) entwickelt projekt- und behördenübergreifende Vorgaben für die Gestaltung der künftigen IT-Landschaft (Zielarchitekturen mit Standards). Methodisch kann dies als ein (übergrei-

² Zusammenstellung von Referenzen auf Spezifikationen und Methoden für Software-Systeme der öffentlichen Verwaltung (Eigenname, der ursprünglich als Abkürzung von „Standards und Architekturen für E-Government-Anwendungen“ eingeführt wurde)

fender) Planungsprozess, der auch als **Sollbebauungsplanung** bezeichnet wird, verstanden werden. Um die Ergebnisse der Sollbebauungsplanung in einer gut nutzbaren, verständlichen Form mit den unterschiedlichen Stakeholdern zu erarbeiten, wurde bereits eine Reihe von sogenannten EAM-Frameworks entwickelt. Dazu gehören u.a. die im öffentlichen Bereich besonders genutzten Frameworks TOGAF [Op11] und FEAF [OM12]. Diese wurden stark von Zachman (siehe z.B. [Za04]) beeinflusst. Einen systematischen Ansatz zur Abteilung eines Metamodells, das den Informations- und Entscheidungsbedarf der Verantwortlichen für die IT-Steuerung eines Unternehmens widerspiegelt, liefern Arbeiten der TU München (siehe z.B. [BU10]). Diese methodischen Ansätze wurden auch beim Entwurf der Rahmenarchitektur IT-Steuerung Bund und dem Aufbau von Architekturmanagement-Funktionen berücksichtigt.

Die gängigen **EAM-Frameworks** liefern Modelle zur Beschreibung einer IT-Landschaft im Ist- und Soll-Zustand sowie Vorgehensweisen, wie eine Ziel-Architektur entwickelt wird. Die Rahmenarchitektur IT-Steuerung Bund stellt einen auf die Bedürfnisse der Bundesverwaltung ausgerichteten Ansatz hierzu dar. Das Vorgehen zur systematischen Berücksichtigung der entstandenen Ziel-Architektur in der Durchführung von IT-Projekten steht jedoch hierbei ebenso wenig wie bei den genannten Frameworks im Zentrum. Die **Dienstmodellierung** mit der Methode zur Ableitung von Diensten aus Geschäftsprozessen (siehe [BM10]) schafft eine erste Basis, um die fachlichen Anforderungen aus einem Geschäftsprozess gegenüber einem Referenzmodell der IT-Organisation, dem Dienstekatalog, abzugleichen und damit Gemeinsamkeiten zu Anforderungen anderer Geschäftsprozesse und Potenzial zur Wiederverwendung bzw. Nutzung von Standardkomponenten zu erkennen. Die Anwendung dieser Methodik ist jedoch noch nicht fest in den führenden Vorgehensmodellen der Bundesverwaltung verankert.

Im Rahmen des Projekts Aufbau Geschäftsprozessmanagement des BMI wurde 2011 das Rahmenwerk für ein einheitliches Prozessmanagement entwickelt (siehe [BM11B]). Im Zusammenhang mit dem Konventionenhandbuch für eine einheitliche Prozessmodellierung [BM11a] ist so eine methodische Grundlage für Initialisierung und Durchführung von Projekten zum Geschäftsprozessmanagement der Bundesverwaltung gegeben. Die im Rahmenwerk dokumentierte **SDN-Methode** stellt ein Vorgehensmodell für Projekte zum Geschäftsprozessmanagement (GPM) dar. Eine Anbindung an das V-Modell XT existiert bislang nicht.

Es gilt nun, diese Vorgehensmodelle **aufeinander abzustimmen** und in eine konsistente **Gesamtsicht** zu überführen, die mit einer neuen **Durchgängigkeit** der Vorgehensweisen weiteren Mehrwert für die Fach- und die IT-Seite liefert. Mit einigen Ergänzungen und Anpassungen (z.B. Einführung weiterer Artefakte bzw. Produkttypen) soll eine konsis-

tente Basis für eine einheitliche und auf den genannten Zweck ausgerichtete Fortschreibung des V-Modells XT gelegt werden. Das V-Modell XT muss künftig neben den unmittelbaren fachlichen Anforderungen des Geschäftsprozesses auch den Anforderungen aus den übergreifenden Managementaufgaben gerecht werden. Die Entwicklung einer IT-Lösung von den ersten Ansätzen der Geschäftsprozessmodellierung bis hin zu einem lauffähigen System in einer serviceorientierten Architektur ist ein komplexer und mehrstufiger Prozess. Diese Aussage trifft insbesondere für IT-Lösungen in der **Rahmenarchitektur IT-Steuerung Bund** zu, die sowohl ressortübergreifende als auch ressortspezifische Vorgaben aus der Sollbebauungsplanung erfüllen müssen. Es gibt bis heute kein Vorgehensmodell, das diesen Prozess in seiner Gesamtheit ohne Methodenbruch unterstützt und abdeckt. Vielmehr existieren zum Zeitpunkt der Erstellung dieser Untersuchung in der Governance der Bundesverwaltung **vier Vorgehensmodelle bzw. Rahmenwerke**, die sich jeweils auf einen bestimmten Aspekt der Entwicklung und Weiterentwicklung der IT beziehen. Um hier Abhilfe zu schaffen, liefert die durchgeführte Untersuchung einen integrierten Lösungsansatz und betritt damit Neuland für die Bundesverwaltung.

Als **Erweiterungsmodell** zum V-Modell XT wird der bisher vom V-Modell XT abgedeckte Bereich (Realisierung einer IT-Lösung als *System* auf Grundlage eines Lastenhefts) erweitert um die Abschnitte Prozess- und Dienstmodellierung und Abstimmung auf sowie Integration in eine Zielarchitektur (geplante IT-Landschaft). Dabei werden die charakteristischen Merkmale von IT-Lösungen in der Rahmenarchitektur IT-Steuerung Bund aufgegriffen:

- **Geschäftsprozesse** sind Ausgangspunkt der Entwicklung einer IT-Lösung,
- **Funktionalitäten / Dienste** werden durch Dienstesteckbriefe lösungsübergreifend transparent,
- Dienstesteckbriefe und Lastenhefte sind die Grundlagen der **Anforderungsdokumente**, die den weiteren Verlauf der **Entwicklung der IT-Lösung** steuern,
- IT-Lösungen sind konform zu einer **Zielarchitektur** zu entwickeln und in diese schließlich zu integrieren (bisher im V-Modell XT nicht vorgesehene Anforderungen und Randbedingungen sind zu betrachten).

Die Ausgangslage wird durch die Vorarbeiten bestimmt, die dieser Untersuchung zu Grunde liegen. Diese wurden im Wesentlichen im Projekt *Aufbau koordiniertes Prozessmanagement für das Bundesministerium des Innern und seinem Geschäftsbereich 2010-2011* geleistet. In Abbildung 2 ist das Zusammenspiel der beteiligten Aufgaben, Architekturen und Modellen dargestellt.

Zusammenspiel von Aufgaben, Architekturen und Modellen

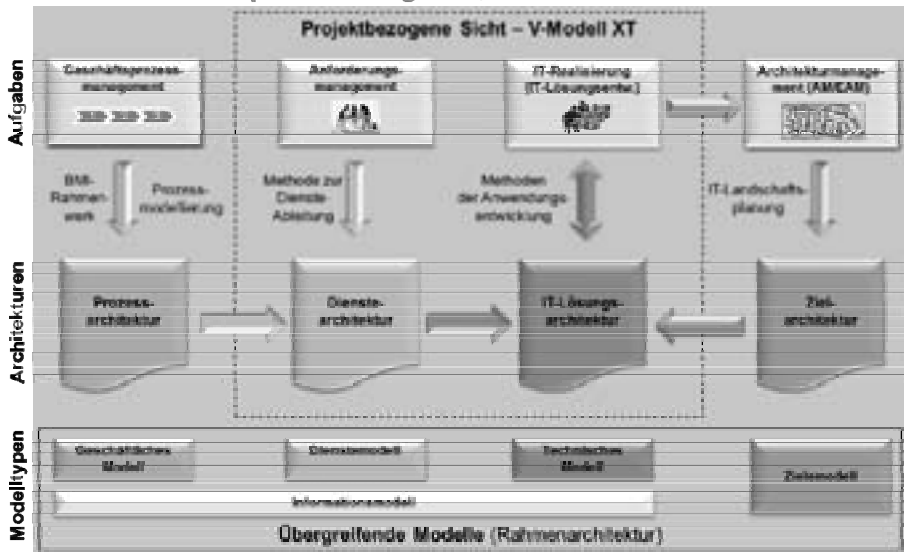


Abbildung 2: Aufgaben, Architekturen und Modelle

3.2 Vorgehensweise

Das V-Modell XT ist der vorgegebene Standard zur Erstellung von IT-Systemen im Öffentlichen Sektor. Die Begriffswelt des V-Modells ist zentral und findet, speziell in der Zusammenarbeit zwischen der Öffentlichen Verwaltung (Auftraggeber) und der IT-Industrie (Auftragnehmer) mittlerweile breite Anwendung. Es ist **daher naheliegend, dass die Terminologie des V-Modells XT als die führende Terminologie in dieser Untersuchung** zu Grunde gelegt wird.

Die wesentlichen Beschreibungselemente des V-Modells XT wurden mit der Neuauflage des V-Modells '97 als V-Modell XT entwickelt und haben sich seither in einer Vielzahl von Projekten bewährt. Diese Elemente bilden

- die **statische** Struktur des V-Modells XT, die durch die **Vorgehensbausteine** vorgegeben ist. Ein Vorgehensbaustein kapselt zusammengehörige Produkte, Aktivitäten und Rollen zu einem bestimmten Teilbereich der IT-Entwicklung (z.B. *Projektmanagement*);
- die **dynamische** Struktur, die mittels **Ablaufbausteinen** beschrieben wird. Ein Ablaufbaustein ist eine Sequenz von zwei oder mehreren Entscheidungspunkten, die auch Kontrollstrukturen (Verzweigungen, Schleifen, Rücksprünge) enthalten kann. Eine vollständige Sequenz von Ablaufbausteinen konstituiert eine Projektdurchführungsstrategie

Die Beschreibungselemente des V-Modells XT sind so universell, dass sie auch zur Modellierung von statischen und dynamischen **Prozessmodellen** verwendet werden können, die über die eigentliche Domäne des V-Modells XT hinausgehen.

4 Anforderung an die Erweiterung des V-Modells XT

4.1 Anforderungen an das V-Modell XT aus Sicht des Gesamtprojekts

Dem V-Modell XT liegt ein Prozessmodell der IT-Lösungsentwicklung zugrunde, das aus der hier eingenommenen Perspektive neu festzulegen ist. Das V-Modell XT definiert dabei einen „maximalen“ Projekttyp, bei dem im Zuge einer konkreten Anpassung an eine Projektsituation ggf. Produkte und Aktivitäten weggelassen werden (Tailoring).

Im V-Modell XT und V-Modell XT Bund erstreckt sich ein Projekt des Auftraggebers über die in der u.a. Abbildung 3 dargestellten Entscheidungspunkte:

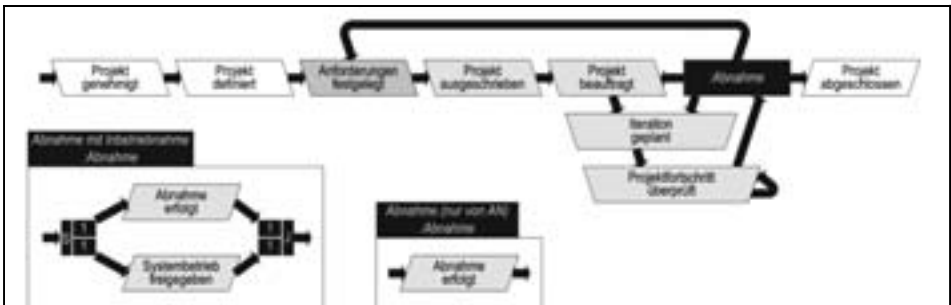


Abbildung 3: Entscheidungspunkte des Projekttyps AG-Projekt mit einem Auftraggeber
(Quelle: V-Modell XT Bund, [CI10])

Dies ist der Projektzyklus eines „klassischen“ IT-Entwicklungsprojekts (Auftraggeber-Sicht). Wenn im Folgenden der Begriff **Projekt** verwendet wird, dann soll darunter ein Projekt des **Erweiterungsmodells** unter Einbeziehung der zusätzlichen Aktivitäten zur Geschäftsmodellierung, Dienstmodellierung und Integration in die IT-Landschaft verstanden werden. Gegenstand des hier vorgestellten Erweiterungsmodells ist es ja gerade, einen Gleichklang mit Begriffen, Strukturen und Notation des V-Modells XT herbeizuführen.

Aufgrund dieser Anforderungen ergeben sich folgende **Vorgaben** an das Erweiterungsmodell zum V-Modell XT:

1. Folgende zur reinen IT-Lösung **externe Produkte** und **Artefakte** sind zu berücksichtigen:
 - **Dienstmodell, Informationsmodell** und **Technisches Modell** als *Referenz*, die darstellt, welche Dienste, Informationsobjekte und Systemelemente (Hardware-/Softwareelemente) im projektübergreifenden Modell auf Organisationsebene bereits vorhanden oder vorgesehen sind,
 - **Zielarchitektur** mit Architekturprinzipien, Anwendungs- und Technologielandkarten sowie Standards.
2. Die genannten externen Produkte und Artefakte werden bei der Spezifikation und dem Entwurf der IT-Lösung genutzt, um als neues Produkt die
 - **Dienstarchitektur** mit **Dienstesteckbriefen** und **Informationsobjekten** inkl. den Beziehungen der Dienste untereinander sowie zu den unterstützten Sollprozessen der Geschäftsarchitekturzu erstellen. Die Dienstarchitektur vermittelt zwischen der fachlichen Sicht auf die Anforderungen und der im weiteren Projektverlauf zu erstellenden technischen Architektur mit den zu verwendenden Komponenten.
3. **Bezeichnungen** von **Produkten** des V-Modells und **Artefakten** des **Gesamtmodells** sind aufeinander abzustimmen und anzupassen.
4. Es sind **Entscheidungspunkte** festzulegen, zu denen eine Überprüfung der Produkte des Projekts (insbesondere der IT-Lösungsarchitektur in ihrem jeweils erreichten Detaillierungsgrad) gegen die Vorgaben der Zielarchitektur erfolgt. Solche Architekturreviews tragen erheblich zur Entwicklung nachhaltiger und auf Dauer besser betreibbarer und pflegbarer IT-Lösungen bei. Zudem wird durch die Sicherstellung der beschriebenen Konformität die Flexibilität bei Änderungen von Anforderungen erhöht, da die gesamte Struktur der langfristig geplanten Basis der IT entspricht (Vermeidung von „Insellösungen“).

Bei der Entwicklung des hier vorgestellten Vorgehensmodells stellt sich ein grundlegendes Unterscheidungsmerkmal zum V-Modell XT (Bund) heraus. Die Projekte nach V-Modell XT (Bund) beginnen auf der „grünen Wiese“ ohne einen Systemkontext und enden mit einem fertig erstellten „System“. Spätestens mit der Veröffentlichung der **Rahmenarchitektur IT-Steuerung Bund** ist diese Sichtweise nicht mehr uneingeschränkt zutreffend. Vielmehr gibt es für Projekte vom hier entwickelten Typ einen

starken **Systemkontext** in Gestalt der **Zielarchitektur**, in die die entwickelte IT-Lösung zu integrieren ist. Die Zielarchitektur muss mit einer Reihe von Dokumenten beschrieben sein, damit das Integrationsziel definiert ist. Demzufolge ergeben sich durch Bereitstellung einer neuen IT-Lösung in der Zielarchitektur Änderungs-/ Ergänzungsanforderungen, die bei der Realisierung umzusetzen sind. Lediglich beim Thema IT-Sicherheitskonzept und Datenschutzkonzept sieht das V-Modell XT Bund bisher eine **inkrementelle Fortschreibung** existierender organisationsweiter Vorgaben vor, nicht aber bei Architektur und Governance.

Beispiel: In einer gemanagten Zielarchitektur ist ein Governance-Konzept in Kraft. Durch die Bereitstellung einer neuen IT-Lösung ergeben sich neue/zusätzliche Anforderungen zur Governance. Diese müssen in einem Projektdokument *Anforderungen an das Governance-Konzept* erfasst werden.

4.2 Überblick zur Vorgehensweise

Die Entwicklung des Erweiterungsmodells beginnt mit der Identifikation der **Entscheidungspunkte** im Gesamtdurchlauf und die **Zuordnung der Produkte** zu den Entscheidungspunkten (**dynamische Struktur**). Entscheidungspunkte („Meilensteine“) gliedern nach V-Modell XT den zeitlichen Ablauf eines Projekts, ihre Festlegung stellt i.d.R. den ersten Planungsschritt des Projektablaufs dar. Es folgen die systematische Auflistung der **Produkte**, die **Zuordnung** von **Rollen** zu den **Produkten** und eine systematische Auflistung und Harmonisierung der beteiligten **Rollen**. Abschließend erfolgt eine Kapselung der Produkte / Aktivitäten und Rollen in einem oder mehreren **Vorgehensbausteinen**. In der vorliegenden Veröffentlichung sind die hier festgelegten Entscheidungspunkte, Produkte, Aktivitäten und Rollen nicht explizit und vollständig dargestellt - dies würde den vorgegebenen Rahmen übersteigen.

4.3 Projektdurchführungsstrategie (dynamisch): Entscheidungspunkte

Um die Gesamtmenge an Produkten, Aktivitäten und Rollen des harmonisierten Vorgehensmodells zu identifizieren, wird ein **gedachter Gesamtdurchlauf** durch ein vollständiges Projekt vollzogen. Ein solcher Durchlauf beginnt bei der Modellierung der Prozesse und führt über die Ableitung von Diensten zu einer ersten Vorstellung vom logischen Aufbau der IT-Lösung. Diese ist dann noch technisch zu spezifizieren, wobei die **Vorgaben** der **Zielarchitektur** der Organisation zu beachten sind. Nach Fertigstellung der IT-Lösungsarchitektur ist diese innerhalb der IT-Landschaft zu positionieren, um die nach der Realisierung und Inbetriebnahme erweiterte Ist-Architektur der Organisation unmittelbar an den aktuellen Stand anpassen zu können. Bei diesem Ablauf dienen Ent-

scheidungspunkte als Strukturierungshilfen, die als Zwischenziele im Gesamtablauf durchlaufen werden.

Zu den Entscheidungspunkten werden jeweils die **vorzulegenden Produkte** definiert. Die Produkte sind so zu gestalten und aufeinander abzustimmen, dass ein konsistentes und durchgängiges **Gesamtbild** der Prozesse, IT-Lösungen und der IT-Landschaft entsteht und erhalten bleibt. Dabei kann die Situation gegeben sein, dass Produkte durch verschiedene Detaillierungsstufen gehen (grob → fein usw.). Für diese Produkte muss eine Vorlage nur einmal erstellt werden.

Dies betrifft die beteiligten Teilprojekte:

- Prozessmodellierung
- Dienstmodellierung,
- IT-Lösungsentwicklung,
- Integration in die IT-Landschaft.

4.4 Produkte des harmonisierten Vorgehensmodells

Die Produkte müssen zu den jeweiligen Entscheidungspunkten vorliegen, damit diese per Entscheidung des Lenkungsgremiums in den Zustand „fertig gestellt“ überführt werden können. Die Produkte müssen die nachfolgenden Anforderungen erfüllen:

- durchgängig einheitliche Produktbezeichnungen,
- durchgängig einheitliche Rollen-Zuordnung,
- kompatibel mit den schon existierenden Produkten der beteiligten Vorgehensmodelle (hierzu ist die Verbindlichkeit und Aktualität der existierenden Standards jeweils zu beachten).

Gemäß V-Modell XT müssen alle Rollen, die für die Produkte verantwortlich sind, eindeutig identifiziert und definiert sein. Weitere *mitwirkende* Rollen können zusätzlich definiert werden.

4.5 Rollenmodell

Die Rollen müssen einheitlich für die gesamte Modellvariante definiert sein, daher sind zu fordern

- einheitliche Rollenbezeichnungen,
- konsistente Produkt-Zuordnung,
- kompatibel mit den schon existierenden Rollen der beteiligten Vorgehensmodelle,
- ggf. Harmonisierungsvorschlag, falls mehrere Stakeholder betroffen sind.

4.6 Projektablauf mit Entscheidungspunkten und Produkten

Auf der Grundlage der dargestellten Rollen und Produkte wird als Ergebnis der Untersuchung der Ablauf eines Projekts im Sinne des V-Modells XT als Projektdurchführungsstrategie **Geschäftsprozessbasierte Entwicklung einer IT-Lösung zur Rahmenarchitektur IT-Steuerung Bund** dargestellt. Nach dem Projektvorlauf, der in der bisherigen Weise abläuft und mit dem Entscheidungspunkt „Projekt definiert“ abschließt, beginnt die Phase der Prozessmodellierung. Hier werden die Geschäftsprozesse analysiert und ein Sollkonzept für die Prozesse erstellt. Parallel wird ein Governance-Konzept erstellt, das Anforderungen an die Steuerung von künftigen Veränderungen an den Geschäftsprozessen und damit der benötigten IT-Unterstützung beinhaltet.

Die Geschäftsprozesse sind später mit übergreifenden Governance-Konzepten und – Prozessen sowohl auf Fach- als auch IT-Seite abzugleichen und notwendige Anpassungen an letzteren zu beschließen. In der Phase der **Dienstmodellierung** geht es darum, die Anforderungen aus den Geschäftsprozessen strukturiert und dem Paradigma der Service-Orientierung folgend in Form von Diensten zu clustern. Die Dienstarchitektur spiegelt die Anforderungen in Form von Diensten wider und bildet eine erste Basis für den Zuschnitt der IT-Lösung. Sie hilft z.B. zu entscheiden, für welche Dienste vorhandene Komponenten genutzt, externe Komponenten beschafft oder Eigenentwicklungen realisiert werden sollen. Diese Entscheidungen werden – wie vom V-Modell XT bereits bisher vorgesehen – in der Phase **IT-Realisierung** getroffen und in der Implementierung umgesetzt. Der Prozess der IT-Realisierung läuft wie gewohnt ab, hat allerdings die hier definierten Produkte (z.B. die Dienstarchitektur) zu berücksichtigen.

Besonderheiten ergeben sich in der abschließenden Phase der **Integration** in die **IT-Landschaft**. Hier wird das neu entwickelte System gegen die Vorgaben der Zielarchitektur für die gesamte IT-Landschaft der Organisation geprüft. Die Integration und der erste Abnahmeschritt aller extern und intern entwickelten Komponenten (als Realisierung der Dienste) erfolgen zunächst in der **Entwicklungsumgebung** (meist eine spezielle Test- und Integrationsumgebung für das Projekt). Ist hier das Funktionieren der integrierten Komponenten gegeben und die Erfüllung der Vorgaben der Zielarchitektur aus

Projektsicht erfüllt, werden die einzelnen Komponenten abgenommen. Im nächsten Schritt ist es notwendig, die Gesamtlösung in der **Zielumgebung** des IT-Betriebs zu installieren und zu testen. Dieser Schritt, in dem auch umfangreiche Tests der nicht-funktionalen (insbesondere betrieblichen) Anforderungen erfolgen, endet mit der **Abnahme** in der **Zielumgebung**. Damit ist die neue IT-Lösung bereit für die Inbetriebnahme bzw. ihren Roll-Out. Abbildung 4 stellt das Modell der Projektdurchführungsstrategie **Geschäftsprozessbasierte Entwicklung einer IT-Lösung zur Rahmenarchitektur IT-Steuerung Bund** im Überblick dar.

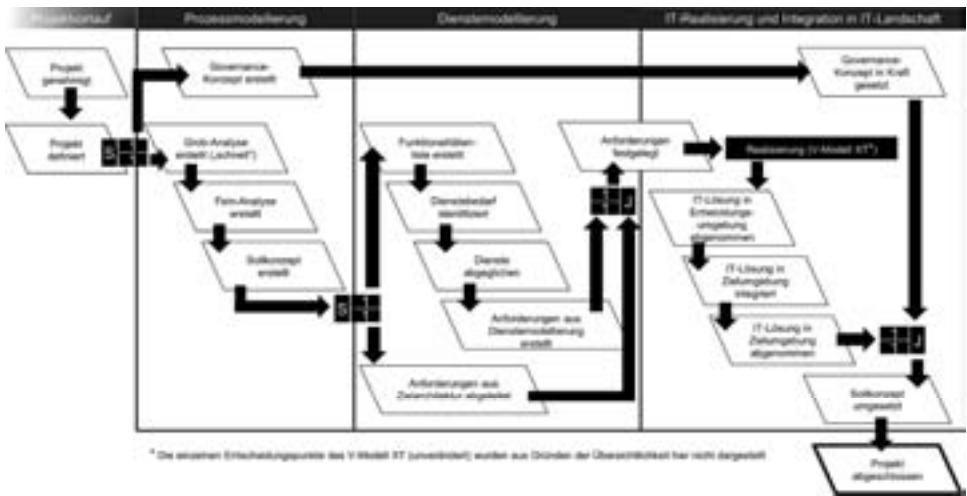


Abbildung 4: Projektablauf in V-Modell XT Notation unter Einbindung von GPM und EAM

In Verbindung mit der Inkraftsetzung von Änderungen an der **Governance** für **Geschäftsprozesse** bzw. **IT-Lösungen** ist die Umsetzung des Sollkonzepts erreicht und das Projekt abgeschlossen.

4.7 Überblick Vorgehensbaustein Prozess- und Dienstmodellierung

Die nachfolgende Abbildung 5 stellt den **Vorgehensbaustein Prozess- und Dienstmodellierung** in der Notation des V-Modells XT dar. Die beteiligten Rollen und Produkte sind im Referenzdokument detailliert beschrieben. Mitwirkende Rollen sind nicht dargestellt.

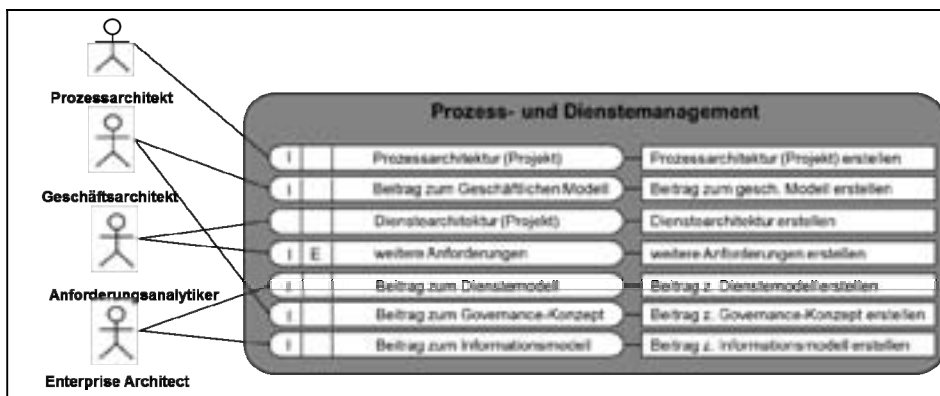


Abbildung 5: Vorgehensbaustein Prozess- und Dienstmodellierung

4.8 Überblick Vorgehensbaustein Lösungsintegration

Die Abbildung 6 auf der folgenden Seite stellt den Vorgehensbaustein **Lösungsintegration** in der Notation des V-Modells XT dar. Die beteiligten Rollen und Produkte sind im Referenzdokument detailliert beschrieben. Mitwirkende Rollen sind nicht dargestellt.

5 Ausblick

Die in diesem Dokument dargestellten Analyseergebnisse sind in **konkrete Vorschläge zur Erweiterung des V-Modell XT** eingegangen. Die Verantwortlichen zur Weiterentwicklung des V-Modells XT (Bund) haben bestätigt, dass der vorliegende Vorschlag bei der Fortschreibung des V-Modells XT bzw. V-Modells XT Bund berücksichtigt werden soll. Dafür sind die Vorschläge weiter zu detaillieren und mit ggf. weiteren Änderungsvorschlägen zu harmonisieren. Parallel sollten **Praxisfälle** identifiziert werden, wo anhand eines konkreten Bedarfs **IT-Lösungen** nach diesem Vorgehen **entwickelt** werden, um Erkenntnisse über die Nutzbarkeit dieses Vorschlags in der Praxis sowie ggf. notwendige Optimierungen daran zu gewinnen.

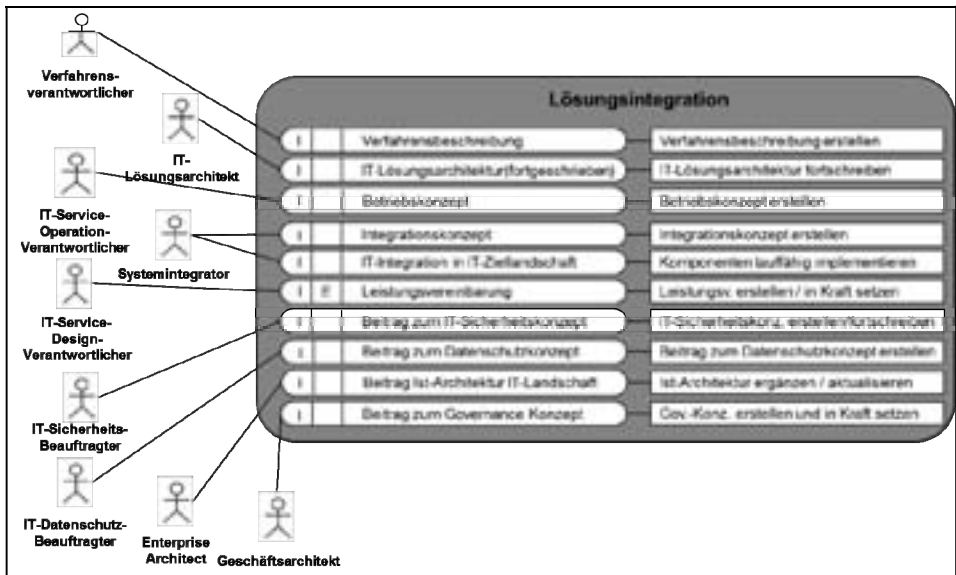


Abbildung 6: Vorgehensbaustein Lösungsintegration

Geeignete Praxisfälle sind z.B. Projekte der **Verwaltungsmodernisierung**, in denen die IT-Unterstützung an neue oder geänderte Geschäftsprozesse angepasst werden muss und zugleich bereits übergreifende **Vorgaben** für die **IT-Architektur** (z.B. eine Zielarchitektur mit konkreten Vorgaben für einzusetzende Technologien und Anwendungen) existieren. Dadurch sollen erfahrene Mitarbeiter der Verwaltung aus den Bereichen der Geschäftsprozessoptimierung sowie des Enterprise Architecture Managements eingebunden und zusammen mit den klassischen Funktionen des Projektmanagements des zugehörigen Anwendungsentwicklungs- bzw. Integrationsprojekts nach V-Modell XT zur Bewertung und Verbesserung des dargestellten Vorschlags beitragen. Dabei sollte insbesondere eine Auseinandersetzung mit dem Stellenwert der Dienste und der damit zu erstellenden *Dienstarchitektur* als Bindeglied zwischen übergreifender IT-Architektur, Anforderungen der Geschäftsprozesse und dem späteren technischen Entwurf (IT-Lösungsarchitektur) erfolgen.

Daneben wird es die Akzeptanz des V-Modells XT entscheidend steigern, wenn die **Werkzeugunterstützung** des V-Modells XT zu einem solchen, erweiterten Zweck ausgebaut wird.

Um einen nachhaltigen Erfolg im Zusammenwirken der Vorgehensmodelle zur Optimierung der Verwaltungsprozesse selbst sowie der Erhöhung der Effizienz und Effektivität der IT-Landschaft zu erreichen, müssen Maßnahmen zu einer ebenfalls miteinander verzahnten **Governance** entwickelt werden. Eine solche Governance betrifft das Vorge-

hen und die Regeln zur Gestaltung und laufenden Veränderung von Geschäftsprozessen einerseits und die **Steuerung** der **IT** mit Hilfe einer **IT-Governance** andererseits. Beide dienen am Ende demselben Zweck und müssen dazu zusammenwirken: Prozesse und IT der Verwaltung modern, zuverlässig und effizient zu gestalten. Als Fernziel sollte ein **konsistentes, vollständiges und aktuelles Rahmenwerk** für die **IT-Governance** der **Öffentlichen Verwaltung** in Deutschland konsequent weiter verfolgt werden.

Literaturverzeichnis

- [BM07] BMI; BMF :Konzept IT-Steuerung Bund. 2007. Siehe:
http://www.cio.bund.de/SharedDocs/Publikationen/DE/ Bundesbeauftragte-fuer-Informationstechnik/konzept_it_steuerung_bund_download.pdf
- [BM09] BMI: Rahmenarchitektur IT-Steuerung Bund, Version 1.0. 2009. Siehe:
http://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und-Standards/Rahmenarchitektur_itsteuerung_bund_grundlagen_download.pdf
- [BM10] BMI: Ableitung von Diensten aus Geschäftsprozessen; methodischer Leitfaden zur Rahmenarchitektur IT-Steuerung Bund., 2010. Siehe:
http://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und-Standards/leitfaden_rahmenarchitektur_download.pdf
- [BM11a] BMI; BVA: Konventionenhandbuch (Teil 1) für eine einheitliche Prozessmodellierung im Bundesministerium des Innern und seinen nachgeordneten Behörden. 2011. Siehe:
http://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und Standards/konventionenhandbuch_Teil1_download.pdf
- [BM11B] BMI: Rahmenwerk für ein einheitliches Geschäftsprozessmanagement im Bundesministerium des Innern und seinen nachgeordneten Behörden, Version 1.0. 2011. Internes Dokument
- [BU10] Buckl, S. et al.: EAM Pattern Catalog. Technical Report TB 0801, TU München, 2008-2010. Siehe: <http://www.matthes.in.tum.de/pages/146i8ldzg690d/EAM-Pattern-Catalog>
- [CI10] CIO Bund: V-Modell XT® Bund 1.0. 2010. Siehe:
http://www.cio.bund.de/DE/Architekturen-und-Standards/V-Modell-XT-Bund/vmodellxt_bund_node.html
- [CI11] CIO Bund: SAGA 5, Rat der IT-Beauftragten der Bundesregierung, 2011. Siehe:
http://www.cio.bund.de/DE/Architekturen-und-Standards/SAGA/saga_node.html
- [CI12] CIO Bund: V-Modell XT® 1.4. 2012. Siehe: http://www.cio.bund.de/DE/Architekturen-und-Standards/V-Modell-X T/vmodell_xt_node.html
- [OM12] OMB: The common approach to federal architecture. 2012. Siehe:
http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/common_approach_to_federal_ea.pdf
- [Op11] The Open Group: An Introduction to TOGAF® 9.1. 2011. Siehe:
<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>
- [Za04] Zachman, J. A: Enterprise Architecture, A Framework. Zachman Institute, 2004.

Verantwortung übernehmen: Das Project Management Office als Process Owner für das unternehmensweite Projektmanagement

Stefan Hilmer

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

Abstract: Die Diskrepanz zwischen unternehmensweiten Vorgehensmodellen und den tatsächlich gelebten Prozessen des Projektmanagements lässt sich durch die konsequente Zuweisung von Verantwortlichkeiten verringern. Als Lösung schlägt dieser Beitrag vor, die Verantwortung des aus dem Business Process Management bekannten Process Owner auf das Project Management Office zu übertragen und dabei konkrete Aufgaben des Project Management Office festzulegen. Auf diesem Weg wird die Verantwortung für das unternehmensweite Projektmanagement sichergestellt und damit die einheitliche Nutzung vorhandener Vorgehensmodelle verbessert.

1 Einleitung

Anspruch und Wirklichkeit bei der Verwendung von Vorgehensmodellen liegen im unternehmensweiten Projektmanagement in der Praxis oft weit auseinander. Dort wo diese Differenz besonders groß ist, mangelt es erfahrungsgemäß häufig an der Verankerung von Verantwortlichkeiten. Diese Verantwortlichkeiten sind sowohl bei der Erstellung bzw. der Auswahl der Modelle als auch bei deren Verwendung zu regeln. Hier sind nicht zwingend neuen Methoden oder Verfahren zu entwickeln, vielmehr können vorhandene Erkenntnisse und Ansätze aus dem Business Process Management (BPM) übernommen werden. Im BPM wird dazu die Rolle des Prozessverantwortlichen, des Process Owner, definiert. Die Übertragung dieser Definition auf das Project Management Office (PMO) als zentrale Instanz des unternehmensweiten Projektmanagements bietet einen nachhaltigen Ansatz für die konsequente Anwendung von Vorgehensmodellen im Projektmanagement.

Dieser Ansatz wird im Folgenden beschrieben. Dazu wird in Abschnitt 2 zunächst die Rolle des Process Owner im Rahmen des BPM-Lebenszyklus betrachtet. Anschließend wird in Abschnitt 3 ein allgemeingültiges Bild eines PMO aufgezeigt. Dieses Bild stützt sich auf Darstellungen in der gängigen Literatur, in der vorzugsweise die Aufgaben eines PMO aufgezeigt werden. Die Übernahme der zugehörigen Verantwortlichkeiten wird auch in der Praxis oft vernachlässigt. Abhilfe bietet die in Abschnitt 4. beschriebene Zuweisung von Verantwortung nach dem Muster des Process Owner.

2 Der Process Owner

Die Rolle des Process Owner im Sinne des BPM wird in [Eu09] ausführlich beschrieben. Demnach hat sich in der Praxis eine ganze Fülle unterschiedlichster Rollenbezeichnungen im BPM herausgebildet, Process Manager, Process Analyst, Process Architect usw. Bei dieser Vielfalt hebt sich jedoch die Rolle des Process Owner heraus. Gemäß [Eu09] spielt sie in fast allen Unternehmen, die das Prozessmanagement erfolgreich eingeführt haben, eine zentrale Rolle. Diese gilt als wesentlicher Erfolgsfaktor für das Prozessmanagement. Vor diesem Hintergrund soll die Rolle des Process Owner hier in den verschiedenen Phasen des BPM-Lebenszyklus betrachtet werden.

2.1 Der BPM-Lebenszyklus

Der Lebenszyklus eines BPM kann durch verschiedene Aktivitäten beschrieben werden. An dieser Stelle sollen fünf typische BPM-Aktivitäten betrachtet werden, die in [Eu09] exemplarisch zusammengestellt sind. Der so entstehende BPM-Lebenszyklus wird in Abbildung 1 dargestellt. Die ersten vier Aktivitäten unterscheiden sich dabei von der fünften, da sie einmalig durchlaufen werden. Die anschließende kontinuierliche Prozesssteuerung und -optimierung hingegen wiederholt sich fortwährend.

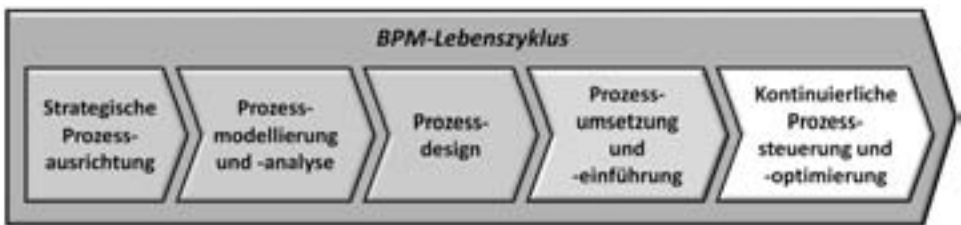


Abbildung 1: Aktivitäten im BPM-Lebenszyklus

In diesem Modell beginnt der Lebenszyklus mit der strategischen Prozessausrichtung. Sie umfasst die Entwicklung einer am Prozessgedanken ausgerichteten Strategie und der Vorgehensplanung. Letztlich geht es dabei um Strategien und Ziele mit dem speziellen Augenmerk darauf, mit einem ganzheitlichen BPM-Ansatz überzeugende Leistungen anzubieten.

Im Rahmen der Prozessmodellierung und -analyse werden Prozesse erhoben und Faktoren ermittelt, die für den Prozess förderlich oder hinderlich sind. Dabei werden die Ist-Zustände der Prozesse dokumentiert.

Das anschließende Prozessdesign beschäftigt sich mit der Gestaltung des zukünftigen Soll-Prozesses. Dazu werden die Abfolge der Prozessschritte und deren Inhalt sowie Zeit und Ort der Ausführung festgelegt. Außerdem sind angemessene Kontrollmechanismen einzurichten und Kennzahlen für die zukünftige Leistungsmessung zu definieren.

Mit der Prozessumsetzung und -einführung werden die geplanten Soll-Prozesse realisiert und in das Unternehmen hineingetragen, so dass sie dort anwendbar sind. Dieser Schritt

erfordert ein umfassendes Change Management, bei dem alle Stakeholder von der Qualität der Prozesse zu überzeugen sind.

Eingeführte Prozesse sind nach diesem Modell einer kontinuierlichen Prozesssteuerung und -optimierung zu unterziehen. Kontinuierliches Messen und Verfolgen von Prozessen liefert die Informationen, um Prozesse anzupassen und BPM-Ziele besser erreichen zu können.

2.2 Die Rolle des Process Owner im BPM-Lebenszyklus

Zu dem in Abschnitt 2.1 dargestellten BPM-Lebenszyklus wird in [Eu09] die zentrale Rolle des Process Owner definiert. Dieser trägt die dauerhafte Verantwortung für den kompletten End-to-end-Prozess. Diese Verantwortung umfasst die Gestaltung, Umsetzung und Auszuführung des Prozesses und verteilt sich somit über den gesamten Lebenszyklus.

Process Owner sollten von Anfang an im BPM-Lebenszyklus benannt sein, um über den gesamten Zyklus hinweg ihre Verantwortung wahrnehmen zu können. Daher ist ihre Rolle schon für die strategische Prozessausrichtung in der Organisationsstruktur zu verankern.

In der Entwicklung von Prozessmodellen sind aufgrund der verschiedenen Zielsetzungen in der Regel sehr viele Personen in verschiedenen Rollen involviert, um die Vollständigkeit der Modelle zu gewährleisten. Prozessanalysen erfordern daher das Zusammenspiel vieler Personen. Die Verantwortung für dieses Zusammenspiel sowie für Vollständigkeit und Korrektheit der Ergebnisse werden beim Process Owner gebündelt.

Process Owner müssen viele Fragen zum Prozessdesign mit anderen Managern oder Prozessbeteiligten abstimmen. Dennoch sind sie verantwortlich für das Prozessdesign in seiner Gesamtheit, auch wenn viele Prozessdetails von anderen entschieden werden.

Im Rahmen der Prozessumsetzung und -einführung können verschiedenste Methoden zum Einsatz kommen, wie z. B. Total Quality Management (TQM), Prozesskostenrechnung (activity based costing) oder Performance Improvement (PI). Für jede angewandte Methode wird eine Expertise benötigt, die der Process Owner nicht selbst mitbringen muss. Dennoch obliegt ihm in dieser Phase die Verantwortung für die zielführende Einführung der Prozesse.

Die Verantwortung des Process Owner im Rahmen der kontinuierliche Prozesssteuerung und -optimierung konzentriert sich im BPM stark auf die Prozessleistung. Der Prozessverantwortliche stellt dafür die Prozessleistungsmessung gemäß Prozessdesign sicher. Dabei benötigt er Entscheidungshilfen, die er über Managementmethoden, wie beispielsweise Process Cockpits oder Balanced Scorecards (BSC), gewinnt. Geeignete Entscheidungshilfen können dem Process Owner dazu dienen, die Prozessleistung aufrecht zu erhalten und genau hier liegt seine Verantwortung.

Insgesamt übernimmt der Process Owner zudem eine allgemeine Interessenvertretung, um sicherzustellen, dass benötigte Ressourcen und notwendige Maßnahmen bereitgestellt werden und dass das Top-Management nachhaltig die von ihm verantworteten Prozesse unterstützt.

3 Das Project Management Office

Projektorientierte Unternehmen verankern das Projektmanagement nachhaltig in ihrer Stammorganisation. Aufbauorganisatorisch geschieht dies häufig in Form einer festen Stelle oder Abteilung (vergl. [Ge09]). Für diese Rolle hat sich der Begriff Project Management Office (PMO) weitestgehend durchgesetzt. Ein PMO kann für ein Unternehmen oder eine Teilorganisation (Bereich bzw. Abteilung) eingerichtet werden. Diese Einrichtung kann als organisatorische Maßnahme die Qualität und Performance der gesamten Projektlandschaft steigern und die Projektsteuerung vereinheitlichen. Wichtig ist die Dauerhaftigkeit der Einrichtung PMO. An dieser Stelle treten häufig Verwechslungen mit einer anderen Organisationseinheit, dem Project Office auf. Ein Project Office wird jedoch nur für die Dauer eines Projektes eingerichtet (vergl. [Ge09]).

3.1 Die Aufgaben des PMO

Ein PMO dient als projektübergreifende Unterstützungsfunktion zur Einführung und Optimierung von Projektmanagementsystemen sowie der operativen Unterstützung von Projekten und Projektbeteiligten. So definiert es die DIN 69901-5. In dieser Funktion werden dem PMO verschiedene Aufgaben zugewiesen (vergl. [Ge09], [SOP08] und [Sc10]), die sich drei grundsätzlichen Aufgabenfelder zuordnen lassen:

- Standardisierung von Projektmanagementsystemen
- Unterstützung des operativen Projektmanagements
- Organisation des Multi-Projektmanagements

Die Standardisierung von Projektmanagementsystemen umfasst typischerweise die Erarbeitung, Pflege und Überwachung von Standards und einheitlichen Vorgehensmodellen vor dem Hintergrund der vorhandenen Unternehmensstrategie.

Im operativen Projektmanagement steht das PMO allen Projekten innerhalb des Unternehmens unterstützend und beratend zur Seite. Dies gilt insbesondere in Bezug auf unternehmenseinheitliche Vorgehensmodelle und anderer Standards. Die Aus- und Weiterbildung von Projektmanagern können dabei eine wesentliche Unterstützungsmaßnahme bilden. Aber auch die Bereitstellung qualifizierter Projektmanager, z.B. aus einem zentralen Pool, ist möglich.

Im Bereich Multi-Projektmanagement verantwortet das PMO geeignete Prozesse zur Planung, Umsetzung und Kontrolle aller Projekte des Unternehmens bzw. der vom PMO

betreuten Organisationseinheit. Das Spektrum reicht von der projektübergreifenden Kapazitätsplanung bis hin zum kompletten Project Portfolio Management. In dieser Funktion unterstützt das PMO das Management durch Aufbereitung von aggregierten Informationen für die Entscheidungsfindung.

Zur Erfüllung dieser Aufgaben im Betrieb des PMO ist dieses zunächst entsprechend im Unternehmen aufzusetzen. Dieses Setup und der anschließende Betrieb bilden den PMO-Lebenszyklus.

3. 2 PMO-Lebenszyklus

Der Lebenszyklus eines PMO beginnt mit seinem Setup. Für dieses hat sich in der Praxis ein Vorgehen bewährt, das aus fünf Schritten besteht, die zusammen mit dem nachfolgenden PMO-Betrieb in Abbildung 2 dargestellt werden.



Abbildung 2: Aktivitäten im PMO-Lebenszyklus

Die erste Aktivität des PMO-Lebenszyklus beinhaltet, ausgehend von einer Analyse des individuellen Projektportfolios, die Zieldefinition für das zukünftige PMO. Mit diesem, in der Praxis oft vernachlässigten Schritt, werden klare Ziele aufgestellt und dokumentiert (vergl. [Sc13]).

Das Leistungsangebot festzulegen umfasst im Wesentlichen die Ableitung konkreter Maßnahmen aus den gefundenen Zielen und deren Festlegung im Rahmen eines Umsetzungsplans (vergl. [Sc13]). Das Spektrum möglicher Maßnahmen erstreckt sich auf die Bereiche Standardisierung von Projektmanagementsystemen, Unterstützung des operativen Projektmanagements sowie Institutionalisierung des Multi-Projektmanagements. Nachdem im vorherigen Schritt definiert wurde, was das zukünftige PMO erreichen soll, wird hier festgelegt, welche Mittel dafür eingesetzt werden. In der Regel zählen geeignete Vorgehensmodelle zu diesen Mitteln.

Während des Aufbaus des Leistungsangebotes werden geeignete Vorgehensmodelle und standardisierte Projektmanagement-Methoden, beispielsweise zur Projektplanung und -steuerung, entwickelt (vergl. z. B. [Ge09] und [SOP08]). Zu diesen werden zugehörige Unterstützungsmaßnahmen, z.B. Schulungen, für das operative Projektmanagement vorbereitet. Zudem werden Methoden zum Multi-Projektmanagement definiert und für die Etablierung in der Organisation vorbereitet.

Parallel zum Aufbau des Leistungsangebots ist eine PMO-Organisation zu schaffen, die geeignet ist, die definierten Leistungen zu erbringen. Diese steht in enger Wechselwir-

kung mit dem Unternehmen und ist daher an ihm auszurichten. Bei Planung und Aufbau von Leistungsangebot und PMO-Organisation ist insbesondere das individuelle Umfeld des PMO zu berücksichtigen, um die Akzeptanz im Unternehmen und damit den Erfolg des PMO sicherzustellen (vergl. [KS10]).

Mit dem nächsten Schritt wird der Übergang des PMO in den operativen Betrieb vollzogen, d.h. das gesamte geschaffene Leistungsangebot und die erbringende Organisation werden so im Unternehmen etabliert, dass die angestrebten Ziele passgenau erreicht werden. Beides, Leistungsangebot und PMO-Organisation, zu etablieren setzt voraus, dass in den vorherigen Phasen eine starke Akzeptanz und ein hohes Maß an Qualität im Leistungsangebot erreicht wurden. Erfahrungsgemäß kann nur so ein nachhaltiger PMO-Betrieb sichergestellt werden.

In dem so vorbereiteten Betrieb werden vom PMO die drei benannten Aufgaben, die Standardisierung von Projektmanagementsystemen, die Unterstützung des operativen Projektmanagements sowie die Organisation des Multi-Projektmanagements, wahrgenommen. Dabei gilt es, das gesamte Leistungsangebot kontinuierlich zu verbessern.

Den in Abschnitt 3.1 dargestellten Aufgaben des PMO werden in der Regel keine konkreten Verantwortungen zugewiesen, nicht in der DIN, nicht in der Literatur und erfahrungsgemäß auch nicht in der Praxis. Diese zu übernehmende Verantwortung soll im Folgenden über den PMO-Lebenszyklus hinweg betrachtet werden.

4 Das PMO als Process Owner

Die in diesem Beitrag betrachteten Vorgehensmodelle bilden die Prozesse, für die das PMO im Sinne eines Process Owner (s. Abschnitt 2) die Prozessverantwortung übernehmen soll. Sie sind einer der zentralen Aufgaben eines PMO (s. Abschnitt 3) zuzuordnen: der Standardisierung von Projektmanagementsystemen. Bezüglich der Prozessverantwortung ist die Art des Vorgehensmodells unerheblich. Möglich sind beispielsweise die in der Software-Entwicklung weit verbreiteten standardisierten Modelle Rational Unified Process (RUP), V-Modell oder Scrum, aber auch proprietäre Modelle, klassisch, agil oder hybrid, sind möglich (vergl. [Hi12]). Allein die zu übernehmende Verantwortung des PMO soll an dieser Stelle im Vordergrund stehen. Diese Verantwortung umfasst die Gestaltung, Umsetzung und Ausführung des Vorgehensmodells und verteilt sich somit über den gesamten Lebenszyklus des PMO.

Im Sinne der BPM-Definitionen zum BPM (s. Abschnitt 2) übernimmt das PMO als Prozessverantwortlicher eine zentrale Rolle in Bezug auf die Projektmanagement-Vorgehensmodelle des Unternehmens. Damit trägt es in jedem Schritt des PMO-Lebenszyklus die Verantwortung. Diese Schritte sollen in den folgenden Abschnitten einzeln betrachtet werden.

4.1 Ziele definieren

So wie der Process Owner im BPM die Verantwortung für einen Prozess bereits in der strategischen Prozessausrichtung übernimmt, sollte auch das PMO bereits bei der Zieldefinition seine Verantwortung übernehmen. An dieser Stelle kann sicher noch nicht von einer dedizierten Verantwortung für einen bestimmten, in einem Vorgehensmodell verankerten Prozess gesprochen werden. Allerdings sollte sich das PMO schon jetzt generell zu seiner Verantwortung für die zu entwickelnden Modelle bekennen und dieses Bekenntnis auch in verwendeten strategischen Werkzeugen, wie beispielsweise der PMO-Vision und insbesondere dem PMO-Leitbild, verankern (vergl. [Sc13]).

Eine PMO-Vision beschreibt die Ziele des angestrebten PMO. Dazu zählt auch eine Darstellung, wo es in drei bis fünf Jahren stehen will. Damit legt das PMO seine langfristige Ausrichtung fest und macht diese gegenüber den Stakeholdern sichtbar. Genau in einer solchen nach außen gerichteten Selbstdefinition darf die Benennung der eigenen Verantwortung nicht fehlen. Das gleiche gilt auch für das PMO-Leitbild, oder eine andere, nach innen und außen gerichtete Darstellung der vorgesehenen Zusammenarbeit und des eigenen Selbstverständnis, zu dem auch die eigene Verantwortung zählt.

4.2 Leistungsangebot festlegen

Vorgehensmodelle sind in praktisch jedem PMO-Leistungsangebot enthalten. Entweder wird ein standardisiertes Modell ausgewählt oder ein proprietäres Modell entwickelt. In jedem Fall wird ein Modell definiert, um die Standardisierung von Projektmanagementsystemen voranzutreiben, die Unterstützung des operativen Projektmanagements sicherzustellen oder eine Basis für die Institutionalisierung des Multi-Projektmanagements zu schaffen. Es ist also davon auszugehen, dass ein Vorgehensmodell zu den Mitteln zählt, die zur Erreichung der im ersten Schritt des PMO-Lebenszyklus aufgestellten Ziele ausgewählt werden.

Das Leistungsangebot wird beispielsweise in einem PMO-Umsetzungsplan verankert (vergl. [Sc13]). Ein PMO-Umsetzungsplan zeigt, welche Maßnahmen das PMO ergreifen will und welche Ziele damit erreicht werden bzw. wie die Mission und das Leitbild umgesetzt werden. Zudem, wird festgelegt, wer an der Umsetzung beteiligt ist. Mit dieser Planung übernimmt das PMO seine Verantwortung, überträgt aber gleichzeitig auch Verantwortung auf andere Stakeholder.

4.3 Leistungsangebot aufbauen

Wie im vorangegangenen Abschnitt dargestellt, ist ein Vorgehensmodell im Leistungsangebot der meisten PMO enthalten. Unabhängig davon, ob ein standardisiertes Modell ausgewählt oder ein eigenes Modell entwickelt werden soll, das entsprechende Vorgehen zur Auswahl oder Entwicklung ist zu planen und umzusetzen. Sowohl in die Auswahl als auch in die Entwicklung von Vorgehensmodellen sind aufgrund der verschiedenen Zielsetzungen viele Personen involviert. Dies gilt von der Analyse bis zum Design. Nur so gelingt es, ein passgenaues Vorgehen in dem Zielunternehmen zu etablieren.

So wie ein Process Owner viele Fragen zum Prozessdesign mit anderen Managern und Prozessbeteiligten abstimmt, muss sich auch das PMO bei allen Maßnahmen zum Vorgehensmodell mit vielen Stakeholdern abstimmen. Dennoch ist das PMO verantwortlich für das Modelldesign in seiner Gesamtheit. Das PMO muss in diesem Zusammenhang zudem die Verantwortung für das Zusammenspiel der Beteiligten sowie für Vollständigkeit und Korrektheit der Ergebnismodelle übernehmen. Insgesamt plant das PMO mit seinem Leistungsangebot auch seine Verantwortung. Auch dieser Teil der Planung ist dabei zu dokumentieren und gegenüber den Stakeholdern transparent zu machen.

4.4 PMO-Organisation aufbauen

Parallel zum Leistungsangebot ist eine PMO-Organisation aufzubauen, die geeignet ist, die angebotenen Leistungen zu erbringen. Dies gilt auch, oder insbesondere, für die PMO-Aktivitäten rund um die vorgesehenen Vorgehensmodelle. Sie stehen in einer sehr engen Wechselwirkung mit dem Unternehmen bzw. den darin durchgeführten Projekten. Hier liegt die besondere Verantwortung des PMO in der Sicherstellung der Akzeptanz der angestrebten Modelle bei allen Stakeholdern.

Akzeptanz ist nur schwer messbar. In [KS10] wird jedoch mit dem „Blitzcheck von Stakeholdern“ ein geeigneter Ansatz dafür vorgestellt. Die Akzeptanz von Vorgehensmodellen ist bei allen Stakeholdern stark durch deren Empfindungen gesteuert. Damit ist auch die Anzahl der Faktoren sehr groß, die diese Akzeptanz beeinflussen. Der „Blitzcheck von Stakeholdern“ fasst die wichtigsten Faktoren zusammen und gewichtet diese mit einem einfachen Punktemodell. Dieser Ansatz erlaubt zumindest eine grobe Einordnung der Stakeholder in Gruppen, und zwar danach, wie wahrscheinlich sie dem Vorhaben positiv oder negativ gegenüberstehen werden.

4.5 PMO etablieren

Vorgehensmodelle im Unternehmen zu etablieren, d. h. sie in den operativen Betrieb zu übernehmen, gelingt nur wenn ein passgenaues Modell entwickelt wurde, das von allen Stakeholdern im Unternehmen akzeptiert wird. Beides ist vor allem über die beiden vorherigen Schritte sicherzustellen.

Allerdings verantwortet das PMO auch die Aufrechterhaltung der Akzeptanz, beispielsweise durch Aus- und Weiterbildungsmaßnahmen oder durch gezielte Information aller Beteiligten. Zudem muss schon in diesem Schritt eine kontinuierliche Prozesssteuerung und -optimierung vorbereitet werden, um die angestrebten Ziele des PMO dauerhaft zu erreichen.

4.6 PMO betreiben und weiterentwickeln

Die Verantwortung des PMO im Betrieb liegt in der Wahrnehmung seiner projektübergreifenden Unterstützungsfunktion. Dabei sorgt es auch dafür, dass die benötigten Ressourcen und notwendigen Maßnahmen bereitgestellt werden und dass das Management

nachhaltig die von ihm verantworteten Vorgehensmodelle unterstützt. In Abschnitt 3.1 wurden die grundsätzlichen Aufgaben eines PMO dargestellt, für die im Folgenden hinsichtlich der Verantwortungen des PMO im laufenden Betrieb genauer betrachtet werden.

4.6.1 Standardisierung von Projektmanagementsystemen

Ein wesentlicher Bestandteil der Aktivitäten eines PMO im laufenden Betrieb bildet die kontinuierliche Weiterentwicklung von Projektmanagementsystemen. Zu diesen zählen selbstverständlich die eingeführten Vorgehensmodelle. Ihre Pflege vor dem Hintergrund der vorhandenen Unternehmensstrategie zählt zu den zentralen Verantwortungen des PMO. Sie erfolgt typischerweise im Rahmen eines kontinuierlichen Verbesserungsprozesses, der sich mehr an der Zufriedenheit der Stakeholder orientiert als an einer gemessenen Prozessleistung. Letzter lässt sich in einem Vorgehensmodell für das Projektmanagement nur schwer ermitteln, da Projekte per Definition einen individuellen Charakter aufweisen.

Die Überwachung der Modelle bringt eine zusätzliche, zentrale Verantwortung mit sich. Hier kommt dem PMO die Rolle eines Projektcontrollers zu (vergl. [Ge09] und [SOP08]). Dabei beschreiten die Mitarbeiter des PMO oft einen schmalen Grad zwischen der Verantwortung für die Einhaltung der Modelle und der Sicherung der Akzeptanz bei den überwachten Stakeholdern. Dies bildet eine große Herausforderung, die vorrangig mit Hilfe individueller Maßnahmen in dem jeweiligen PMO-Umfeld gelöst werden kann. Eine grundsätzliche Maßnahme bildete hierbei beispielsweise die einheitliche Ablage aller verbindlich im Vorgehensmodell definierten sowie aller weiteren zusätzlichen Projektmanagement-Dokumente in einer standardisierten Struktur. Dies kann eine Verzeichnisstruktur, ein Dokumentenmanagement-System oder eine andere strukturierte Dokumentenablage sein. Wichtig ist, dass alle Dokumente einheitlich und für alle Beteiligten transparent und dauerhaft verfügbar sind. Für die Inhalte, also die erstellten Projektmanagement-Dokumente, sind die Projekte respektive die Projektmanager verantwortlich. Die Ablage, deren Struktur und vor allem die Überwachung der Einhaltung verantwortet das PMO.

4.6.2 Unterstützung des operativen Projektmanagements

Eine der Kernaufgaben des PMO im laufenden Betrieb bildet die Unterstützung und Beratung aller Projekte innerhalb des Unternehmens. Dies gilt selbstverständlich auch in allen Fragen zu den unternehmenseinheitlichen Vorgehensmodellen. Die Aus- und Weiterbildung von Projektmanagern kann genauso dazugehören wie die Bereitstellung qualifizierter Projektmanager oder Berater mit hinreichender Erfahrung in den vorhandenen Modellen. Mit diesen Maßnahmen kommt das PMO auch der bereits beschriebenen Verantwortung für die Sicherstellung der Akzeptanz nach.

4.6.3 Organisation des Multi-Projektmanagements

Im Bereich Multi-Projektmanagement verantwortet das PMO geeignete Prozesse zur Planung, Umsetzung und Kontrolle aller Projekte. Diese Kontrolle umfasst neben der bereits beschriebenen Einhaltung der unternehmenseigenen Standards vor allem die Kontrolle des Projektfortschritts, der Zielerreichung und der Einhaltung von Termin- und Budgetgrenzen. Vorgehensmodelle bilden dabei eine wesentliche Grundlage. Im Falle eines etablierten Multi-Projektmanagements, z. B. in Form eines dedizierten Project Portfolio Managements, liegt die Verantwortung des PMO zusätzlich in der Ausrichtung der Vorgehensmodelle auf die Aufbereitung von aggregierten Informationen für die Entscheidungsfindung des Managements. Die Verantwortung für das Modell selbst bleibt dabei unverändert.

5 Fazit

Nach dem Vorbild des Process Owner und seiner Verantwortung im Rahmen des BPM-Lebenszyklus können auch dem PMO Verantwortungen für das unternehmensweite Projektmanagement und die zugehörigen Vorgehensmodelle übertragen werden. Dies geschieht in der Regel in Zusammenhang mit konkreten, dem PMO zugewiesenen oder von ihm selbst geplanten Aufgaben.

Bei einer geeigneten Definition und einer vollständigen Wahrnehmung aller definierten Verantwortungen, wird das PMO in die Lage versetzt, die in der Praxis übliche Lücke zwischen Anspruch und Wirklichkeit eines unternehmensweiten Vorgehens für das Projektmanagement zu verkleinern.

Um den damit verbundenen Verantwortungen gerecht werden zu können muss ein PMO angemessen – insbesondere personell – ausgestattet sein. Der verantwortungsbezogene Aufwand und die damit verbundenen Kosten sind nicht unerheblich und müssen erbracht werden. Zudem muss die gesamte Organisation die Vergabe der Verantwortlichkeiten kennen und akzeptieren. Nur wenn die Projekte die Verantwortung des PMO anerkennen, wird es zu einer verbesserten Einhaltung von Vorgehensmodellen kommen. Dies bringt jedoch normalerweise starke Veränderungen in der Projektmanagement-Kultur mit sich, die wiederum einen erheblichen Aufwand erfordern. Aber, nur so ist es möglich, dass die beschriebenen Verantwortungen gelebt werden können.

Literaturverzeichnis

- [Eu09] European Association of Business Process Management - EABPM: Business Process Management - BPM Common Body of Knowledge - BPM CBOK®. Verlag Dr. Götz Schmidt, Gießen, 2009.
- [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, 2012; S. 129-138.
- [KS10] Kuhlmei, A.; Schneyder, W. von: Leitstand für das Projektgeschäft / Project Management Offices implementieren und Akzeptanz sichern. Projekt Magazin 21/2010.
- [Sc10] Schneyder, W. von: Leitstand für das Projektportfolio / Das Project Management Office - eine Einführung. Projekt Magazin, 17/2010.
- [Sc13] Schuchardt, J.: Visionen und Leitbilder – wirksame Werkzeuge für den PMO-Erfolg. Projekt Magazin, 01/2013.
- [SOP08] Schelle, H.; Ottmann, R.; Pfeiffer, A.: ProjektManager. GPM - Deutsche Gesellschaft für Projektmanagement, 2008.

Wer macht eigentlich Requirements Engineering & Management?

Andrea Herrmann¹, Rüdiger Weißbach²

¹Freie Software Engineering Trainerin
Daimlerstr. 121, 70372 Stuttgart
herrmann@herrmann-ehrlich.de

²Hochschule für Angewandte Wissenschaften (HAW) Hamburg
Berliner Tor 5, 20099 Hamburg
ruediger.weissbach@haw-hamburg.de

Abstract: Rollen als Teile eines Vorgehensmodells dienen dazu, Aufgaben und Verantwortlichkeiten eindeutig und optimal Personen zuzuweisen. Da das Requirements Engineering und Management (RE&M) wichtig ist, gibt es bei vielen - jedoch nicht allen - Vorgehensmodellen eine oder mehrere Rollen, die das RE&M durchführen. Tatsächlich ergeben sich in der Praxis einige Schwierigkeiten bei der Arbeit im RE&M aufgrund von suboptimalen Rollendefinitionen oder dem Fehlen von solchen. Dieser Artikel diskutiert solche Schwierigkeiten anhand von zwei Studien.

1 Einleitung

Dass Requirements Engineering & Management (RE&M) einen wesentlichen Einfluss auf den Projekt(miss)erfolg hat, ist beispielsweise durch die Untersuchungen der Standish Group [St01] aufgezeigt worden. Bei solch einer Kerntätigkeit müsste anzunehmen sein, dass sie in den gängigen Vorgehensmodellen entsprechend Berücksichtigung findet. In den Arbeiten zur Untersuchung an der „Nahtstelle“ zwischen Requirements Engineering und Projektmanagement des GI-Arbeitskreises „Requirements Engineering und Projektmanagement“ (www.repm.de) zeigte sich, dass dies nicht so stringent ist, wie erhofft. So identifizierten wir bei der Analyse des V-Modell XT allein 12 Rollen, die zum RE&M beitragen [FW07], sowohl auf Seiten des Auftragnehmers als auch des Auftraggebers. Diese beschäftigen sich aber alle gleichzeitig auch mehr oder weniger intensiv mit Themen des Projektmanagements und der Implementierung. Das PMBOK [PM13] kennt kein Requirements Engineering, sondern definiert die Verantwortlichkeit des Projektleiters als: “Managing a project includes: identifying requirements, ...”. Das Gebiet des Scope Managements im PMBOK enthält Tätigkeiten des RE&M, aber eben auch im Projektmanagement und anderen Bereichen. In der agilen Entwicklung dagegen, sind quasi alle Mitglieder des Projektteams am RE&M beteiligt.

Dieses Prinzip - Verteilung des RE&M auf viele Rollen und Personen sowie gleichzeitige Arbeit der für die Anforderungen Verantwortlichen auch in anderen Fachgebieten - fanden wir auch in unseren empirischen Studien sowie in den Studien anderer Forscher.

Im weiteren Verlauf der Arbeit werden wir im Kapitel 2 den Begriff des RE&M klären. Im Kapitel 3 wird zunächst der Stand der Forschung dargestellt, anschließend werden zwei empirische Studien vorgestellt, eine Studie, in der Stellenangebote hinsichtlich der dort repräsentierten Rollenmodelle untersucht wurden, sowie eine Studie, die die Diversität in der Partizipation der Fachabteilungen untersuchte. Zusammenfassung und Ausblick schließen die Arbeit ab.

2 Was ist RE&M?

Zur Bearbeitung der Frage, welche Akteure den RE&M-Prozess durchführen, müssen wir den Begriff des RE&M klären. Dazu gehen wir von der Definition der IEEE [Iee90] aus.

Demnach ist eine Anforderung ...

1. eine Bedingung oder Fähigkeit, die von einem Benutzer benötigt wird, um ein Problem zu lösen oder ein Ziel zu erreichen,
2. eine Bedingung oder Fähigkeit, die ein System oder eine Komponente des Systems erfüllen oder besitzen muss, um einen Vertrag, einen Standard, eine Spezifikation, oder ein anderes formal auferlegtes Dokument zu erfüllen
3. eine dokumentierte Darstellung einer Bedingung oder Fähigkeit wie in 1. und 2. [Iee90].

Requirements Engineering umfasst sämtliche Tätigkeiten, die erforderlich sind, um (Produkt- und Projekt-) Anforderungen zu erheben, zu analysieren, zu verstehen und zu dokumentieren. Schließlich sind auch Tätigkeiten zur Auflösung von Unstimmigkeiten, zur Verifikation und zur Validierung von Anforderungen (z.B. Anforderungsreviews) Teil des Requirements Engineering [FHW06].

Requirements Management „umfasst Prozesse, die notwendig sind, um einerseits Anforderungen und die dazugehörigen Informationen für verschiedene Rollen aufzubereiten und andererseits diese konsistent zu ändern“. ([Ru07], S. 350)

Dies sind konkret alle Tätigkeiten, um

1. die verwalteten Anforderungen allen anderen Disziplinen der Projektdurchführung und allen Stakeholdern zur Verfügung zu stellen und an diese zu kommunizieren, gegebenenfalls auch zielgruppenspezifisch aufbereitet,
2. Änderungs- und Konfigurationsverwaltung für Anforderungen durchzuführen, z.B. durch Versionsverwaltung und Vorabschätzung der Einflüsse von Anforderungsänderungen,
3. die Anforderungsentwicklung anhand von Statusattributen oder offenen Punktelisten zu verfolgen,
4. die Beziehungen zwischen Anforderungen u.a. zum Zwecke der Rückverfolgbarkeit zu pflegen.

3 Wer macht tatsächlich RE&M?

Es gibt nur relativ wenige Untersuchungen darüber, wer in der Praxis RE&M macht. Zunächst fassen wir im Folgenden einige Studien anderer Autoren zusammen und beschreiben dann unsere eigenen.

3.1 Stand der Forschung

Untersuchungen über RE&M in der Praxis zeigen eine unklare Definition der Rolle des Requirements-Engineers. Die Verteilung von Verantwortlichkeiten und Aufgaben variiert je nach Struktur der Organisation, den Projektumständen oder individuellen Fähigkeiten (vgl. [AW05]). Hinzu kommt, dass am RE&M mehrere Rollen beteiligt sind, wie Management, Architekt, Berater, Projektmanager, System-Designer, Analyst und technischer Spezialist, Marketing, Produktentwicklung, Support, Produktnutzer, Kunden und Fachabteilungen [ZWO01], [NL03]. Gerade in klein- und mittelständischen Unternehmen gibt es oft keine spezialisierten Rollen, sondern nur „Entwickler“ (in [NSK00] war dies bei 6 von 12 finnischen KMUs der Fall). Allein daraus, dass es üblicherweise keine spezielle RE&M-Rolle gibt, folgt, dass die Person, die das RE&M durchführt, noch andere Aufgaben haben muss. Doch welche und wie viele? Hierüber gibt es keine Untersuchungen, aber eine Studie stellte bereits fest, dass die Aufgabenvielfalt diejenigen, die im RE arbeiten, sehr fordert [AW05]. Klendauer et al. [Kr12] beobachteten in ihrer internationalen Fallstudie in Europa und Nordamerika, dass eine formale RE&M-Rolle die Entwickler entlastet, indem der Requirements Engineer die kommunikativen Aufgaben übernehmen kann, während sich die Entwickler auf das Programmieren konzentrieren können. In der Praxis wird RE&M oft ohne spezielle RE&M-Kenntnisse durchgeführt, z. B. von ungeschulten Berufsanfängern [AP05]. Als Ursache dafür wird unter anderem mangelndes Wissen über die Existenz von RE&M-Methoden und -Werkzeugen genannt, insbesondere bei KMUs [NSK00]. Darum werden auch keine RE-Methoden verwendet [Kr12].

3.2 Analyse von Stellenanzeigen

Um die Arbeit von RE&M in der Praxis zu untersuchen, wertete Herrmann zu zwei Zeitpunkten (2009 und 2012) 141 bzw. 67 Stellenanzeigen aus. Stellenanzeigen stellen ein offizielles Modell dessen dar, wie Firmen das RE&M in ihr Organigramm integrieren und welche Kompetenzen sie für diese Aufgabe für nötig halten. Im Idealfall spiegeln Stellenanzeigen auch frühere Erfahrungen wider, d.h. sie verlangen solche Kompetenzen, die sich in der Vergangenheit als erfolgskritisch herausgestellt haben oder gar gefehlt hatten.

Für diese Studie wurden IT-bezogene Stellenanzeigen in dem Jobportal www.stepstone.de untersucht. Es wurden diejenigen ausgewählt, die RE&M-Aufgaben enthielten. Dies waren rund 10% der analysierten IT-bezogenen Anzeigen. RE&M-Aufgaben werden nicht immer als „RE&M“ bezeichnet, sondern können viele Namen tragen. Darum konnten die entsprechenden Anzeigen nicht automatisch, sondern nur von Hand extrahiert werden. Die häufigsten Schlüsselwörter, die auf RE-Aufgaben hinwie-

sen, waren: Fachkonzept, Kundenanforderungen, Problemanalyse, Analyse von Geschäftsprozessen, Abstimmen der Anforderungen mit unseren Kunden, fachliche Entwicklung von Software-Produkten, Analyse von Prozessen und deren Aufbereitung für die Kollegen der Softwareentwicklung.

Dann wurde der Text der Anzeige klassifiziert: Titel, Aufgaben und geforderte Kompetenzen wurden in einer Tabelle gesammelt und Kategorien zugeordnet.

Der Titel „Requirements Engineer“ existiert kaum. 2009 wurde er nur ein einziges Mal in den 141 Anzeigen gefunden, die RE&M-Aufgaben enthalten, in 2012 waren es auch nur 3 von 67. RE&M wird ansonsten durchgeführt von Beratern, Software Engineers, Entwicklern und anderen. Tabelle 1 zeigt die Verteilung der Positions-Titel derjenigen Personen, die RE machen, für 2009 und 2012. Die größte Gruppe darunter sind die Berater. Im Vergleich von 2009 mit 2012 zeigt sich der Trend, dass es mehr Requirements Engineers gibt, aber auch mehr Architekten, etwas mehr Software Engineers und auch mehr ausdrückliche Doppelpositionen unter denjenigen, die im RE arbeiten. Andererseits gibt es 2012 deutlich weniger Entwickler, die das RE zusätzlich miterledigen. Möglicherweise stellte sich diese Kombination von Entwickler-Position und RE-Aufgaben als nicht ideal heraus.

Positions-Titel	Anzahl (Anteil) 2009	Anzahl (Anteil) 2012
Requirements Engineer	1 (0,7 %)	3 (4,5 %)
Berater	72 (51 %)	29 (43,3 %)
Architekt	12 (8,5 %)	11 (16,4 %)
Entwickler	36 (25,5 %)	5 (7,5 %)
Vertriebsmitarbeiter	2 (1,4 %)	0
Projektmanager	3 (2,1 %)	2 (3,0 %)
Software Engineer	10 (7,1 %)	7 (10,4 %)
Doppelposition	5 (3,5 %)	7 (10,4 %)
Andere	0	3 (4,5 %)

Tabelle 1.: Verteilung der Positions-Titel derjenigen, die RE machen, 2009 und 2012

RE&M ist keine Vollzeitbeschäftigung. Dies zeigt sich nicht nur daran, dass es wenige Positionen mit dem Titel „Requirements Engineer“ gibt, sondern zusätzlich auch an den Aufgaben-Listen: Wer RE&M macht, hat noch viele weitere Aufgaben. Tabelle 2 zeigt, welche Nicht-RE&M-Aufgaben als Aufgaben zusätzlich zu RE&M wie oft genannt wurden. Während die Lösungskonzeption nach wie vor die wichtigste Nicht-RE&M-Aufgabe ist, die mit RE&M kombiniert wird, sank deren Häufigkeit von 2009 nach 2012 von 77,3% auf 61,2%. Genauso sank auch die Häufigkeit anderer Aufgaben, die techni-

sches Wissen verlangen, wie Realisierung, Einführung und Machbarkeitsanalyse/ Kostenschätzung. Bei den nicht-technischen Aufgaben blieben die Anteile ungefähr gleich. In Folge dieser Entwicklung nahm die Anzahl der Aufgaben, die zusätzlich zum RE&M gemacht werden sollen, von 3,23 (2009) auf 2,79 (2012) ab. Dies reflektiert vermutlich die Erkenntnis, dass RE&M zunehmend als Tätigkeit gesehen wird, die nicht parallel zur Entwicklung ausgeübt werden kann, und die einen größeren Anteil des Arbeitstags einnimmt als früher angenommen. Technisches Wissen wird nach wie vor zu 76% verlangt. Das heißt, die Entlastung des RE&M-Personals von technischen Aufgaben bedeutet keine Entlastung von der Notwendigkeit technischen Wissens.

Aufgabe	Anteil 2009	Anteil 2012
Lösungskonzeption	77,3 %	61,2 %
Realisierung	53,9 %	44,8 %
Einführung	41,1 %	23,9 %
Qualitätssicherung	37,6 %	38,8 %
Projektmanagement	34,8 %	31,3 %
Wartung/ Hotline	24,8 %	23,9 %
Dokumentation/ Training	22,7 %	25,4 %
Vertrieb	19,1 %	19,4 %
Machbarkeitsanalyse / Kostenschätzung	12,1 %	10,4 %

Tabelle 2: Nicht-RE&M-Aufgaben, die 2009 und 2012 zusätzlich zu RE&M in den Anzeigen genannt wurden

Nur ungefähr ein Drittel der Stellenanzeigen verlangen RE&M-spezifische Kompetenzen: 2009 waren es 37% (51) und 2012 nur 34% (23). Diejenigen Anzeigen, die die Lösungskonzeption als Aufgabe nennen, verlangen technisches Wissen zu 82% (2009) bzw. 76% (2012), und für die Entwicklung wird technisches Wissen zu 86% (2009) bzw. 83% (2012) gefordert. Unter denjenigen Anzeigen, die Projektmanagement-Aufgaben nennen, verlangen nur 26% Projektmanagement-Wissen (2009) bzw. 14% (2012).

Auf jeden Fall ist RE&M keine Aufgabe für Berufsanfänger: 72% (2009) bzw. 73% (2012) der Anzeigen wünschen oder verlangen vorherige Berufserfahrung. Studium oder Ausbildung werden zu 89% (2009) bzw. 85% (2012) vorausgesetzt. Weitere wichtige Kompetenzen sind: 94% der Anzeigen verlangen Softskills (die Top 3 Softskills sind Teamfähigkeit, Englischkenntnisse und Kommunikationsfähigkeit), 76% technisches Wissen. Die Forderung von oder der Wunsch nach Domänenwissen sank von 50% auf 34%. Dies kann zweierlei bedeuten: Entweder hat man erkannt, dass Domänenwissen

schnell angeeignet werden kann oder man möchte aufgrund des Fachkräftemangels die Anzahl der möglichen Bewerber nicht unnötig einschränken - oder beides. Eine Langfassung dieser Studie können Sie nachlesen in [He13]. Gerne hätten wir noch den Einfluss der Unternehmensgröße untersucht, da zu erwarten ist, dass in größeren Unternehmen Rollen klarer definiert sind [DBH13]. Jedoch war in den meisten Stellenanzeigen die Unternehmensgröße nicht angegeben.

Wie diese Analyse zeigt, kann in der deutschen Praxis der Softwareentwicklung nicht von der Existenz expliziter RE&M-Positionen ausgegangen werden, zumindest nicht in relevantem Umfang. RE&M wird zusätzlich zu mehreren weiteren Aufgaben gemacht, zumeist von Beratern, Architekten und Entwicklern. Dies ist ein Anzeichen dafür, dass in der Praxis die Arbeit im RE&M querschnittlich auf viele Rollen aufgeteilt wird, ohne eine eigene Rolle „Requirements Engineer“ zu definieren. Die Situation in der Praxis scheint noch unklarer zu sein als in den Vorgehensmodellen.

3.3 Partizipation von Fachabteilungen im RE&M-Prozess

Weißbach untersuchte in einer qualitativen Studie [We13] die Partizipation von Fachabteilungen im RE-Prozess. Die Ergebnisse zeigen, dass in der organisationalen Realität die „Lehrbuchsituation“, in der gut qualifizierte „Requirements Engineers“ den Anwendern gegenüber treten und von diesen die zu implementierenden Anforderungen erheben, nur eine Möglichkeit ist, die nicht den Regelfall darstellt.

Weißbach fand folgende Situationen vor, wobei der (selbst eingeschätzte) Erfolg der Firmen in ihrem Vorgehen offenbar kaum Abhängigkeit von dem gewählten Modell des RE&M-Prozesses aufwies:

- Es wird gar keine Anforderungsanalyse durchgeführt, statt dessen „einfach losgelegt“. (Einzig dieses Modell wurde als erfolglos beschrieben.)
- Fachabteilungen kennen ihre Aufgaben „sowieso am besten“ und definieren die Anforderungen selbst; ggf. ist überhaupt keine IT-Abteilung vorhanden.
- In den IT-Abteilungen ist so viel Fachwissen über „das Geschäft“ vorhanden, dass diese die Anforderungsdefinition selbst initiieren bzw. komplett durchführen, ggf. nach Rücksprache mit den Fachabteilungen.
- IT-Abteilungen nehmen über die technisch fokussierten Aufgaben hinaus auch Organisationsaufgaben wahr, z.B. im Bereich des Prozessmanagements oder der Prozessoptimierung.
- Anforderungen in Projekten werden durch eine zentrale (IT-) Abteilung analysiert, im „laufenden Geschäft“ hingegen vom Linienmanagement.
- Projekte werden auf der Ebene der Geschäftsführung aufgesetzt, die Ziele definiert und diese – ggf. mit Externen – auf einzelne Anforderungen herunterbricht. Dies kann im Kontext von Change-Management-Prozessen relevant

sein, da hier das Beharrungsvermögen der mikropolitischen „Routinespieler“ [Or90] diskontinuierliche Entwicklungen zu verhindern sucht.

Tatsächlich scheinen gerade Klein- und Mittelbetriebe (KMUs) in vielen Fällen ohne exakte Rollen- und Vorgehensmodelle zu arbeiten, sondern auf eine Weise, die man als „chaotisch-agil“ bezeichnen kann.

Weitere Fragen tauchen auf, wenn nicht das Projekt alleine betrachtet wird, sondern das Softwareartefakt während seiner gesamten Lebensdauer: Eine wesentliche Aufgabe von Führungskräften, gerade im operativen Bereich, also im Linienmanagement, ist es, die verantworteten Prozesse laufend zu optimieren. Damit zählt aber letztendlich auch die Anforderungsanalyse zu den Aufgaben dieser Linienmanager, unabhängig davon, ob die Ergebnisse dieser Analyse in dedizierte Projekte münden oder nicht (sondern „nur“ in laufende Verbesserungen).

Tatsächlich also ist die Beteiligung der Mitarbeiter aus Fachabteilungen am RE&M in der Organisationspraxis wesentlich differenzierter zu sehen, als es Vorgehensmodelle suggerieren. Unterschiede zwischen Betrieben scheinen eher organisationsindividuell zu sein bzw. branchenspezifisch (so sind formale Dokumentationen in Banken auf Grund von Revisionsanforderungen unabhängig von deren Unternehmensgröße verbreitet). Für die Zukunft wird eine zunehmende Bedeutung von Fachabteilungen bei der Entscheidung über IT-Ausgaben erwartet [Cg13]. Dies dürfte die Funktion der Fachabteilungen im RE&M-Prozess weiter stärken.

Auf Grund der Beschränkung des Samples auf deutsche Unternehmen konnte nicht der Einfluss nationaler Spezifika in der Ausbildung und der Beschäftigung Auswirkungen geprüft werden. Ob bzw. welchen Einfluss die deutsche Tradition einer dualen, nicht-akademischen Ausbildung in Verbindung mit relativ hoher Beschäftigungsdauer auf die Durchführung des RE&M-Prozesses gerade in KMUs besitzt, müsste mittels eines anderen Forschungsdesigns untersucht werden.

4 Zusammenfassung und Ausblick

Nicht nur in gängigen Vorgehensmodellen, sondern insbesondere auch in der Praxis scheint für das RE&M oft keine eigene, einzige Rolle definiert zu sein: Wer im RE&M arbeitet, macht noch viele andere Tätigkeiten zusätzlich, und mehrere verschiedenste Rollen arbeiten im RE&M.

Eine Position „Requirements Engineer“ ist selten. Stattdessen wird RE&M durchgeführt von Fachabteilungen, Beratern, Software Engineers, Software-Architekten, Entwicklern und Projektleitern. Sie machen RE&M zusätzlich zu durchschnittlich drei weiteren Aufgaben, am häufigsten Lösungskonzeption. RE&M ist keine Arbeit für Berufsanfänger: 73% der Stellenanzeigen wünschen oder verlangen vorherige Berufserfahrung. Weitere wichtige Kompetenzen sind: 94% der Anzeigen verlangen Softskills, 76% technisches Wissen, während nur 34% RE-Kenntnisse erwähnen.

Wir halten diese Situation des RE&M in der Praxis für verbesserungswürdig, da es sich bei RE&M um eine Tätigkeit handelt, die spezielles Wissen erfordert. Außerdem muss eindeutig geklärt sein, wer das RE&M koordiniert und für die Qualität der Anforderungen verantwortlich ist. Ansonsten kann diese erfolgskritische Tätigkeit nicht erfolgsfördernd durchgeführt werden.

Wir planen weitere Forschungen zur Praxis des RE&M. So sind weitere Untersuchungen von Stellenanzeigen in anderen Ländern und zu späteren Zeitpunkten (z.B. 2015) geplant, außerdem eine Interview-Studie, in der genauer die alltägliche Aufgaben- und Rollenverteilung in Projekten untersucht werden soll, wobei die Unternehmensgröße erhoben wird und Daten mindestens aus Deutschland und den Niederlanden miteinander verglichen werden. Weitere Forschungsvorhaben werden sich mit RE&M in der laufenden Linienarbeit, außerhalb von Projekten, beschäftigen.

Literaturverzeichnis

- [AP05] Alenljung, B.; Persson, A.: Factors that Affect Requirements Engineers in their Decision Situations: A Case Study. Proc. of REFSQ Workshop 2005.
- [AW05] Aurum, A.; Wohlin, C.: Requirements engineering: Setting the context. Engineering and Managing Software Requirements, Springer, 2005. Siehe: http://wohlin.eu/rm_chapter05.pdf
- [Cg13] Capgemini: IT-Trends 2013. Capgemini, Berlin 2013, http://www.de.capgemini.com/sites/default/files/resource/pdf/capgemini-studie_it-trends_2013.pdf
- [DBH13] Daneva, M.; Buglione, L.; Herrmann, A.: Software Architects' Experiences of Quality Requirements: What we Know and What we do not Know? REFSQ 2013 Konferenz, April 2013, Essen.
- [FHW06] Fahney, R.; Herrmann, A.; Weißbach, R.: A new dimension in the distinction between Requirements Engineering from Project Management. Bericht des GI-Arbeitskreises „Requirements Engineering und Projektmanagement“, Oktober 2006.
- [FHW07] Fahney, R.; Herrmann, A.; Weißbach, R.: Wie viel Requirements Engineering steckt im Software Engineering? Workshop “Wie viel Requirements Engineering steckt im Software Engineering?“, SE 2007 Tagung, 27.03.2007, Hamburg, Germany.
- [He13] Herrmann, A.: Requirements Engineering in Practice: There is no Requirements Engineer Position. REFSQ 2013 Konferenz, April 2013, Essen.
- [Iee90] IEEE: IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology. IEEE, Washington, USA, 1990.
- [Kr12] Klendauer, R. et. al.: Towards a competency model for requirements analysts. Information Systems Journal, Number 6, Vol. 22, Januar 2012.
- [NL03] Neill, C.J.; Laplante, P.A.: Requirements Engineering: State of the Practice. IEEE Software 20(6) 2003.
- [NSK00] Nikula, U.; Sajaniemi, J.; Kalviainen, H.: Management view on current requirements engineering practices in small and medium enterprises. Proc. of Australian Workshop on Requirements Engineering, 2000.
- [Or90] Ortmann, G.; Windeler, A.; Becker, A.; Schulz, H.-J.: Computer und Macht in Organisationen. Mikropolitische Analysen. Westdeutscher Verlag, Opladen, 1990.
- [PM13] PMI: PMBOK 5: A Guide to Project Management Body of Knowledge (PMBOK® Guide) - Fifth Edition, 2013.

- [Ru07] Rupp, C.: SOPHISTen: Requirements-Engineering und Management. Professionelle, iterative Anforderungsanalyse für die Praxis. Carl Hanser Verlag München, 4., aktualisierte und erweiterte Auflage, 2007.
- [St01] Standish Group: Extreme CHAOS, 2001, http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf (nicht mehr online).
- [We13] Weißbach, R.: How Business Departments Manage the Requirements Engineering Process in Information Systems Projects in Small and Medium Enterprises. InSite 2013, Porto, Portugal, 2013, <http://iisit.org/Vol10/IISITv10p539-549Weissbach0093.pdf>
- [ZWO01] Zowghi, D.; Damian, D.; Offen, R.: Field Studies of Requirements Engineering in a Multi-Site Software Development Organization. Australian Workshop on Requirements Engineering, Univ. of New South Wales, 2001.

Prioritisation and selection of the right business and IT requirements in the software engineering process

Feyyat Kaymaz

Canon Europe Ltd
3 The Square, Stockley Park
Uxbridge, Middlesex, UB11 1ET, United Kingdom
feyyat.kaymaz@canon-europe.com

Abstract: The overall prioritisation and selection of the right business and IT change requirements are an important area in the software engineering process as this is a critical factor that can ultimately impact the success of the business. In most cases the overall prioritization of business and IT change requests and the steps involved in the decision making process are not transparent enough for all stakeholders. Therefore it is essential that the business change request management process is visible and understandable in order to make the right decisions which are acceptable for all stakeholders. For this reason the prioritization and selection process should be built in an effective and efficient way by using quantitative tools, for example, by following Six Sigma methodology. This article will explain how we manage the prioritization and decision making process.

1 Introduction

One of the neglected or less handled topics in the software engineering and service management process is the prioritisation and selection of the right business and IT requirements in order to conceive and implement the proper software solution. This topic is also very important for the risk management and decision process.

The business and ICT teams gather daily business change requests from users and customers respectively. The users and customers expect that all of their requirements should be considered and implemented if they have a business change request in order to fulfil their tasks and to meet customer needs. As long as you have enough budget and capacity to capture and implement these business and IT requirements it is unlikely that anybody in the company will ask how efficiently and effectively these requirements are prioritised, selected and implemented. However, in times of limited budget, limited capacity, higher quality requirements from your business and customers, and higher competition; you should make your decision more carefully in order to select the business and IT requirements which are most important for your customers and by which you can meet the customer and business requirements. As not all business and IT requirements can be chosen and implemented, due to the reasons which are mentioned above, you should ensure the right choice is made between the business and IT requirements (in form of business and change requests) in order to manage your business in an efficient and effective way.

1.1 Problem

The overall prioritization and selection of the right business change requirements are a challenge for the stakeholders which can ultimately impact the overall success of the business. In most cases the business change request management process is not clear, not explicitly defined or not transparent enough for all stakeholders. For example, it may not be clear why their requirements are not implemented; there might also be confusing surrounding who is responsible for prioritising and approving business change requests. Therefore it is essential, that the business change request management process is clear and coherent in order to allow stakeholders to make the right decisions which are acceptable for all stakeholders.

In order to achieve this goal I suggest that this process should be built utilising a quantitative rather than a qualitative technique. That means the prioritization should be made in a quantitative way and hereby all of the business and IT requirements should be quantified. By working this way you have a ranking list of all of the business and IT requirements. According to the ranking you can select the most important/top requirements and determine where you should focus in the software engineering process. This gives the benefit of allowing you to invest in any critical and urgent requirements, therefore keeping your users and customers happy whilst reducing the risk of selecting of wrong requirements, which may jeopardise the success of the business.

This concept is currently successfully implemented in the small change request management process at Canon Europe in the Services & Support organisation.¹ In this article I would like to demonstrate the following:

- How we have organised the prioritization and selection process,
- How it is working,
- What experiences we have gained,
- What you can learn and apply in your business and IT processes.

1.2 Issues

The main issue is the transparency of the prioritization and selection process of the business and IT requirements for the stakeholders, as explained in the previous section, specifically the reengineering of the business and IT requirements management process in order to ensure the quality (efficiency and effectiveness) of the offered services and their costs.

The overall business change request management and the prioritization and decision process should be visible for all of stakeholders which means that every involved party can see the End-to-End process and act accordingly across the business functions (such as Marketing, Sales, Supply chain, Finance and IT etc.). This interaction across the business functions should be built in an efficient and effective way. In order to ensure the quality of this process a set of general decision criteria should be defined in terms of

¹ Small change means that the implementation of a change request costs not more than 20mandays effort.

time, budget and performance (features). In addition to this, further specific decision criteria should be defined in terms of resources, risk assessment and criticality.

1.3 Goals/Benefits

The following benefits could be achieved by this process improvement project in a business change request management process:

- Ensuring the visibility of the overall business and IT requirements management process for all stakeholders
- Eliminating any confusion which is linked with to the change request management (CHRM) process and making the process transparent.
- To allow you to make a better decision in order to work effectively and efficiently and to achieve better business results, thus increasing customer satisfaction.

2 Stakeholders in the business change request prioritisation and decision process

In order to prioritise and select the business and IT change requests properly (and to make rights decisions) the priorities of all of stakeholders should be considered.[RSH06] This decision should be made in a consensus of the stakeholders in order to get their acceptance and meet the customer requirements. This is one of the key factors in minimising the business and IT risks. For example, selecting of minor or (in a worst case) the wrong business and IT requirements, increases the risk of the implementing a poor service or software solution. On the other hand, you can increase the acceptance of management decisions and business success.

In the tension field of the stakeholders there are three parts for stakeholders to prioritise and select the (right) business and IT requirements (change requests) (see figure 1). The business team(s) which requests the business changes (technology, process, policy and people), the management who decides on the right business change requests and the ICT team who analyses, plans and implements the business change requests in the business and IT environment. In this tension field of the stakeholders, there should, therefore, be a balance between these three stakeholders.

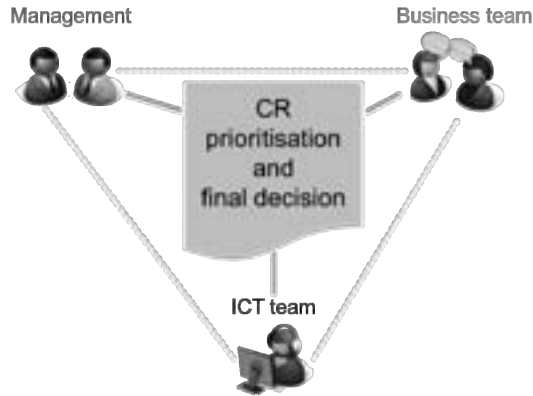


Figure 1: Tension field in the decision making process

3 Business change request prioritization and decision process

The prioritization of business change requests involves making multiple decisions. This means that each priority of all stakeholders should be taken into account as explained above. Now in the following section, I will explain which priorities can and should be considered. The first priority is given on a long term basis by the management on a general criterion. This follows the business priority which is made by the business teams (on short or long term) and the IT priority which is made by the ICT team. The final priority is made by the combination of the three priorities of the stakeholders where a ranking list of prioritised business change requests is created.

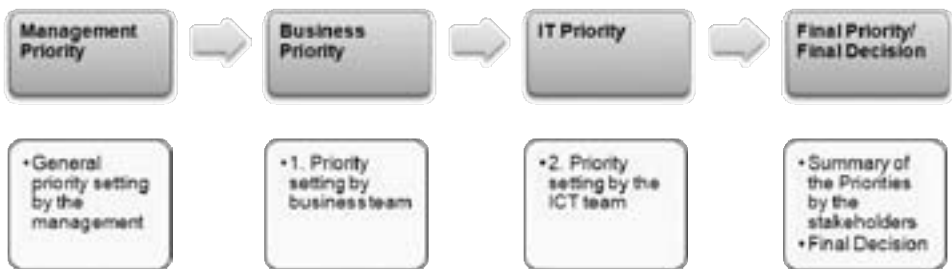


Figure 2: Prioritization and decision process

The (general) Management Priority for the business change requests (projects and enhancements) is given by management. The general priority should be in line with the business strategy and objectives. Management priority is achieved by considering the Time, Cost and Performance of the implementation of the business change requests. For the Management Priority/Decision, the costs of implementation could have more weight in times of limited budget than other decision criteria Time and Performance.

The Business Priority is made and given by the business teams. According to the given management priority, business teams should prioritise their business change requests,

which come from users, business teams and national sales and operation companies, by the RPN (Risk Priority Number) analysis. Business Priority is based on the management priority and can be assessed and respectively calculated by the RPN (SOD score) analysis within a team discussion which will be explained in the next section.

Once the business priority is done, the ICT team should prioritise the business change requests by the IT risk analysis. IT Priority is made and given by the ICT teams. IT Priority is based on the weighted criteria of the ICT teams which are for example the following points:

- Maintainability
- Performance
- Functionality
- Security
- Integration
- Etc.

For the final decision and the final priority of the business change requests the management should take the results from the RPN (Risk Priority Number) analysis, IT risk analysis, Budget, Time and Resources into account.

4 Prioritization, ranking and decision process model

4.1 Management Priority

The Management Priority provides a general and strategic aspect to the decision making process on a long term basis for all business change requests (new projects and enhancements). This means that the management decision making process should be aligned with the objectives of the business in different areas, which can be defined in a Strategy Map (i.e. by using a Balanced Scorecard). [RSH06]



Figure 3: Cost situation and management priority

In order to achieve the business objectives, the management also defines priorities for its decisions by considering four components of its resources; Time, Cost, Performance and People. Whenever the management prioritises a business change request it should always consider the given circumstances of the business. For example, in figure 3, the management prioritises the cost of the implementing the business change request over the other decision criteria and says the cost situation is a constraint which should be handled with high priority. The other conditions Time, Performance and People should be accepted and optimized.

4.2 Business Priority

In order to determine the business priority a business risk assessment should be performed with the use of the RPN analysis by the business teams. The RPN analysis consists of the risk parameters Severity, Occurrence and Detection (SOD score) of a business change request. At the same time, the possible risks and their effects (as well as the possible reasons of these risks) can be determined. The RPN is an important component of a FMEA (Failure Mode and Effect Analysis) which is known from Six Sigma methodology and engineering process in different industries and companies.² By the FMEA tool the potential risks of a project or design and its effects on the product, process and systems can be identified and a control mechanism can be defined and applied by appropriate responses and actions [KN07][Ka12].

² See [Ka12] for the benefits of using Six Sigma methodology and its tools in the software quality engineering process.

Business Change Request	Potential risks	Potential effects of the risks	Severity (S)	Occurrence (O)	Detection/Current Control (D)	Risk Priority Number (S*O*D)	Ranking
Availability of an online billing service	Manual process for billing	Long process cycle time	9	7	5	315	1
Availability of an online invoicing service	Invoices with errors	Wrong amount of money collected	9	5	5	225	2

Figure 4: Business Risk Assessment and Business Priority determination by the RPN Analysis (SOD score)

We use the Risk Priority Number (SOD score) from the Six Sigma tool FMEA in our risk management, prioritization and decision process and have had positive experiences. The risks and their effects can be calculated by the following formula:

$$\text{Risk Priority Number (RPN)} = S * O * D$$

The Risk Priority Number can be calculated by the multiplication of the following parameters:

- Severity (S)
 - Impact of the risks on the business
 - Rating Scale 1-10 (1=Minor - 10=Hazardous)
- Occurrence (O)
 - Probability of the risks
 - Rating scale 1-10 (1=Unlikely – 10=Very High/Critical)
- Detection (D)
 - Manageability of the risks
 - Rating Scale 1-10 (1=Almost certain – 10=Almost Impossible)

After the execution of RPN Analysis a ranking list of the business change requests can be prepared which shows the business priority as this is determined by the team discussion (See Figure 4).

4.3 IT Priority

On the other hand, IT Priority is made and given by the ICT teams who should prioritise the business change requests by the IT risk analysis of their implementation within team discussions. IT Priority is based on the weighted criteria of the involvement of different ICT teams along the progressing of the business change requests. The ICT teams can be listed by the following tasks:

- Maintainability
- Performance
- Functionality
- Security
- Integration
- etc.

Business Change Request	Potential Risks	Potential effects of the risks	Required Functionality	Criteria/Weight						IT Risk Number	Ranking
				Maintainability	Performance	Functionality	Security	Integration	Other		
				4	6	6	7	3	1		
Availability of an online invoicing service	Invoices with errors	Wrong amount of money collected	Ability to create invoices and correct errors automatically	2	6	7	8	2	3	151	1
Availability of an online billing service	Manual process for billing	Long process cycle time	Ability to bill automatically	3	4	3	7	3	3	115	2

Figure 5: IT Risk Assessment

In order to determine the IT Priority each ICT team assigns a priority number on a scale of 1-10 (where 1 is low and 10 is high) to the business change request. The priorities of each ICT team will be weighted according to their involvement in the implementation of the business change request. This means, the criteria weight of each team is determined within the discussion of ICT teams. After the criteria are weighted for each team, the risk values for each business request should be determined during the discussions with the relevant ICT teams. The IT risk value is the correlation of the business change requests and the weighted criteria. The IT Risk Number is determined by the sum of the correlation of each criterion and its weight in the implementation of this business request. This can be calculated by the following formula:

$$\text{IT Risk Number} = \sum_{i=1}^n \text{Criterion}_i \cdot \text{Weight}_i$$

The IT Risk Number should be determined for each business change request in order to create a ranking list at the end of the IT Risk Assessment. In the next step the IT Priorities can be compared with the business priorities in order to make a final ranking list. This is explained in the following section.

4.4 Final Ranking

The Final ranking is a simple comparison of the initial business priority of the business change requests vs. business risk and IT risk assessments. The initial business priority is specified when a business request is raised. This priority could be in a scale of 1-5, (where 1 is low and 5 is high). During the verification of this business request by a busi-

ness analyst and the owner of the business request a risk priority number (SOD score) is determined as explained in the previous chapter. A technical risk assessment is also performed for each business request. Once the business and IT risk assessments are completed, the business and IT priorities should be determined. In the next step the final ranking can be created. The final ranking will be prepared by comparing the prioritised business change requests in a ranking list. The higher the scores of the business and technical risk assessment results the higher their ranking in the final ranking list. Once the final ranking is created, a final decision can be made. This will be explained in the next chapter.

Business Change Request	Severity (S)	Occurrence (O)	Detection/Current Control (D)	Risk Priority Number (S*O*D)	Criteria/Weight						IT Risk Number	Final Ranking
					Maintainability	Performance	Flexibility	Security	Integration	Other		
Availability of an online billing service	9	7	5	315	3	4	3	7	3	3	115	1
Availability of an online invoicing service	9	5	5	225	2	6	7	8	2	3	151	2

Figure 6: Final ranking according to the Business and IT Risk Assessments

4.5 Final Decision

After the business and IT risk assessments have been completed, a decision board, consisting of the stakeholders (see the chapter 2), should take the final ranking of the change requests and different circumstances of the stakeholders into account for the final decision. This means that the decision board’s decision making process is not completed by simply selecting change requests from the final ranking list, but also by considering the relative business benefits. The relative business benefit means that the most important business areas can get their business change requests considered first in the Final Ranking list. This is determined by the management decision board focusing on the most important business areas which are defined by the business strategy. This aspect can also have a negative effect on the less important business areas which could struggle to get their business change requests considered with high priority. This is also a challenge for the management decision board to always ensure that they select the right business and IT change requests. In order to mitigate the negative effect of this business strategy aspect on the less important business areas and to ensure that there exists a healthy balance between the different levels of the business, we introduced a business fairness mechanism into the decision process.

Business Teams	Budget Allocation Criteria		Final Allocation Weight in %
	Current Business Priority	Business Fairness Factor	
	75%	25%	100%
Team 1	32	5	37
Team 2	30	5	35
Team 3	7	5	12
Team 4	3	5	8
Team 5	3	5	8
Total	75	25	100

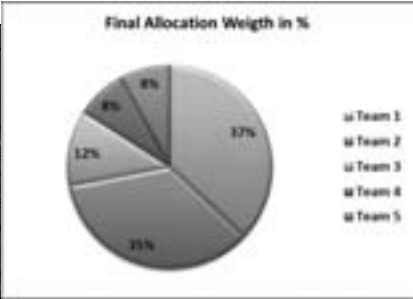


Figure 7: Budget Allocation Matrix

The business fairness mechanism should come into effect, if a limited budget needs to be allocated between the smaller and bigger business teams for the implementation of their business change requests. We call this point “the business fairness factor” when the budget will be allocated according to this final ranking list. The business fairness factor reduces the risk that smaller business teams cannot get their change requests approved. This is an important point as it will allow smaller business teams to be made aware of the relative importance of their business change requests in comparison with those of the larger business teams. In figure 7, the budget can be allocated according to the relative importance of the business teams (current business priority). The business fairness factor describes the situation and ensures that every team gets the same amount of the money (25%) from the budget, regardless of their relative importance in the business. The rest of the money (75%) can be allocated according to the current business priority. See the column “Current Business Priority” in figure 7.

5 Risk Management Model - Qualitative vs. Quantitative Rating

In practice, different risk assessment and prioritization models are used, in most cases e. g. MoSCoW-Principle or three level risk matrix (low, Medium, High) etc. [Ge04] For example, if the rating of the risk assessments is completed by using a three level risk matrix consisting of a simple scale of importance (low, medium, high) for business change requests (see figure 8), then the results of this business risk assessment are not sufficient for an optimal business decision.

Impact	Medium	High	High
	Low	Medium	High
	Low	Low	Medium
Probability			

Figure 8: A typical Risk Management Model

The problem with these three levels is that if there are hundreds of business change requests and a decision board has to correctly select the valid business change requests. If a ranking list of selected business change requests is created by the three level risk matrix then there are three groups of ranked business change requests but the decision board cannot judge simply which of them in each group is most important, e. g. if there are 100 business change requests and each group has the same number business change requests, then we cannot say that every business request has the same importance in each group.

		Detection (1-10)		
		1-3	4-6	7-9
Severity (1-10)	7-9	Medium	High	Critical (10)
	4-6	Low	Medium	High
	1-3	Minor	Low	Medium
		1-3	4-6	7-9
		Occurrence (1-10)		

Figure 9: Qualitative vs. Quantitative Risk Assessment

This point was the main problem (and caused many long discussions without any reasonable decision) for the stakeholders in the risk management and prioritization process. Instead of this, prioritising with a three level importance/priority scale which is more a qualitative rating/ranking and not adequate instrument for making a right decision, we proposed that a quantitative rating instrument should be applied. In the previous chapters we explained the RPN/SOD (severity, occurrence, detection). If we put the elements of

the RPN in the following matrix, we can calculate the relative importance and the priority of each business requirements exactly (see figure 9). Instead of just saying that a certain business request is in a group of priority levels and has the same priority with the other business change requests, we can calculate the exact position of a business request according to the calculation of risks and effects by their severity, occurrence and detection.

6 Conclusion

The prioritization and selection of the right business change requirements are an important area in the software engineering process as this is a critical factor that can impact the overall success of the business. Therefore the business change request management process should be transparent for all stakeholders and built in an effective and efficient way by using quantitative methods and tools. We could improve and optimize our prioritization and decision making in the business change request management process by the using Six Sigma methodology which helps the application of quantitative methods and tools and makes the results more accurate, understandable and acceptable for the stakeholders. We could also reduce the variation of different business initiation and change request management processes and their cycle time. The prioritization and decision process is now visible and transparent for the stakeholders; the discussions and the risk of the suboptimal decisions have been reduced and the planning of resources and budget allocation is improved by the new prioritization and decision making process. The active involvement of all of stakeholders is also essential.

References

- [Ge04] George, M. L. et al.: The Lean Six Sigma Pocket Toolbook, McGraw-Hill, 2004.
- [Ka12] Kaymaz, F.; Software Quality Engineering mit Six Sigma Methoden, In (Linssen, O.; Kuhrmann, M. Hrsg.): Qualitätsmanagement und Vorgehensmodelle: 19. Workshop der Fachgruppe Vorgehensmodelle (WI-VM) der Gesellschaft für Informatik e.V., Düsseldorf, Shaker Verlag, Aachen, 2012.
- [KN07] Kaplan, R. S.; Norton, D. P.: Using the Balanced Scorecard as a Strategic Management System, Harvard Business Review, Managing for the long term, 2007.
- [Ma09] Ma, Q.: The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review, Auckland University of Technology, 2009.
- [RSH06] Reynolds, S. J.; Schultz, F. C.; Hekman, D. R.: Stakeholder Theory and Managerial Decision-Making: Constraints and Implications of Balancing Stakeholder Interests, Journal of Business Ethics, 64, Springer Verlag, 2006.

Ausgewählte Problemfelder und Lösungsansatz des Requirements Engineering in der Telekommunikationsbranche

Alexander Rachmann, Sven Eselgrimm, Frank Engel

Unit Customer & Service Consulting
7P Solutions & Consulting AG
Calor-Emag-Straße 1
40878 Ratingen
alexander.rachmann@7p-group.com
sven.eselgrimm@7p-group.com
frank.engel@7p-group.com

Abstract: In diesem Erfahrungsbericht aus der Telekommunikationsbranche werden aktuelle Problemfelder der Unternehmensberatung zum Requirements Engineering vorgestellt. Ursachen der Problemfelder werden analysiert (vorhandene Prozesse in Unternehmen, mangelnde Toolunterstützung, Enterprise Mobility als zentraler Faktor für das RE). Die unzureichende Verbindung des Vorgehensmodells Requirements Engineering zur operativen Umsetzung wird dabei als maßgeblich identifiziert. Ein Lösungsansatz wird vorgestellt.

1 Einleitung

Requirements Engineering ist eine Teildisziplin des Software Engineering, die den Umgang mit Anforderungen an die zu entwickelnde Software behandelt. Akademisch wurden mehrere Ansätze (siehe Kapitel 2) hierzu erarbeitet. Trotzdem lässt sich in der Praxis des Software Engineerings erkennen, dass der Umgang mit Anforderungen oftmals unzureichend funktioniert. Unsere Hypothese ist, dass die – durchaus komplexen – bestehenden Vorgehensmodelle aus verschiedenen Gründen schwierig umzusetzen sind (siehe Kapitel 3). Abhilfe kann eine Sammlung von niedrigschwelligen und kleinschrittigen Umsetzungshandreichungen und Dokumentenvorlagen schaffen.

Dieser Beitrag basiert auf den Erfahrungen der Fachgruppe Requirements Engineering der 7P Solutions & Consulting AG (kurz: 7P). Der Schwerpunkt der Arbeit der Fachgruppe liegt in der Telekommunikationsbranche. Die Erhebung der vorgestellten Hypothesen basiert nicht auf einer wissenschaftlichen Methode, sondern dem Erlebnisalltag der Gruppe. Es liegt nahe, dass ähnliche Erfahrungen auch in anderen Branchen gemacht werden können.

2 Vorgehensmodelle im Requirements Engineering

Eine umfassende Einführung in die Theorie (und verschiedene Ansätze) des Requirements Engineering (RE) ist schon aus Platzgründen in diesem Beitrag nicht sinnvoll. Daher sei nur kurz die Sicht der Telekommunikationspraxis angerissen: Allgemein akzeptiert ist die Interpretation des Requirements Engineering als ein Prozess, in dem die Ziele eines Softwaresystems erhoben und dokumentiert werden [Po08]. In diesem Prozess spielt die Kommunikation mit den Personen, die ein Interesse an dem System oder dem Entwicklungsprojekt besitzen („Stakeholder“), eine besondere Rolle. Im deutschsprachigen Raum gilt [Po08] als wissenschaftliches Referenzwerk, die Praxissicht wird dominiert von [Ru09]. [Po08] beschreibt drei Kernaktivitäten (Gewinnung, Dokumentation und Übereinstimmung der Anforderungen) und zwei Querschnittsaktivitäten (Validierung und Management der Anforderungen) des RE.

RE ist eine grundsätzlich branchenneutrale Methodik. Da die Branchen in der Art ihrer Anforderungen und ihren Prozessen sich stark unterscheiden, muss RE branchenspezifisch durchdacht und umgesetzt werden. Die Branchen sind dabei auf unterschiedlichen Entwicklungsständen, zum Beispiel ist die Automobilindustrie im RE sehr weit fortgeschritten (siehe z.B. [Ho12], [WW03]).

Als Requirements Engineers in der Telekommunikationsbranche beobachten wir, dass in dieser Branche erst langsam die Erkenntnis wächst, dass RE die Softwareentwicklung und die Software verbessern kann. Es wurden einige grundlegende Problemstellungen identifiziert, die wir im Folgenden vorstellen möchten. Die Aufzählung erhebt nicht den Anspruch vollständig zu sein oder alle Probleme des RE in der Telekommunikationsbranche zu adressieren. Zum Beispiel werden „klassische“ Problemfelder wie die unterschiedliche Behandlung von funktionalen und nicht-funktionalen Anforderungen oder die Reibungsfläche zwischen agilen und planbasierten Vorgehensmodellen nicht diskutiert.

3 Problemstellungen

Die Telekommunikationsbranche ist eine vergleichsweise junge Branche, die sich ab Mitte des 20ten Jahrhunderts mit der Informationswirtschaft parallel entwickelt hat. Aktuell wird die Telekommunikationsbranche als Teil der Informationswirtschaft gesehen. Die Telekommunikationsbranche war in den vergangenen Jahren eine der Branchen mit einem enorm starken Wachstum. 2005 war das umsatzstärkste Jahr der Telekommunikationsbranche [StoJ]; derzeit findet eine Konsolidierung der Branche statt. Nationale Marktführer müssen sich auf ihrem Heimmarkt gegenüber internationalen Investoren behaupten. Die Marktanteile der Unternehmen der Telekommunikationsbranche verändern sich derzeit deutlich. Gleichzeitig verwischt die Grenze zwischen klassisch getrennten Dienstleistungen (z.B. Dienstleistungen von Fernsehsender und Telekommunikationsanbietern). Daher stoßen neue Anbieter in die Telekommunikationsbranche.

3.1 Vorhandene Prozesse im Unternehmen

Die Schlüsselunternehmen der Telekommunikationsbranche agieren mit ausgeprägten Ablauf- und Aufbauorganisationen. Die Ablauforganisation wird durch Prozessdefinitionen umgesetzt. Dies gilt für alle Unternehmensprozesse, auch für das Software Engineering und damit das Requirements Engineering. Diese Prozessdefinitionen entsprechen oftmals nicht den veröffentlichten RE-Ansätzen (siehe Kapitel 2), sondern sind entweder von Grund auf selbst definiert oder lehnen sich nur an die Vorbilder an. (Ein Beispiel für ein angepasstes Vorgehensmodell für einen Spezialfall in der Telekommunikationsbranche kann in [RMG12] gefunden werden.)

Ein Unternehmen in der Telekommunikationsbranche steht daher im Konflikt: Entweder wird das RE umgesetzt, wie in den Referenzwerken vorgegeben oder wie es Unternehmensvorgaben implizieren. In der Regel wird die zweite Option gewählt, vorwiegend aus zwei Gründen:

- „Not-invented-here“-Syndrom: Konzepte, in diesem Fall Prozesse von extern, werden nicht übernommen um organisationsinternen Konzepten den Vorrang zu geben. Dies kann durchaus gerechtfertigt sein: Das RE ist explizit branchenneutral, eine regulierte Branche wie die Telekommunikationsbranche muss jedoch andere Rahmenbedingungen erfüllen als andere Branchen. Die Folge: Entweder werden RE-Prozesse branchenspezifisch adaptiert oder es werden gar keine branchenneutralen Prozesse verwendet. Mit der enhanced Telecom Operation Map (eTOM) besteht ein Prozessrahmenwerk für die Telekommunikationsindustrie, welches die speziellen Anforderungen der Telekommunikationsbranche erfüllt [tm13] – aber keinen dezidierten RE-Prozess enthält.
- Haupt- vs. Teil- vs. Parallelprozess: Bei der Auswahl der Prozesse richten sich die Telekommunikationsunternehmen nach fachlich getriebenen Einteilungen im Produktlebenszyklus. Typische Prozesse sind dann z.B. Marktanalyse, Machbarkeitsstudie, Produktentwicklung, Produktlaunch, Produktbetreuung (relaunch etc.) oder Produktbeendigung. Hierin kann als Teilprozess RE stattfinden – es wird aber nicht die Notwendigkeit gesehen, einen übergreifenden, parallel laufenden RE-Prozess einzuführen.

Dabei wird die Chance nicht genutzt, aus den Erfahrungen der Wissenschaft in Gänze zu partizipieren. Probleme in der Prozessumsetzung sind daher u.U. sehr unternehmensbezogen (siehe auch Kapitel 3.2).

3.2 Mangelnde Toolunterstützung

Es existiert eine Vielzahl von Tools für das Requirements Engineering, angefangen von kommerziellen Werkzeugen über Open-Source-Software bis zu Modellierungstools mit RE-Komponente. Trotzdem ist keines dieser Werkzeuge tatsächlich im RE in der Telekommunikationsbranche etabliert. Immer wieder ergeben Umfragen, dass eine bessere Toolunterstützung gewünscht würde, das Standardwerkzeug aber praktisch Office-

Produkte sind. Aktuelle Umfrageergebnisse sind in [HO12], [STP12] und [SQ12] verfügbar.

Das Thema „Toolunterstützung“ lässt sich jedoch auch umdeuten: Ein Ansatz ist, RE toolneutral zu betrachten und den Schwerpunkt des RE wieder auf die Kommunikation zwischen Stakeholder und Requirements Engineer zu legen. Die Bedeutung des Tools würde dadurch abnehmen, die Umsetzung über Office-Produkte nicht nur organisatorisch einfacher, sondern sogar gewünscht (da weniger Overhead).

3.3 Enterprise Mobility als zentraler Faktor für das Anforderungsmanagement

„Enterprise Mobility“ bezeichnet das „Mobil-Werden“ der Unternehmen in den Dimensionen Personal, Prozessen und der eingesetzten Technologien. Die Themen innerhalb der einzelnen Dimensionen reichen von einem neuen Umfang der IT-Sicherheit über ein geändertes Applikationsmanagement bis hin zu vertraglichen Veränderungen und finanziellen Herausforderungen bei der Finanzierung und Nutzung mobiler Geräte. Die starke Verzahnung der Anforderungen zwischen den Dimensionen und den Themen erfordert eine erhöhte Aufmerksamkeit auf Verfolgbarkeit, Vererbung und Wiederverwendbarkeit von erhobenen Anforderungen. Die Unternehmen der Telekommunikationsbranche sind zweifach von Enterprise Mobility betroffen: Einerseits sind sie Treiber und Enabler der notwendigen Anforderungen und Technologien (d.h. zu ihren Kunden hin), andererseits müssen sie ihre eigene Organisation in Richtung mobile Anwendungen entwickeln. Für das RE bedeutet Enterprise Mobility, dass der Requirements Engineer noch stärker in räumlich verteilten Teams und Stakeholder denken muss. Es liegt nahe, dass sich für Anwendungen der Enterprise Mobility Anforderungsmuster ergeben, wie z.B. in [KC02] für Anwendungen von Embedded Systems dokumentiert sind. Im Fazit dieses Artikels wird auf diesen Aspekt eingegangen.

4 Lösungsansatz: REflex

4.1 Bausteine des REflex-Rahmenwerks

REflex baut auf den Aktivitäten des RE auf und konkretisiert diese aus operativer Sicht. Es wird bewusst Bezug auf die Prozesse in den Referenzwerken genommen. REflex ist nicht als Ersatz der Rahmenwerke in [Po08], [Ru09], sondern als ergänzende Hilfe für die Umsetzung zu verstehen.

In [SV10] wird „Pragmatisches IT-Projektmanagement“ (PITPM) beschrieben. Hiermit wird eine Verbindung zwischen dem formalen Rahmen des Guide to the Project Management Body of Knowledge („PMBok Guide“) [PM13] und praxisrelevanten Aspekten und Elementen der Softwareentwicklung beschrieben. Gleichzeitig erfolgt eine Anpassung an den deutschsprachigen Kulturraum. Dokumentvorlagen („Artefakte“) liefert PITPM über eine entsprechende Website mit. Ohne zu tief in PITPM einzusteigen – dies kann ein sinnvolles Vorbild sein, um das RE auf die Rahmenbedingungen und Anforderungen der Telekommunikationsbranche anzupassen.

Die Bausteine von REflex sind in Kürze beschrieben:

1. Rahmen: Der Rahmen des Projektes wird innerhalb der drei Aktivitäten ‚Bedarf formulieren‘, ‚Scope ermitteln‘ und ‚Einflussfaktoren analysieren‘ festgelegt.
2. Quellen: Die Quellen der Anforderungen werden identifiziert, d.h. wer oder was die Anforderungen vorgibt.
3. Ermitteln: Durch die Aktivitäten ‚Ermittlungstechniken festlegen‘ und ‚Anforderungen ermitteln‘ werden die Anforderungen identifiziert.
4. Dokumentieren: Die zuvor ermittelten Anforderungen werden dokumentiert. (Aktivitäten ‚Kategorien festlegen‘, ‚Dokumentationsrichtlinien festlegen‘, ‚Qualitätskriterien festlegen‘, Anforderungen spezifizieren und modellieren‘ und ‚Anforderungen prüfen‘).
5. Managen wird als eine phasenübergreifende, unterstützende Querschnittsaktivität definiert, die parallel alle vorherigen Phasen begleitet und die Aktivitäten ‚Kommunikation‘, ‚Änderungsmanagement‘, ‚Prozess sicherstellen‘, ‚nächste Phase planen‘ sowie ‚Akzeptanz / RE-Motivation‘ beinhaltet.

REflex lehnt sich an das bekannte Spiralmodell an. Die Bausteine werden iterativ-inkrementell durchlaufen. Managen ist dabei als Querschnittstätigkeit zu verstehen. In der Regel wird mit der Klärung des fachlichen Verständnisses begonnen, die Kundenanforderungen werden in der zweiten Iteration bearbeitet. In der folgenden Iteration werden Systemanforderungen behandelt. REflex trifft keine verbindliche Aussage darüber, welche Anforderungsarten in welcher Reihenfolge bearbeitet werden.

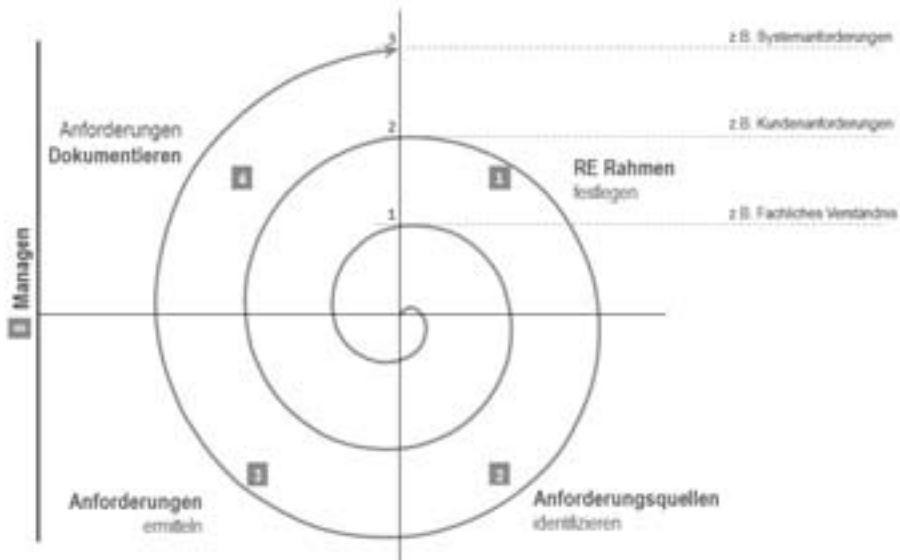


Abbildung 1: REflex-Bausteine, dargestellt als Spiralmodell

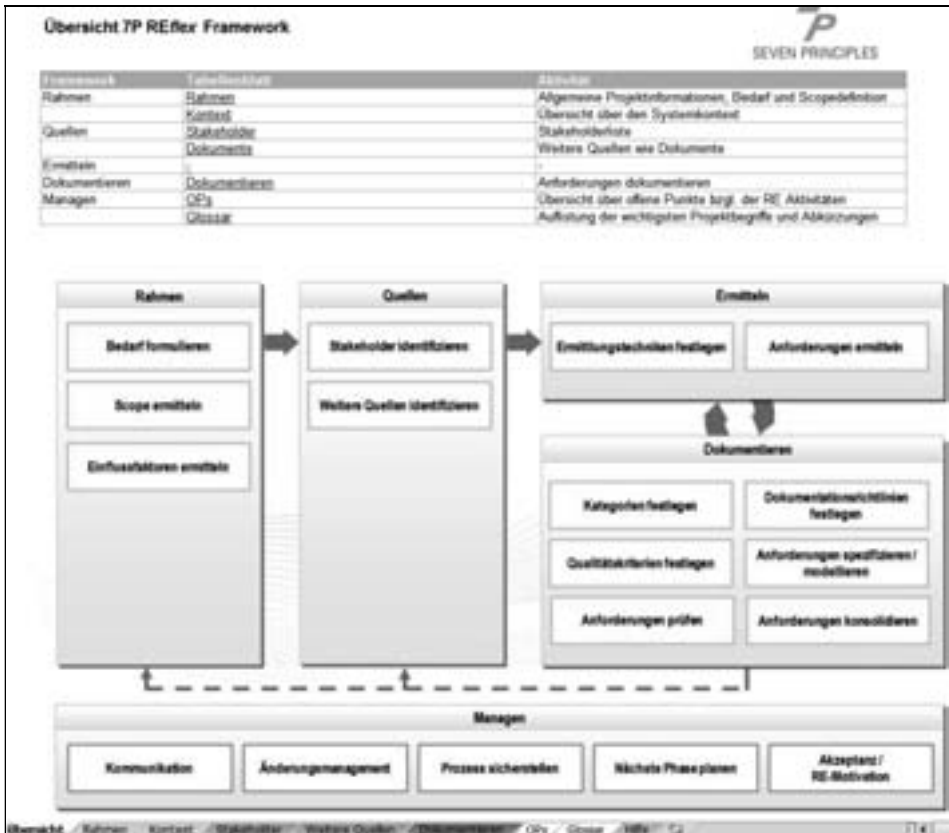
4.2 Umsetzung des REflex-Rahmenwerks in Office-Tools

In Projekten in der Telekommunikationsbranche sind i.d.R. bereits Softwaretools zur Softwareentwicklung und für das Projektmanagement vorhanden. Die Etablierung eines weiteren Tools ist meist nicht gewünscht. Daher ist REflex bewusst toolneutral gehalten. Gleichzeitig ist die Dokumentation von Anforderungen eine Kerntätigkeit des RE. Eine Anbindung an eine Software ist daher unbedingt notwendig. Es liegt daher nahe, die gängigen Office-Lösungen als Tools für das RE zu nutzen. Dies gilt aus Sicht der Autoren trotz typischer Nachteile von niedrighschwelligen Tools: Geringe Verfolgbarkeit, Vererbung, und Identifikation der Wiederverwendung. Eine Tabellenkalkulation ist ein Anfang für kleine und mittlere Projekte und kann von anderen Tools leicht importiert werden.

In Abbildung 2 ist der Screenshot REflex -Datei zu sehen. Es handelt sich um eine gängige Tabellenkalkulationssoftware. Die Datei enthält mehrere Reiter. Der aufgeschlagene Reiter bietet einen Überblick mit einer Darstellung von REflex. Den Bausteinen (wie oben vorgestellt) sind Aktivitäten zugeordnet, die in dieser REflex -Datei umgesetzt werden. Nicht jeder Baustein wird in der Datei berücksichtigt (z.B. gibt es bewusst keinen Reiter für die Ermittlung), sondern nur diese Aktivitäten werden benannt, die tatsächlich mit einer Tabellenkalkulation sinnvoll umzusetzen sind. Es bleibt dem Nutzer freigestellt, ob funktionale oder nicht-funktionale Anforderungen dokumentiert werden oder eine andere Einteilung zu nutzen.

Die Umsetzung in einem Office-Tool ist für ein kollaboratives Arbeiten nur eingeschränkt nutzbar. Die Nutzung von Änderungsnachverfolgungen ist derzeit nach Erfahrung der Autoren nur über eine begrenzte Anzahl von Versionen angenehm: zu schnell nimmt die Größe eines Dokuments zu, wenn z.B. mehrere Versionen mehrerer Bilder in einem Textdokument gespeichert werden müssen. Insofern wird hieraus eine Tugend gemacht: Es obliegt dem Requirements Engineer Änderungen in der Datei zu führen. Bezüglich widersprüchlichen Anforderungen muss vom Requirements Engineer ein Konsens gefunden und in die Datei überführt werden – eine gesonderte Handhabung im Tool ist nicht vorhanden. (Die Autoren behandeln Konsensfindung im RE in [Ra12].)

Diese Datei ist nur eine von mehreren Vorlagen. Eine komplette Übersicht ist hier aus Platzgründen leider nicht möglich.



5 Generalisierung der Erfahrungen: Bildung von Umsetzungshandreichungen für Vorgehensmodelle

Aus wissenschaftlicher Sicht ist REflex als weitere Abstraktionsschicht zwischen den etablierten Rahmenwerken und der Praxis des Requirements Engineering zu verstehen. Ein sinnvoller Begriff für diese Zwischenschicht wäre „Umsetzungshandreichung“.

Das Ziel von REflex ist daher nicht, die bestehenden Ansätze zu verändern oder zu ersetzen. Dies ist analog zum Einsatz von PITPM. Aus Sicht der Autoren ist es wahrscheinlich, dass sich solche Zwischenschichten immer dann ausbilden, wenn die Rahmenwerke aufgrund ihrer Komplexität nicht umgesetzt werden (können).

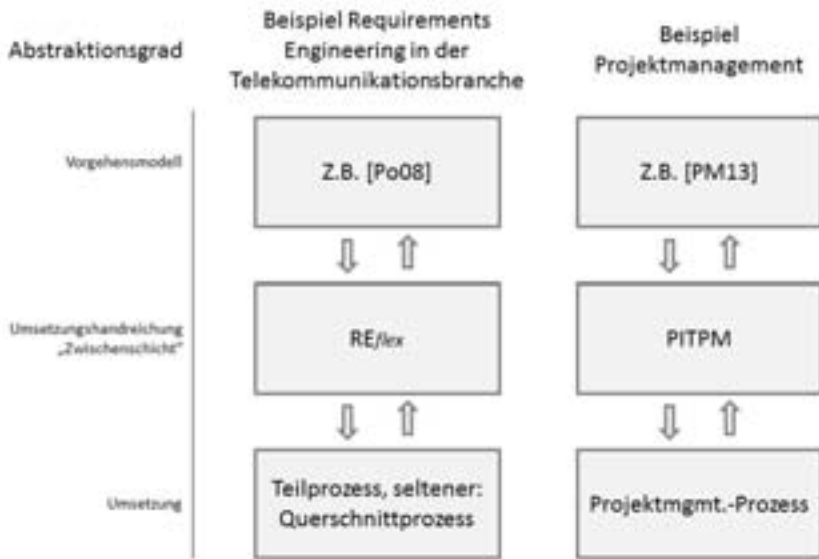


Abbildung 3: Rahmenwerk, Umsetzungshandreichung und Umsetzung in Schichtenarchitektur

Zum Konzept der Zwischenschicht gehört aus Sicht der Autoren, dass die Rahmenwerke noch stärker in praxisrelevante Komponenten unterteilt werden. Die Schnittstellen der Komponenten müssen sich in erster Linie an deren Einsetzbarkeit in der Praxis ausrichten. Ein Vorgehen hierzu wäre z.B.

1. Aufgliederung der methodenzentrierte RE-Prozesse (Ermitteln, Dokumentieren, Prüfen und Übereinstimmen) in feingranulare Aktivitäten (wie in *RE/lex* bereits geschehen)
2. Zuordnung der feingranularen Aktivitäten zu fachlichen Prozessen (wie in Kapitel 3.1 beschrieben z.B. Marktanalyse, Machbarkeitsstudie und Produktentwicklung)
Ausgewählte Fragen hierzu sind:

- a. Wie werden Anforderungen an neue Produkte in dem sich wandelnden Telekommunikationsmarkt verlässlich erhoben? Eine Analyse für einen Spezialfall aus wissenschaftlicher Sicht wird in [BK11] vorgestellt.
- b. Welche Modellierungssprachen sind geeignet, um die Anforderungen branchengerecht zu dokumentieren? Ein Ansatz aus wissenschaftlicher Sicht ist in [PH11] dokumentiert.

Die Überführung der Ergebnisse aus der Wissenschaft in die Praxis steht derzeit aus.

6 Fazit und Ausblick

6.1 Fazit

Dieser Erfahrungsbericht dokumentiert den Stand der Umsetzung des Requirements Engineering in der Telekommunikationsbranche, insbesondere

- Schwierigkeiten in der Integration des RE in die Ablauf- und Aufbauorganisation und
- Umsetzung von RE-Aktivitäten in Office-Tools.

Beide Punkte beziehen sich auf das Vorgehen des RE. Das dritte Thema Enterprise Mobility ist dagegen in erster Linie keine Frage des Vorgehens. Unter Enterprise Mobility werden Anwendungen zur Unterstützung der mobilen Aktivität von Unternehmen verstanden. Es ist wahrscheinlich, dass Enterprise Mobility bestimmte Anforderungsmuster impliziert. Welche dies sind, sollte Gegenstand weiterer angewandter Forschung sein.

REflex wurde als neutrale Umsetzungshandreichung des RE vorgestellt. Toolneutral bedeutet dabei, dass kein spezielles RE-Tool benötigt wird um REflex umzusetzen, sondern mit weit verbreiteten Office-Anwendungen Anforderungen dokumentiert werden. Gleichzeitig ist REflex in Tools integrierbar, wie z.B. in [En13] gezeigt wird.

6.2. Ausblick

Wie schon in der Einleitung geschrieben, beziehen sich diese Erfahrungen auf die Telekommunikationsbranche. Ob ähnliche Erfahrungen auch in anderen Branchen gemacht werden können, sollte Gegenstand anderer Untersuchungen sein.

In der Automobilindustrie ist das RE bereits als Disziplin etabliert. Entsprechende Fachgruppen in Praxis und Wissenschaft bestehen. Eine Selbstorganisation der Requirements Engineers in der Telekommunikationsbranche sollte ein nächster Schritt sein. In einer solchen Gruppe lassen sich die oben beschriebenen Erfahrungen weiter generalisieren.

Die oben angesprochene weitere Aufgliederung in Komponenten in der Zwischenschicht sollte weitergetrieben werden. Eine Kooperation von Wissenschaft und Praxis ist hierfür notwendig. Weitere Untersuchungsgebiete könnten der branchenspezifische Umgang mit funktionale und nicht-funktionalen Anforderungen, Agilität und Refactoring/Ablösung von Altsystemen sein.

Literaturverzeichnis

- [BK11] Basole, R.; Karla, J.: Entwicklung von Mobile-Platform-Ecosystem-Strukturen und – Strategien. In: Wirtschaftsinformatik. 5.2011; S. 301-311.
- [En13] Engel, F.: 7P-Reflex mit Toolunterstützung durch in-Step. Blogeintrag vom 11. April 2013, abrufbar unter <http://microtool.de/blog/post/7P-REflex-mit-Toolunterstuetzung-durch-in-Step.aspx>. 2013.
- [HO12] HOOD Group: RE-Kompass. 2012.
- [Ho12] Houdek, F.: Improving Requirements Engineering Processes - Impressions during One Decade of Improvement at Daimler. In: Dieste, O., Jedlitschka, A., Juzgado, N.J. (Hrsg.): Product-Focused Software Process Improvement. Lecture Notes in Computer Science 7343 Springer 2012; S. 1-2
- [KC02] Konrad, S.; Cheng, B.: Requirements Patterns for Embedded Systems. In: Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE02), Essen, Germany, September 2002; S. 127-136.
- [NE00] Nuseibeh, B.; Easterbrook, S.: Requirements Engineering: A Roadmap. In: Proceedings of the Conference on the Future of Software Engineering. ACM. 2000; S. 35-46.
- [PH11] Pousttchi, K.; Hufenbach, Y.: Wertschöpfung im Mobilfunkmarkt. In: Wirtschaftsinformatik. 5.2011; S. 287-300.
- [PM13] Project Management Institute: A Guide to the Project Management Body of Knowledge. 2013. Weitere Ressourcen sind unter <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx> verfügbar.
- [Po08] Pohl, K.: Requirements Engineering. Grundlagen, Prinzipien, Techniken. dpunkt, 2008.
- [Ra12] Rachmann, A.: Achieving Consensus in Requirements Engineering from the Viewpoint of Disourse Ethics. In: Proceedings of the RePriCo-Workshop as a part of the 18th International Working Conference on Requirements Engineering: Foundation for Software Quality (RefSQ 2012); S. 153-162.
- [RMG12] Rachmann, A.; Maucher, I.; Gök, M.: „Der Einbezug von Nutzungsversprechen und Requirements Engineering in die Entwicklung von AAL-Systemen“. In: 4. Deutscher AAL-Kongress „Innovative Assistenzsysteme im Dienste des Menschen – Von der Forschung für den Markt“. 2012.
- [Ru09] Rupp, C. und die Sophisten: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. Fünfte Auflage, Carl Hanser, 2009.
- [SQ12] SwissQ: Requirements Trends & Benchmarks. 2012.
- [StoJ] Statista. Daten & Fakten zur Telekommunikationsbranche. <http://de.statista.com/statistik/faktenbuch/359/a/branche-industrie-markt/it-telekommunikation/telekommunikationsbranche/>. Ohne Jahresangabe.
- [STP12] Sikora, E.; Tenbergen, B.; Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. In: Requirements Engineering 17, 2012; S. 57-78.
- [SV10] Spitzcok von Brisinski, N.; Vollmer, G.: Pragmatisches IT-Projektmanagement: Softwareentwicklungsprojekte auf Basis des PMBOK® Guide führen. dpunkt, 2010.
- [tm13] tmforum: Business Process Framework. Abrufbar unter <http://www.tmforum.org/BusinessProcessFramework/1647/home.html>. Abgerufen am 2.8.2013.
- [WW03] Weber, M.; Weisbrod, J.: „Requirements engineering in automotive development: experiences and challenges“ Software 20.1, 2003; S. 16-24.

Das „agile“ Vorgehen: Neuer Wein in alte Schläuche - oder ein „Déjà-vu“?

Hans-Peter Korn

KORN AG
Turnweg 13
CH 5507 Mellingen
contact@korn.ch

Abstract: Nach dem Hinterfragen essentieller Hintergründe des agilen Vorgehens (Planbarkeit, Komplexität, Selbstorganisation, Kommunikation, Vertrauen) werden die unterschiedlichen Sichtweisen von „agil“ dargestellt und heute verbreitete agile Konzepte und Praktiken und deren historische Wurzeln zum Entwickeln von Software, für das Management von Projekten, zur Neu- und Weiterentwicklung und Wartung von Produkten und die insgesamt agile Organisation diskutiert.

1 Einleitung

Agil zu sein ist unbestritten attraktiv: Wer will denn nicht - gemäß dem Duden-Fremdwörterbuch - „behände, flink, gewandt, regsam, geschäftig“ sein? Wer will als das Gegenteil von agil im Sinn seines lateinischen Ursprungs (agilis, abgeleitet von agere) gesehen werden, als nicht handelnd oder unbeweglich? Langsamkeit und Müßiggang (als sprichwörtlicher Anfang aller Laster) sind spätestens seit Martin Luther verpönt. Er schrieb: „*Von Arbeit stirbt kein Mensch, aber von Ledig- und Müßiggehen kommen die Leute um Leib und Leben; denn der Mensch ist zum Arbeiten geboren wie der Vogel zum Fliegen.*“

Wieso aber, obwohl „behände, flink, gewandt, regsam, geschäftig“ immer schon erstrebenswert war, ist Agilität in den letzten rund zehn Jahren zu einem Modebegriff geworden? Was macht „Agilität“ heute noch viel mehr als früher so attraktiv - oder sogar unabdingbar?

Einige Hinweise sind im seit 2009 vom amerikanischen Deloitte's Center for The Edge jährlich herausgegebenen Shift Index nachzulesen [Ha11]. Im Shift Index wird seit 1965 bis heute die Performance von 20.000 US-Unternehmen analysiert.

Der Return on Assets (ROA) sinkt seit Jahrzehnten kontinuierlich und ist heute nur noch knapp über null. Die Volatilität der Aktienkurse nimmt zu. Die topple rate als Maß für den Verlust der Marktführerschaft großer Firmen hat sich mehr als verdoppelt. Die Treue der Kunden ist einer hohen Wahlbereitschaft gewichen. Produktinnovationen sind nicht mehr das Privileg hoch entwickelter Ökonomien, sondern werden mit immer rasan-

terem Tempo im asiatischen und pazifischen Raum entwickelt, zwischen 1995 und 2006 haben die Wechsel der Chief Executive Officers (CEO) um 59% zugenommen, die durch mangelnde Performance begründeten sogar um 318%.

Und ein Drittel der 1970 in Fortune gelisteten 500 umsatzstärksten Unternehmen der Welt existierte 1983 nicht mehr. Von den 1917 von Forbes genannten 100 größten Firmen existierte 2001 nur noch eine, nämlich General Electric.

Deloitte's „Shift Index 2011“ fasst das so zusammen: *„Surveying today's business landscape, perhaps investors intuitively grasp that "normal" is a thing of the past—that we have entered a world that does not stabilize as easily as it once might have.“*

Das „agile“ Vorgehen wird heute - losgelöst von der SW-Entwicklung als solche - als Antwort auf diese Herausforderungen gesehen. Dieser Beitrag beschreibt die unterschiedlichen Sichtweisen dessen, was heute unter „agil“ verstanden wird und widmet sich dann typischen Konzepten und Praktiken des agilen Vorgehens. Dabei wird gezeigt, dass Vieles davon bereits seit Jahrzehnten bekannt ist und auch - allerdings vergleichsweise selten - praktiziert wird.

Im Abschnitt *„Agiles“ Vorgehen als Lösung?* beschäftigt sich der Beitrag zunächst mit einigen essentiellen Hintergründen der Agilität (Planbarkeit, Komplexität, Selbstorganisation, Kommunikation, Vertrauen) und beschreibt danach in vier Gruppen die unterschiedlichen Sichtweisen von „agil“:

- Agil = Flexibilität und Adaptivität mittels „Empirical Process Control“
- Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“
- Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ verbunden mit einer Sammlung „post-tayloristischer“, hierarchie- und autoritätsfreier, partizipativer, selbststeuernd-kollaborativer, kommunikationsbasierter, systemischer Praktiken und „Glaubenssätze“
- Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit unabhängig vom Empirical Process Control

Im Abschnitt *Heute verbreitete agile Konzepte und Praktiken* werden Konzepte und Praktiken

- des agilen Entwickelns von Software,
- des agilen Managements von Projekten,
- der agilen Neu- und Weiterentwicklung und Wartung von Produkten

und als Abschluss die insgesamt agile Organisation vorgestellt.

2 Agiles Vorgehen als Lösung?

Wie geht das agile Vorgehen mit den in der Einleitung dargestellten und von der reinen SW-Entwicklung unabhängigen Herausforderungen um?

Im Kern beruht es - in der einen Sichtweise - auf einer im Gegensatz zur uns vertrauten tiefgehend analysierenden, dann in Detail planenden und schlussendlich plangemäß handelnden Herangehensweise auf einem empirischen, betont kommunikations- und kooperationsorientierten und vertrauensbasierten Arbeiten. Es geht dabei darum, die von uns wahrgenommene Komplexität (im Sinn einer grossen Unsicherheit bezüglich der Vorhersehbarkeit zukünftiger Situationen und Ereignisse) zu akzeptieren und mit ihr konstruktiv umzugehen statt sie beherrschen zu wollen.

In einer anderen Sichtweise geht es dabei darum „Selbstorganisation“ zu verstehen als Ergebnis der fortlaufenden, auch das eigene Handeln als Individuum und Team reflektierende, Kommunikation; und darum, Vertrauen als wesentliche Essenz funktionsfähiger sozialer Systeme zu erkennen und aufzubauen.

Vor diesem Hintergrund werden dann die verschiedenen Ausprägungen von „agil“ beschrieben:

- Flexibilität und Adaptivität mittels „Empirical Process Control“ stellt die kurzen Iterationen verbunden mit einem fortlaufenden Anpassen und Lernen ins Zentrum und reicht in der SW-Entwicklung zurück in die 1970er-Jahre.
- Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ ergänzt diese Sichtweise um die vom „Toyota Way“ begründeten Prinzipien des Lean Management.
- Eine weitere Ausprägung umfasst zusätzlich noch etliche Praktiken und „Glaubenssätze“ der Organisationsgestaltung der letzten Jahrzehnte geprägt von post-tayloristischen, hierarchie- und autoritätsfreien, partizipativen, selbststeuernd-kollaborativen und systemischen Konzepten.
- Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit bilden eine weitere Gruppe jener Ausprägungen von „agil“ die unabhängig vom „Empirical Process Control“ sozialen Systemen i.a. dienen können.

Diese Auslegeordnung zeigt, dass der Begriff „agil“ so vieldeutig ist, dass er stets einer zum jeweiligen Kontext passenden Präzisierung bedarf. Eigentlich sollte daher auf den Gebrauch dieses Begriffes verzichtet werden und statt dessen von dem gesprochen werden, was jeweils konkret gemeint ist.

2.1 Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen

Die unbegrenzt erscheinende Menge der sich rasch verändernden und ständig wachsenden Informationen und die uns heute spontan mögliche weltumspannende Vernetzung

nehmen wir als extrem komplex wahr. In immer mehr bislang einigermaßen voraussehbaren Lebensbereichen werden Planungen zunehmend zu unsicheren Prognosen oder gar nur zu Hoffnungen.

Es ist dann wie beim Überqueren eines frisch zugefrorenen und unbekannten Flusses, um die schemenhaft im Nebel am gegenüberliegenden Ufer erkennbare einfache, aber - vermutlich - gut beheizte Hütte zu erreichen.

Die eine Vorgehensweise ist:

Vorsichtig, Schritt für Schritt, bei jedem Knistern die Richtung ändernd, versuchen wir die Überquerung. Und etwa in der Mitte lichtet sich der Nebel etwas und einige hundert Meter neben der Hütte erscheint ein sicher viel angenehmeres Hotel. Und, vorsichtig, Schritt für Schritt, ändern wir unsere Richtung in Richtung Hotel.

Eine andere Vorgehensweise wäre:

Zunächst mit speziellen Wärmebild-Technologien die Beheizung der Hütte überprüfen. Wenn sie geeignet erscheint, wird sie als verbindliches Ziel definiert. Dann wird anhand der Klimawerte der letzten Wochen und der dokumentierten Erfahrungen der letzten Jahre die Eisdicke berechnet und mit Würfeln von Steinen bekannten Gewichts und berechneter Ballistik stichprobenartig verifiziert. Dann wird der beste Weg über das Eis zur Hütte geplant. Und dann wird dieser Weg, ausgerüstet mit Schwimmweste und einer langen Leiter, gemäß Plan zügig und nach Plan bewältigt.

Diese zweite Vorgehensweise entspricht einem von der technologischen Analysierbarkeit, Machbarkeit und Planbarkeit bestimmten Denken und ist Basis aller „plangetriebenen“ Vorgehensmethoden der Produktentwicklung und des Projektmanagements auf der Grundlage eines möglichst frühen Big Design Up Front (BDUF).

Die erste Vorgehensweise hingegen entspricht den Erkenntnissen im Umgang mit Complex Adaptive Systems, wie sie u. a. von Edwin E. Olson und Glenda H. Eoyang beschrieben werden [OEG01]. Geprägt ist diese Vorgehensweise von der Einsicht, dass uns Situationen als komplex dann erscheinen, wenn wir Beziehungen zwischen Ursache und Wirkung nur im Nachhinein erkennen können, nicht aber im Voraus. Unser Vorgehen beruht dann auf Handeln-Beobachten-Reagieren und emergenten Praktiken. Im Gegensatz dazu stehen die uns kompliziert erscheinenden Situationen mit - wenn auch sehr aufwendig - analysierbaren und voraussagbaren Ursache-Wirkungs-Beziehungen oder den von uns als simpel betrachteten Situationen mit für alle offensichtlich erscheinenden Ursache-Wirkungs-Beziehungen [KS03].

Bei dieser ersten - empirischen - Vorgehensweise verzichten wir bewusst auf eine unangemessen aufwendiges und noch dazu unzuverlässiges BDUF. Stattdessen planen wir nur jeweils das für die nächsten Schritte wirklich Erforderliche auf Basis der bisherigen Einsichten. Zu Beginn unseres Vorhabens gehen wir von einem eher groben Just Enough Design Up Front (JEDUF) aus.

Dieser Wechsel vom BDUF mit seiner Paukenschlag-Einführung (big bang) der kompletten Lösung zum JEDUF mit seinen häufigen Einführungen von aufeinander aufbauenden Lösungsteilen (Lösungsinkrementen) und den die weitere Entwicklung steuernden raschen Feedbacks zeigt sich exemplarisch in der Softwareentwicklung (als wesentlichem Treiber des agilen Vorgehens) in den letzten 30 Jahren, wie sie etwa von Brian Wernham im Kapitel 19 „The Lure of Big Design Up Front“ seines Buches Agile Project Management For Government [We12] gut beschrieben ist.

James Martin prägte ab dem Ende der 1980er-Jahre mit seinen Büchern [Ma90] die von sequenziellen und jeweils etliche Monate dauernden Phasen (Analyse, Konzept, Realisierung, Einführung) bestimmten „wasserfallartigen“, betont arbeitsteiligen und stark werkzeugunterstützten Vorgehensweisen. Allerdings wies bereits 1970 Winston W. Royce im Artikel „Management of Large Software Systems“ in Proceedings IEEE WESCON August 1970 pp. 1 - 9 darauf hin, dass das in seinem Artikel in Fig.2 dargestellte „wasserfallartige“ Vorgehen ungeeignet sei und empfiehlt statt dessen in Fig.10 ein iteratives Vorgehen mit sehr vielen Rückkopplungsschleifen unter Einbezug auch des Benutzers. Dennoch haben sich seit den 1980er-Jahren in Anlehnung an die Prozesse zur Fertigung physischer Produkte Vorgehensweisen auf Basis streng getrennter sequentieller Phasen i.S. von Analyse, Konzept, Realisierung, Einführung etabliert und gipfelten in der Idee der „Software Factory“, also Software unter Außerachtlassen des fundamentalen Unterschieds zwischen „Design“ und „Make“ auf Basis industrieller Fertigungsprozesse zu entwickeln. [Po02]

Spätestens seit der Jahrtausendwende erleben wir jedoch einen immer deutlicheren Wechsel hin zu flexiblen, schlanken und stark teamorientierten Entwicklungsmethoden, die in jeweils wenigen Wochen verfügbare und konkret nutzbare Teilergebnisse liefern. Dabei wird im Bereich von Architektur und Design auf den intensiven Einsatz von Modellierungswerkzeugen verzichtet. Stattdessen erfolgt die Software-Realisierung (das Programmieren und Testen) zu einem möglichst hohen Grad toolunterstützt und - beim Testen - automatisiert.

Der Wechsel weg vom ab Beginn im Detail ingenieurmäßig analysierten, modellierten und durchgeplanten Vorgehen hin zum nur grob im Voraus abgesteckten und danach schrittweise angepassten inkrementell-adaptiven Vorgehen ist natürlich auch eine psychologische Herausforderung: In der Regel ist es vielen von uns wohlher, wenn wir einem detaillierten Plan folgen können [Do12].

2.2 „Selbstorganisation“ als Lösung - oder als Missverständnis?

„Selbstorganisation“ gilt schlechthin als „die“ Alternative zu einer von „command and control“ geprägten, stark arbeitsteiligen, unflexiblen und hierarchischen „Organisation 0.0“ [Kr09]. Beim Hinterfragen der Bedeutung von „Selbstorganisation“ jedoch stellt sich diese bald als recht buntes Gemisch aus abstrakten Prinzipien der Systemtheorie, aus chaotisch und eigendynamisch gesehenen Naturprozessen, aus Evolutionsphänomenen bei Organismen, aus Mustern der Emergenz sozialer Strukturen, aus Übertragungen neoliberaler Mechanismen des freien Markts auf die Organisationseinheiten im Unter-

nehmen bis hin zu basisdemokratischen und anarchischen Vorstellungen heraus. Also: „Selbstorganisation ist Chiffre für einen Erklärungsnotstand, keine Erklärung.“ [Tu02]

Die Forderung nach Selbstorganisation kann überdies zu einem leistungspolitischen „double bind“ führen: *„Einerseits gehören nunmehr Selbstkoordination und kreative Problemlösung zum offiziellen Aufgabenkanon der Gruppe, andererseits fehlt Zeit und Personal, um diese Aufgaben angemessen erfüllen zu können.“* [Wo03] Andere Stimmen sehen in der Förderung der Selbstorganisation im Unternehmen ein Mittel der Selbstdisziplinierung der Mitarbeitenden im Interesse der Profitmaximierung der Kapitalgeber und des Top-Managements: *„Management expects individuals in post-Fordist capitalism to be flexible, innovative, motivated, dynamic, modern, young, and agile, and it wants them to identify with the corporation and to have fun at work. Strategies of participatory management aim at the ideological integration of laborers into corporations. This is a new quality of the disciplinary regime that aims at a rise of profits by an increase in productivity and cost reductions achieved by the workers' permanent self-discipline“* [Fu08]

2.3 Kommunikation und Vertrauen bilden soziale Systeme

Auf einem festeren Boden stehen wir, wenn wir den Begriff „Selbstorganisation“ vermeiden und uns statt dessen auf die Erkenntnisse der Theorie komplexer adaptiver Systeme, insbesondere sozialer Systeme, abstützen. Auf diesem Feld gewinnt dann ein anderer Begriff weitaus größere Bedeutung als jener der Selbstorganisation, nämlich „Kommunikation“. Gemäß Niklas Luhmann bildet allein Kommunikation - und nicht Personen - Systeme. Kommunikation versteht er als die kleinste Einheit eines Systems, wobei nicht Menschen (oder psychische Systeme) kommunizieren sondern Kommunikation kommuniziert und nur in sozialen Systemen stattfindet. [Lu84]. Ein großes Unternehmen stellt somit ein sehr komplexes Gebilde „kommunizierender Kommunikation“ dar. Damit wir Menschen diese umfassende Komplexität bewältigen können, so Luhmann, müssen wir sie reduzierend betrachten. Gemäss seiner Systemtheorie ist die Reduktion von Komplexität die Hauptaufgabe von sozialen Systemen (also auch von Organisationen). Nur so können wir Menschen überhaupt überleben. Ein wesentliches Element der Komplexitätsreduktion ist - gemäss Luhmann - das Vertrauen: *„Vertrauen ist stets in die Zukunft gerichtet. ... im Akt des Vertrauens (wird) die Komplexität der zukünftigen Welt reduziert... Vertrauen erschließt durch die Reduktion von Komplexität Handlungsmöglichkeiten, die ohne Vertrauen - nach Luhmann - unwahrscheinlich und unattraktiv geblieben und somit nicht zum Zuge gekommen wären“* [Ta09] [Lu00]

Ausgehend von Luhmann geht es also nicht so sehr darum, "besser" mit dieser sich durch den Verlust an Vertrauen verstärkt - oder gar als überfordernd - wahrgenommenen Komplexität so umzugehen, dass wir jederzeit flexibel und mit kurzfristig wechselnden Zielen in unserer Welt leben können sondern eher darum, wie wir wieder mehr Vertrauen zu Personen, Rollenträgern, Teams, Normen, Organisationen, Strukturen und Prozessen erlangen. Und Vertrauen erlangen wir durch direkte und persönliche Kommunikation und Kooperation.

Viel zu theoretisch und - was hat das mit „agil“ zu tun?

Sehr viel: Werfen wir doch einen Blick auf den ersten und dritten der vier „Werte“ des „Agile Manifest der Software-Entwicklung“: [Be01]

„... wir ((haben)) diese Werte zu schätzen gelernt:

- 1. Individuen und Interaktionen mehr als Prozesse und Werkzeuge*
- 2. Funktionierende Software mehr als umfassende Dokumentation*
- 3. Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung*
- 4. Reagieren auf Veränderung mehr als das Befolgen eines Plans*

Obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Und beachten wir auch diese drei seiner zwölf „Prinzipien“:

- *Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.*
- *Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteam zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.*
- *In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.*

All das bedingt und fördert Kommunikation. Und funktioniert nur auf Basis von „Vertrauen“ statt auf Basis einer alle Beteiligten überfordernden und noch dazu tragen granularen Steuerung via „Auftragserteilung von oben“.

Wenn auch all das unter „agil“ subsumiert wird - was genau bedeutet dann „agil“?

Und: Ist „agil“ im Rahmen der etablierter Führungsstrukturen und Vorgehensweisen nicht wie „neuer, noch gärender, Wein in alte Schläuche“? Oder ist es gar nicht so „neu“ - sondern ein „Déjà-vu“?

2.4 Die Vieldeutigkeit des Begriffes „agil“

AGIL erscheint in den 1950er-Jahren beim amerikanischen Soziologen Talcott Parsons als Akronym für vier überlebenswichtige Funktionen lebender und somit auch sozialer Systeme [Pa12]:

A: Anpassung (Adaptation) an die Umwelt

G: Zielerreichung (Goal Attainment)

I: Integration als Mechanismus zur Leistungserbringung der Teilsysteme untereinander

L: Strukturhaltung oder Latenz als Mechanismus zur Erhaltung der Identität des Systems, obwohl alles stetig im Wandel ist.

Im 1997 erschienenen Buch „Komplexität und Agilität: Steckt die Produktion in der Sackgasse?“ beleuchtet eine Reihe von Beiträgen „... . die Widersprüche zwischen Komplexität im Produktionsumfeld und Agilität im Markt.. . und [gibt] Perspektiven für deren Auflösung. Insbesondere wird den Fragen nachgegangen, was Management heute und in Zukunft bedeutet, wohin sich die Märkte bewegen werden und wie man ihnen folgt, wie die technischen Innovationen aussehen und was sie bewirken werden und wie sich die industrielle Produktion durch neue Technologien und Organisationsprinzipien ändern wird.“ [SW97]

Im Februar 2001 trafen sich 17 Vertreter der damals „leichtgewichtig“ genannten Methoden der Softwareentwicklung und formulierten das sie Verbindende als die vier Werte und zwölf Prinzipien des Agilen Manifests der Softwareentwicklung [Be01], auf das sich auch heute noch alle agilen Vorgehensweisen der Softwareentwicklung implizit oder explizit berufen.

Das war auch der Ausgangspunkt für die rasche Verbreitung des Begriffs „agil“ und seine Ausdehnung auf andere Bereiche außerhalb der SW-Entwicklung - und auch der Differenzierung seiner Bedeutung.

Heute können im Kontext der Produktentwicklung und des Projektmanagements, des Managements und der Organisationsgestaltung folgende vier Gruppen der Bedeutung von agil unterschieden werden:

2.4.1 Agil = Flexibilität und Adaptivität mittels „Empirical Process Control“:

Kennzeichnend dafür ist ein Vorgehen in kleinen, stets gleich langen Schritten, ausgehend von einem möglichst leichtgewichtigen Just Enough Design Up Front. Jeder Schritt liefert dabei ein - wenn immer möglich bereits praktisch nutzbares - Teilergebnis, das Grundlage für Feedbacks zur Gestaltung des nächsten Schritts ist. Das Vorgehen in jedem Schritt wird im Team reflektiert, um kontinuierliche Verbesserung, also Lernen, zu erreichen.

Für Tom Gilb, der bereits in den 1970er-Jahren für ein evolutionäres IT-Projektmanagement eintrat, besteht die zentrale Bedeutung von „agil“ in diesem inkrementell-adaptiven Vorgehen und kontinuierlichem Lernen. Und er meint, dass alle anderen agilen Taktiken optionale Details seien [Po02].

Diese Sicht entspricht auch der von Volker Nissen. Er unterscheidet dabei zwischen einer „kapazitiven Agilität“ als Fähigkeit der IT, auf schwankende mengenmäßige Anforderungen schnell antworten zu können (Skalierbarkeit, Performanz), und einer „funktionalen Agilität“, die sich auf die funktionalen Anforderungen bezieht. Bei vorhersehbaren Anforderungsänderungen spricht er von „reaktiver Agilität“. Wenn die IT jedoch eine aktive Rolle für das fachliche Geschäft übernimmt und unvorhergesehene Änderungen und IT- Innovationen aufspürt, spricht er von „proaktiver Agilität“ [NM09].

2.4.2 Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“:

Zusätzlich zur obigen Bedeutung werden hier die Prinzipien des Lean Management als eigentliche Basis von „agil“ gesehen, oft dargestellt als House of Lean, ausgehend von Larman und Vodde (2009) und The Toyota Way (2004) [Le09].

Das Ziel „Wert schaffen“ als Dach des House of Lean ist gekennzeichnet durch:

- Kurze Durchlaufzeiten
- Beste Qualität und höchsten Wert
- Höchste Kundenzufriedenheit
- Niedrigste Kosten
- Hohe Moral
- Sicherheit

und wird getragen von den zwei Säulen:

- Respekt für alle Personen
- Kontinuierliche Verbesserung

Erreicht wird das Ziel durch spezifische Entwicklungsmethoden, basierend auf 14 Lean Principles.

Dieser Begriff von agil wird vertreten auch in der Definition von Agilität gemäß Lexikon IT-Management [Lex10].

2.4.3 Agil = Flexibilität und Adaptivität (Empirical Process Control) plus „Lean Management Principles“ verbunden mit einer Sammlung „post-tayloristischer“, hierarchie- und autoritätsfreier, partizipativer, selbststeuernd-kollaborativer, kommunikationsbasierter, systemischer Praktiken und „Glaubenssätze“:

Dieser Begriff von agil erscheint auch im IT-unabhängigen Kontext des Managements und der Organisationsgestaltung. Das im Agile Enterprise Adaptation Program der Agile Alliance entwickelte Modell ist ein Beispiel für diesen Begriff von agil [Co12].

Auch das bereits weiter oben erwähnte Agile Manifest der Softwareentwicklung [Be01] kann dieser Gruppe zugerechnet werden. Seine vier Werte und die weiteren zwölf Prinzipien sind, dem Charakter eines Manifests entsprechend, jedoch als Appell zu verstehen und nicht als verifizierte Arbeitsregeln. Sie dogmatisch als wortwörtlich zu erfüllende Handlungsanweisungen für die Software- oder Produktentwicklung in unterschiedlichsten Situationen zu betrachten wäre eine Fehlnutzung [JS12].

2.4.4 Agil = Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit unabhängig vom Empirical Process Control:

Eine ganz andere, nicht auf die Software- und Produktentwicklung, sondern die Führung sozialer Systeme in risikoreichen und komplexen Missionen ausgerichtete Sicht von agil findet sich in „Power to the Edge“, einer Publikation des Command and Control Research Program des US Departement of Defense [AH09]:

- *Robustheit*: die Fähigkeit, aufgaben-, situations- und bedingungsübergreifend effektiv zu bleiben;
- *Belastbarkeit*: die Fähigkeit, sich von Unglücksfällen, Schäden oder einer destabilisierenden Störung der Umgebung zu erholen oder sich darauf einzustellen;
- *Reaktionsfähigkeit*: die Fähigkeit, auf eine Veränderung der Umgebung rechtzeitig zu reagieren;
- *Flexibilität*: die Fähigkeit, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen;
- *Innovationsfähigkeit*: die Fähigkeit, neue Dinge zu tun und die Fähigkeit, alte Dinge auf eine neue Art und Weise zu tun;
- *Anpassungsfähigkeit*: die Fähigkeit, Arbeitsprozesse zu ändern und die Fähigkeit, die Organisation zu ändern.

Diese Definition versteht unter Flexibilität im Gegensatz zu anderen Sichtweisen von agil nicht nur ein inkrementell-adaptives Vorgehen in kleinen Schritten, ausgehend von einem JEDUF, sondern lässt etwa auch ein stark plangetriebenes, BDUF-basiertes Vorgehen dann zu, wenn es für die aktuelle Situation passender und effizienter ist. Entscheidend dabei sind jedoch die Fähigkeiten, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen, auf eine Veränderung der Umgebung rechtzeitig zu reagieren und Arbeitsprozesse und die Organisation zu ändern.

Zu dieser Bedeutungsgruppe gehört auch das eingangs erwähnte Akronym „AGIL“ von Talcott Parsons.

3 Heute verbreitete agile Konzepte und Praktiken

Wenn heute von „agil“ gesprochen wird, wird es oft mit Scrum gleichgesetzt. Blicken wir jedoch rund ein Jahrzehnt zurück.

Damals, 2002, war in einem Artikel von Jens Coldewey das zu lesen: „*Noch immer wird agile Entwicklung in weiten Kreisen gleichgesetzt mit eXtreme Programming. Dadurch wird die Chance vergeben, auch von den anderen Verfahren zu lernen*“ [Co02]. Heute - rund ein Jahrzehnt später - wäre in diesem Artikel „eXtreme Programming“ durch „Scrum“ zu ersetzen. Und dadurch wird - auch heute - die Chance vergeben, von den anderen derzeit verbreiteten weiteren agilen Verfahren zu lernen.

Jens Coldewey behandelte 2002 im oben zitierten Artikel sechs agile Konzepte und Praktiken:

Drei Meta-Prozesse, wie das Team zu einem individuellen Prozess kommt:

- Adaptive Software Development,
- die Crystal-Methodenfamilie und
- Scrum.

und drei Konzepte, die recht konkrete Verfahren und Techniken, die dann im Laufe des Projekts angepasst werden können, vorschlagen:

- Dynamic Systems Development Method (DSDM),
- eXtreme Programming (XP) und
- Feature Driven Development (FDD).

Dominant im Jahr 2002 war XP. Heute, ein Jahrzehnt später, dominiert Scrum (allerdings zum größten Teil in diversen Abwandlungen die - gemäß „Scrum Guide“ [SS] - eigentlich nicht „Scrum“ genannt werden sollten) die agile Landschaft. Das zeigen übereinstimmend drei unabhängige Studien aus dem Jahr 2012 [Sw12] [KM12] [Ko12]

Gemäß diesen Studien setzten 2012 bei den antwortenden Firmen - je nach Studie - 51 bis 78 % agile Vorgehensweisen (oft parallel zu klassischen) ein. Davon nutzen 51 bis 84,5 % Scrum (bzw. ein an Scrum orientiertes Vorgehen); es steht damit in allen diesen Studien an erster Stelle. Bei zwei dieser drei Studien stand das vor zehn Jahren noch unbekannte Kanban mit 17 bzw. 43 % an zweiter Stelle, bei der dritten Studie mit 4,5 % an fünfter Stelle. Die Verbindung von Scrum mit Kanban (Scrumban) hatte eine Verbreitung von immerhin bereits 3 bis 8,5 %. Das vor zehn Jahren dominierende eXtreme Programming (XP) stand mit 5,5 bis 33 % Verbreitung in allen Studien nur noch an vierter Stelle, wobei allerdings viele der ursprünglichen XP-Praktiken (wie z. B. User Stories) heute als integraler Bestandteil von Scrum gesehen und praktiziert werden.

Weitere verbreitete - von den Befragten als „agil“ betrachtete - Vorgehensweisen waren 2012 u. a. IBM Rational Unified Process (RUP), Agile Unified Process, Open Unified Process, Feature Driven Development, Usability Driven Development und unterschiedliche hybride Vorgehensweisen.

Die vor rund zehn Jahren noch recht bekannte Vorgehensweise Adaptive Software Development und die Crystal-Methodenfamilie haben heute kaum noch und das Feature Driven Development (FDD) hat nur noch eine begrenzte Verbreitung und die Dynamic Systems Development Method (DSDM/DSDM Atern) ist nach wie vor in Großbritannien - auch bei Großprojekten von Behörden - sehr bekannt. Hingegen stoßen heute weitere etablierte Vorgehensweisen mit immer schon iterativen Elementen wie der Rational Unified Process (RUP) und PRINCE2 (Projects in Controlled Environments) in den Bereich der agilen Vorgehensweisen vor, indem sie ihre iterativen Elemente verstärken. Auch bislang an ausgeprägten sequenziellen Phasen orientierte klassische Vorgehens-

weisen wie das Schweizer HERMES und das V-Modell XT öffnen sich für eine Kombination mit inkrementell-adaptiven Vorgehensweisen zumindest im Sinn eines hybriden Vorgehens.

Interessant sind die in der Swiss Agile Study 2012 erhobenen Auswirkungen. Die Ergebnisse auf die Frage „How has agile software development influenced the following aspects?“ sind in [KM12] auf der Seite 27 dargestellt. Wenn man die Prozent-Werte für „much worse“, „worse“ und „unchanged“ zusammenzählt ergibt sich dieses Bild:

	% much worse + worse + unchanged	
Ability to manage changing priorities	10.0	
Development process	19.0	
Time to market	22.5	
Alignment between IT & business objectives	26.0	
Project visibility	26.5	
Team morale	29.5	
Requirements management	31.5	
Productivity	35.0	
Risk management	37.5	
Engineering discipline	45.5	
Management of distributed teams	46.5	28.5% Don't know
Software quality	47.0	
Software maintainability / extensibility capability	62.0	
Development Cost	65.2	

Demnach blieben aus Sicht der Mehrheit der Antwortenden bei der agilen SW-Entwicklung die Merkmale „Software maintainability / extensibility capability“ und „Development Cost“ unverändert oder haben sich sogar verschlechtert oder sehr verschlechtert. Deutlich verbessert haben sich hingegen die Merkmale: Ability to manage changing priorities, Development process, Time to market, Alignment between IT & business objectives, Project visibility, Team morale, Requirements management, Productivity, Risk management.

In den nächsten zehn Jahren wird sich die agile Methodenlandschaft sicher auch weiterhin erheblich verändern. Wohin die Reise geht, ist schwer abzuschätzen. Möglicherweise wird es dann agile Vorgehensweisen geben, an die wir heute noch gar nicht denken . . . wie im Jahr 2002 in der Softwareentwicklung noch niemand an Kanban dachte. Oder vielleicht wird statt „agil“ ein anderer Begriff in aller Munde sein.

Im Folgenden werden heute aktuelle Konzepte und Praktiken des agilen Entwickelns von Software (mit eXtreme Programming als typischen Vertreter), des agilen Managements von Projekten (mit PRINCE2 und DSDM Atern als typische Vertreter) und der agilen Neu- und Weiterentwicklung und Wartung von Produkten (mit Scrum, Kanban und SAFe als typische Vertreter) vorgestellt. Den Abschluss bilden einige Gedanken zur insgesamt agilen Organisation.

3.1 Konzepte und Praktiken des agilen Entwickelns von Software

Begonnen hat das inkrementell-adaptive Vorgehen zur Lieferung praktisch nutzbarer Software-Lösungen im Abstand von nur wenigen Wochen bereits in den 1970er-Jahren, gefordert u. a. durch Tom Gilb. Also noch vor dem Entstehen des stark plangetriebenen Vorgehens mit seinem BDUF in den 1980er-Jahren. Diese frühen inkrementell-adaptiven Vorgehensweisen wie etwa das Rapid Application Development (RAD) wurden jedoch als eher chaotisch empfunden und die 1997 erstmals publizierte Dynamic Systems Development Method (DSDM) - damals noch eine reine Software-Entwicklungsmethode - hatte das Ziel, hier etwas mehr Disziplin zu schaffen.

Typisches Beispiel für das agile Entwickeln von Software ist eXtreme Programming (XP), siehe Kap. 3.1.1 in [Ko] und [Wd09a] [Wd09b] [Be99].

XP geht davon aus, dass Software nicht auf der Basis bereits zu Beginn möglichst vollständig erhobener Anforderungen realisiert und danach eingeführt werden kann, sondern dass sich die Anforderungen kontinuierlich und differenzierter erst im Verlauf der schrittweisen Realisierung einzelner Softwareteile (Inkmente) ergeben. Diese Software-Inkmente müssen vom Benutzer getestet oder - besser - praktisch genutzt werden können. Daraus ergeben sich dann weitere und noch spezifischere Anforderungen. Jedes Software-Inkrement wird dabei innerhalb einer Iteration von einer bis maximal vier Wochen realisiert. Eine möglichst enge Zusammenarbeit der Benutzer und der Software-Entwickler und eine direkte Kommunikation sind dabei unabdingbar. So etwa werden die Anforderungen aus Benutzersicht nur als grobe Anwendungsfälle mit wenigen Sätzen als User Stories beschrieben. Sie sind keine umfangreichen und verbindlichen Detailspezifikationen, sondern dienen als Diskussionsanstöße für den Benutzer und Entwickler dann, wenn die Anforderung unter Mitwirkung des Benutzers konkret realisiert werden soll.

3.2 Konzepte und Praktiken des agilen Managements von Projekten

Gibt es überhaupt ein „agiles“ Management von Projekten?

Im Abschnitt *„Mit Unberechenbarkeit und Komplexität umgehen statt sie beherrschen zu wollen“* wurden die beiden gegensätzlichen Vorgehensweisen beim Überqueren eines zugefrorenen Flusses beschrieben. Beim sich schrittweise vorantastenden Vorgehen ergab sich mitten auf dem Weg eine neue Richtung hin auf ein bequemerer Hotel, statt wie ursprünglich geplant zur Hütte. Wäre das Überqueren des Flusses ein Projekt, dann hätte sich das ursprüngliche Ziel deutlich verändert und vermutlich auch die Zeit für die Überquerung des Flusses. Stabil wären nur die eingesetzten Ressourcen (die Anzahl der mitwandernden Personen) und die Qualität geblieben (den Fluss sicher überqueren und eine möglichst angenehme Stube finden). Würden wir auch die verfügbare Zeit begrenzen, dann könnte es sein, dass noch auf dem Fluss einige Meter vor dem Ufer angesichts des Hotels die Überquerung zum Stillstand käme. Wir müssen also noch ein paar Weginkremente (also zusätzliche Zeit für die mitwandernden Personen) investieren, um dieses neue, aber bessere Ziel zu erreichen. Aus klassischer Projektsicht wäre das ein typisches aus dem Ruder gelaufenes Projekt, obwohl das Ergebnis schließlich besser ist als das ursprünglich geplante.

Die Flexibilität bezüglich des Ziels (oder Funktionsumfangs) der Produktentwicklung und allenfalls auch der Anzahl der erforderlichen Produktinkremente (die gesamte Durchlaufzeit) ist essentiell für das agile Vorgehen. Das „eiserne Dreieck“ des klassischen Projektmanagements (fixer Umfang der Ergebnisse, fixe Kosten, fixe Zeit) wird damit ersetzt durch seine inkrementell-adaptive Variante (fixe Qualitätsansprüche, fixe Kosten, fixe Zeit, flexibel daran angepasste Ergebnisse). Auf vertraglicher Ebene passt zu dieser Variante der vielerorts beschriebene „Agile Festpreis“.

All das entspricht jedoch schlecht einem Projekt im üblichen Sinn. Ayelt Komus schreibt dazu [Ko12]:

„Agile Methoden wie Scrum sind keine Projektmanagementmethoden im eigentlichen Sinne. Ein Projekt, bspw. nach DIN 69901, ist durch seine „Einmaligkeit der Bedingungen in ihrer Gesamtheit“ gekennzeichnet. Weiterhin werden Projekten klare Ziele und begrenzte zeitliche und finanzielle Ressourcen zugeschrieben. Scrum als besonders populäre agile Methode hingegen zeichnet sich eben dadurch aus, dass es einen festen Rhythmus gleicher Dauer und gleicher Personalressourcen anstrebt. Die Ziele werden laufend weiterentwickelt ... Im Gegensatz zum klassischen Projektmanagement werden die jeweils im Fokus befindlichen Ziele an den Ressourcen ausgerichtet und nicht umgekehrt.“

Und in einem Interview zu dieser Studie sagt er: [Om12]:

„Interessanterweise werden oft Aufgabenstellungen als Projekte verstanden, die im eigentlichen Sinne gar keine sind. Schließlich ist mit der Einführung eines Softwareproduktes, eines Geschäftsprozesses, eines neuen Autos oder einer neuen SAP-Lösung die Arbeit nicht abgeschlossen. Vielmehr geht es um kontinuierliche Verbesserung der Lösungen und der Nutzung derselben. Wahrscheinlich ist das eines der Hauptprobleme klassischer Projektmanagementmethoden. Sie beruhen auf dem Gedanken: ‚Es gibt eine Fertigstellung und dann ist die Arbeit getan.‘ Außerdem verleitet diese Denkweise dazu, viel zu große Aufgabenstellungen in einem Stück zu bearbeiten.“

Dennoch, so schreibt Ayelt Konus in der Langfassung der Studienergebnisse [Ko12], finden „*agile Methoden Eingang in das Projektmanagement - oft auch als Ergänzung oder Erweiterung in Form eines sogenannten hybriden Ansatzes, also einer vermischten bzw. kombinierten Form agiler und klassischer Methoden. An vielen Stellen werden eigentlich kontinuierliche Prozesse, deren Ziele und Ressourcen eben nicht zu Beginn feststanden, als Projekte gemanagt bzw. bezeichnet.*“

Typische Beispiele für das agile Management von Projekten sind

- PRINCE2™, siehe Kap. 3.2.1 in [Ko] und [HL12] [UK09]
- und Dynamic Systems Development Method (DSDM® / DSDM® Atern), siehe Kap. 3.2.2 in [Ko] und [Ca11] [DS08]

Sowohl PRINCE2 als auch DSDM Atern werden oft als „Wasserfallmethoden“ gesehen. Das entspricht - im Gegensatz zur ihrer häufigen praktischen Anwendung - jedoch nicht ihrer Intention:

Speziell nämlich an PRINCE2 ist

- seine konsequente und im Projektverlauf immer wieder überprüfte Ausrichtung auf die geschäftliche Rechtfertigung, dargestellt als Business Case, der fortlaufend aktualisiert wird;
- die Strukturierung seines Ablaufs nach überprüfbaren aufeinander aufbauenden Teilergebnissen (Produkten);
- die Gliederung in die Managementphasen: Initiierungsphase und danach eine oder mehrere Durchführungsphasen (die jeweils die Produktinkremente liefern);
- dass am Ende jeder Phase und nicht nur am Projektende die gemachten Erfahrungen gesammelt werden (das Projekt als lernende Organisation) und der Business Case überprüft und aktualisiert wird;
- dass es zu Beginn über das gesamte Projekt hinweg nur eine grobe Konzeption und Planung gibt, die Details aber jeweils erst zu Beginn einer Durchführungsphase nur für diese Phase erarbeitet werden;
- dass es stets an die jeweilige Größe, Art und Umgebung des Projekts anzupassen ist und nicht sein umfangreicher Maximalumfang 1: 1 genutzt werden soll (Tailoring). Im Minimum umfasst es nur diese vier Dokumentensätze: Projektleitdokumentation - Projektstatusberichte (auch mündlich möglich) - Projekttagbuch - Projektabschlussbericht;
- dass seine Rollenträger möglichst autonom innerhalb eines definierten Rahmens agieren.

Und DSDM Atern ist dadurch gekennzeichnet:

- Das inhaltliche Projektergebnis wird anpassbar gehalten, Kosten, Termine und Qualitätsanforderungen hingegen sind fixiert.

- Einzelne Phasen oder Phasengruppen können iterativ durchlaufen werden.
- Die Lösung kann schrittweise in einzelnen Inkrementen ausgeliefert werden.
- MoSCoW-Priorisierung (Must Have/Should Have/Could Have/Won't Have this Time) der Teilresultate: Pro Timebox werden etwa 60% Must Have, 20% Should Have, 20% Could Have geplant.
- Förderung von Modellen und Prototypen.
- Kein BDUF, aber ein bewusstes JEDUF - mit allen Stakeholdern vereinbart - als Mittel gegen schleichende Funktionserweiterungen.
- Frühzeitige Risikoanalyse (in den Phasen Machbarkeitsprüfung und Rahmenplanung/Grundlagen).

PRINCE2 ist der De-facto-Standard für Projekte in Großbritannien und die Standardmethode für die Projektmanager-Trainings der Vereinten Nationen. PRINCE2 kommt zentralen Elementen des inkrementell-adaptiven Vorgehens im Projektmanagement entgegen, erzwingt sie aber nicht. Es befasst sich jedoch nur mit den Arbeitsprozessen und Rollen bis zum Teammanager, nicht mit jenen auf der Teamebene selbst. Dort können spezifische Methodenrahmen wie Scrum, XP oder DSDM als Ergänzung angeschlossen werden. [Pr06]

DSDM ist einerseits ein für sich allein stehendes Framework zur projektmäßigen Entwicklung dynamischer, also nicht genau vorausplanbarer Systeme einschließlich eines Rahmens für das Projektmanagement. Andererseits fokussiert DSDM (wie Scrum) insbesondere auch auf die teambasierte Entwicklungsarbeit. Dieser Teil des DSDM kann, wie vorhin bereits erwähnt, auch im Rahmen anderer übergeordneter Projektmanagementmethoden wie PRINCE2 eingesetzt werden [Pr06]. DSDM ist mit seinen Phasen, Prozessen und Outputs aus traditioneller Sicht gut verständlich, andererseits ist es durch essenzielle agile Elemente gekennzeichnet: Es ermöglicht, bei entsprechender Verschmelzung seiner eher sequenziellen Phasen, ein iterativ-adaptives und inkrementelles Entwicklungsvorgehen. Es fordert den fortlaufenden Einbezug der Nutzer und Kunden in den Entwicklungsprozess. Es unterstützt die Ausrichtung auf klar definierte strategische Ziele und auf die frühe Lieferung echter Geschäftsvorteile. DSDM ist das einzige als ISO9001-verträglich akkreditierte agile Verfahren.

3.3 Konzepte und Praktiken der agilen Neu- und Weiterentwicklung und Wartung von Produkten

Wie vorhin bereits erläutert, passt die Flexibilität bezüglich des Ziels (oder Funktionsumfangs) und allenfalls auch der investierten Anzahl von Produktinkrementen besser zu einer den gesamten Produktlebenszyklus betrachtenden Vorgehensweise als zu einem bezüglich Zeit, Kosten und Inhalt begrenzten Projekt. So etwa ist im Scrum Guide [SS] zu lesen: „Solange ein Produkt existiert, existiert auch ein Product Backlog.“. Das Wort Projekt kommt im Scrum Guide nur an einer einzigen Stelle vor: „Jeder Sprint kann als Projekt verstanden werden, für das ein zeitlicher Rahmen von maximal einem Monat zur Verfügung steht.“

Deshalb ist Scrum keine Projektmanagement-Methode (auch wenn sie oft so bezeichnet wird), sondern beschreibt generische Vorgehensregeln einer inkrementell-adaptiven Produktentwicklung, beginnend mit der Realisierung des ersten Produktinkrements über alle weiteren Inkremente der Weiterentwicklung des Produkts hinweg bis zu seinem Lebensende.

Auf den gesamten Lebenszyklus von Produkten, nicht nur auf die Phasen seiner projektmäßigen Bearbeitung (Erstentwicklung und umfassende Überarbeitungen) beziehen sich auch das Scaled Agile Framework (SAFe) und Kanban. Wobei Kanban nicht nur das Entwicklungsvorgehen beschreibt, sondern für alle Arten sequenziell aufeinander folgender Arbeitsschritte nutzbar ist.

Typische Beispiele für die agile Neu- und Weiterentwicklung und Wartung von Produkten sind

- auf Teamebene:
 - Scrum, siehe Kap. 3.3.1 in [Ko] und [Sc] [SAa] [SS] [SAb]
- für alle Arten kontinuierlicher Arbeitsflüsse auf Basis sequentieller Bearbeitungsschritte:
 - KANBAN (in Großbuchstaben), eine 2007 veröffentlichte Methode zur evolutionären Verbesserung des Prozesses der Softwareentwicklung und Kanban (in Kleinbuchstaben), ein für die Softwareentwicklung stark abgewandeltes dezentrales Selbststeuerungssystem auf der Grundlage der bei der Toyota Motor Corporation entwickelten Methode der Produktionsablaufsteuerung auf der Basis des Pull-Prinzips, siehe Kap. 3.3.2 in [Ko] und [An11]
- für kontinuierliche Lieferungen von Produktinkrementen in Multiteam / Multiprodukt-Umgebungen:
 - Scaled Agile Framework for Enterprise™ (SAFe) [Le13] [Le] [Le10]

Scrum beschreibt kein Projektvorgehen von der Projektinitialisierung bis zum Projektabschluss, sondern ist ein Framework zum Vorgehen bei der Neu- und Weiterentwicklung komplexer, im Voraus also nicht genau definierbarer und planbarer Produkte auf der Basis empirischer Prozesskontrolle. Das illustriert z. B. auch Ken Schwaber in einem Fallbeispiel, bei dem eine existierende Großrechneranwendung auf eine Webanwendung umgestellt werden sollte. Die Fallbeschreibung beginnt dort, wo das Projekt bereits bewilligt, seine Finanzierung gesichert und der Projektplan und das Entwicklerteam schon vorhanden waren [Sc07]. Scrum beschreibt, unabhängig von der Art des Produkts, wie das Scrum-Team innerhalb einer immer gleich langen Timebox (genannt Sprint) von zwei bis maximal vier Wochen ein fertiges, nutzbares und potenziell auslieferbares Produktinkrement herstellen kann. Das Scrum-Team besteht aus dem Product Owner, den Entwicklern (als Entwicklerteam) und dem Scrum Master (der dem Team dienenden Führungskraft). Zu Beginn jedes Sprints wird im Scrum-Team vereinbart, was für das nächste Produktinkrement zu leisten ist. Basis dafür ist eine fortlaufend aktualisierte und nach Erledigungsreihenfolge geordneten Liste (genannt Product Backlog) von all dem,

was im Produkt künftig benötigt werden könnte. Am Ende des Sprints werden - unter Einbezug relevanter teamexterner Stakeholder - das effektiv Geleistete (Sprint-Review) und die Arbeitsweise während des Sprints (Sprint-Retrospektive) überprüft. Diese Überprüfung dient der fortlaufenden Verbesserung sowohl des Produkts (Review) als auch des Prozesses der Produktentwicklung (Retrospektive). Das - und nicht mehr oder weniger - beschreiben die Regeln von Scrum [SS]. Sie beschreiben z. B. nicht, wie die Elemente des Product Backlogs vor dem ersten Sprint ermittelt, beschrieben, bezüglich Machbarkeit und Risiko beurteilt und mit den Backlogs anderer Produkte koordiniert werden und wie die fortlaufende Aktualisierung und Abstimmung mit anderen Teams und Produkten erfolgt. Scrum gibt auch keine Hinweise, wie eine Handvoll bis mehrere Dutzend Teams gleichzeitig an verschiedenen Produkten und Projekten arbeiten können. Scrum beschreibt auch kein spezielles Vorgehen, wie während eines Sprints ungeplante und sehr dringende Fehler bereits produktiver Produktinkremente behoben werden können. All das und die je nach Produktart ganz unterschiedlichen Methoden und Techniken der Analyse, des Designs, der Modellierung, der Konstruktion/Realisierung, der Tests und der Qualitätssicherungsmaßnahmen und der Inbetriebnahme und fortlaufenden Produktbetreuung müssen ergänzend erarbeitet und vereinbart werden. Scrum will alle diese spezifischen Aspekte auch nicht regeln, sondern ist *„weder ein Prozess noch eine Technik zur Erstellung von Produkten; es ist vielmehr als Framework zu verstehen, innerhalb dessen verschiedene Prozesse und Techniken zum Einsatz gebracht werden können.“* [SS]. Die auf den ersten Blick einfach erscheinenden Regeln von Scrum führen allerdings auch zu ungerechtfertigten Erwartungen und damit verbundenen Fehlnutzungen, siehe siehe Kap. 3.3.1 in [Ko] und [Ko13]:

- Scrum ist keine schlanke Entwicklungsmethode. Es ist ein allgemeines Vorgehens-Framework und ein verfügbarer leerer Container für etablierte spezifische Methoden der professionellen Produktentwicklung.
- Scrum führt nicht zu einer massiven Erhöhung der Effizienz: Es unterstützt die Adaptivität, Qualität und Effektivität. Insgesamt (bezogen auf ein gesamtes fertiges Produkt) wird es mit Scrum nicht eindeutig schneller oder billiger [Ha08].
- Scrum erfordert - wie auch andere agile Vorgehensweisen - die strikte Professionalität erfahrener Entwickler [BT03].
- Scrum legalisiert kein Arbeiten auf Zuruf, sondern setzt ein agiles und professionelles Anforderungsengineering und kontinuierlich adaptierte Design- und Architekturrahmen voraus.
- Scrum kann nicht für JEDE Art von Vorhaben eingesetzt werden, sondern ist dann geeignet, wenn EIN Produkt von einem oder wenigen ausschließlich dafür arbeitenden interdisziplinären Teams allein (also ohne Abhängigkeit von team-externen Spezialisten) so neu- oder weiterentwickelt wird, dass lieferbare Teile davon nach jeweils bereits zwei bis vier Wochen vorhanden sind. Und zwar so, dass die raschen Rückmeldungen ihrer Nutzer unmittelbar zur Steuerung der weiteren inhaltlichen Entwicklung dienen.

- Auch Selbstorganisation braucht Führung: Das Team als Ganzes ist selbstverantwortlich - jedoch innerhalb klar vorgegebener Spielregeln und Rahmenbedingungen, vgl. S. 77 - 81 in [Ko13]

KANBAN (in Großbuchstaben) dient der evolutionären Prozessverbesserung indem es die aktuell praktizierten Arbeitsweisen sicht- und messbar (quantifiziert und statistisch auswertbar) macht und damit objektive Voraussetzungen für schrittweise, auch experimentelle, Verbesserungen schafft. Es geht vom Existierenden und Etablierten aus ohne Anspruch, es sofort verändern zu wollen, und erlaubt evolutionäre Veränderungen in jener Art und Geschwindigkeit, wie sie der Organisation angemessen sind. Das reduziert den Widerstand gegen Veränderungen erheblich.

Kanban (in Kleinbuchstaben) dient der operativen dezentralen (Selbst-) Steuerung der Arbeit auf Basis des Pull-Prinzips. Es ist extrem flexibel und rasch an spezielle, auch nicht vorausgesehene Situationen anpassbar. Die Arbeitsschritte - die Spalten der Kanban-Tafel - und deren Work-In-Progress-Limits (WIP-Limits) können im Sinn der empirischen Prozesskontrolle rasch und auch experimentell verändert werden. In Organisationen mit einem ausgeprägten Top-down-Steuerungs- und Kontrollbedürfnis auch auf Detailebene kann es jedoch zum Gefühl des Kontrollverlusts führen und auch zu Unsicherheiten bezüglich der längerfristigen Planbarkeit und Zuverlässigkeit der Fertigstellungstermine der Arbeitspakete. Erst eine Reihe positiver Erfahrungen mit „trotz“ Pull-Prinzips termingerechten Just-in-time-Lieferungen vermag diese Skepsis ab- und Vertrauen aufzubauen.

Die Grundprinzipien agiler Vorgehensweisen beziehen sich in der Regel auf nur EIN Produkt/Projekt, dessen Inkremente von EINEM Team in einer relativ homogenen Systemlandschaft entwickelt und rasch integriert werden. Das „Scaled Agile Framework for Enterprise (SAFe)“ beschreibt, wie agile Vorgehensweisen für Multi-Team- bzw. Multi-Produkt-Situationen bezogen auf den gesamten Lebenszyklus der Produkte in großen Unternehmen mit einer hohen technologischen Vielfalt bis auf die Ebene des Portfolio-Managements skaliert werden können. Das SAFe offeriert umfassende Beschreibungen der individuellen Rollen, Teams, Tätigkeiten und Artefakte zum agilen Arbeiten auf den Ebenen

- Teams (dort wird als Vorgehen Scrum in Verbindung mit XP empfohlen),
- Programm/Agile Release Trains (diese liefern alle etwa zehn Wochen = fünf Scrum-Sprints in die gesamte IT- Systemlandschaft integrierte Potential Shippable Increments, die aus Beiträgen mehrerer Teams und IT-Plattformen bestehen),
- Portfolio (dort wird ein an Kanban angelehntes Vorgehen empfohlen).

Zentrale Aspekte des SAFe sind:

- Skalierung der zu schaffenden Werte: Nicht alles ist eine User-Story.
- Skalierung von Team und Timebox: Kein Team ist eine Insel.
- Skalierung des Design: Emergentes Design verbindet sich mit bewusst vorausgedachter Architektur.

- Skalierung des Portfoliomanagements: Das althergebrachte Denken in Jahresbudgets hinterfragen und ändern.
- Skalierung der Führerschaft: Das Unternehmen kann nur so „lean“ wie das Denken der Geschäftsführung sein.

Das SAFe versteht sich im Gegensatz zu einem Projektmanagement-Framework als agiles Framework zur kontinuierlichen Lieferung produktiv nutzbarer Lösungen in (internen) Releases im Rahmen der Neu- und Weiterentwicklung und der Wartung von Produkten längs ihrer gesamten Lebenszeit. Angesichts seiner im Internet [Le] frei verfügbaren sehr umfangreichen Beschreibung der Rollen und Praktiken wird das SAFe von einigen Agilisten als „schwergewichtig“ und „wasserfallorientiert“ gesehen, von anderen dennoch als unbestreitbar agil. [Ru12] Dabei zu berücksichtigen ist, dass die vom SAFe angebotenen Praktiken, obwohl aufeinander abgestimmt, auch nur teilweise genutzt werden können. SAFe verlangt nicht, alle seine Praktiken vollumfänglich zu verwenden. Und es bleibt auch zu bedenken, dass andere unbestritten als „schlank“ gesehene agile Rahmenwerke - wie etwa Scrum - als solche allein nicht nutzbar sind sondern zusätzlich eine Vielzahl spezifischer Methoden der professionellen Produktentwicklung erfordern.

3.4 Die insgesamt agile Organisation

Seit einigen Jahren, inspiriert von der Verbreitung des agilen Vorgehens in der SW-Entwicklung, werden die im Abschnitt *Die Vieldeutigkeit des Begriffes „agil“* genannten Aspekte auch als Prinzipien „agiler“ Organisationen und Unternehmen insgesamt empfohlen. Da jedoch Unternehmen nebst der - allenfalls inkrementell-adaptiv gestaltbaren - Entwicklung und Realisierung von Produkten auch viele Prozesse ohne Produktcharakter (wie etwa das Personalwesen) umfassen, werden unter agil dann nicht primär das inkrementell-adaptive Vorgehen sondern diverse post-tayloristische, hierarchie- und autoritätsfreie und selbststeuernd-kollaborative Prinzipien und Überzeugungen verstanden. Oder die in 2.4.4 genannten allgemeinen Aspekte zur Sicherung der Überlebens- und Entwicklungsfähigkeit.

All das sind jedoch in der Organisationsentwicklung schon lange bekannte und diskutierte Ideen. Einige davon wurden - allerdings in vergleichsweise wenigen Fällen - erfolgreich umgesetzt. Vielfach zitierte Beispiele sind u. a. SEMCO [Se95] [St12] und die Morning Star Company Inc. [Gi12]. Weitere Fälle, die man heute als agil bezeichnen könnte, finden sich in [Ra03].

Der Versuch, diese heute unter „agil“ subsumierten Prinzipien im Rahmen der mehrheitlich etablierten Führungsstrukturen und der heute üblichen Vorgehensweisen im Gesamtunternehmen einzuführen ist tatsächlich wie „neuer, noch gärender, Wein in alte Schläuche“, braucht also „neue Schläuche“, also ein im Unternehmenskontext grundsätzlich verändertes Denken und Handeln. Diese Forderung nach „neuen Schläuchen“, also nach einem neuen Denken und Handeln, wird jedoch bereits seit Jahrzehnten erhoben, ist also ein „Déjà-vu“. Allerdings immer noch ein selten erfülltes.

Dieses veränderte Denken und Handeln setzt voraus, dass beginnend bei der Unternehmensspitze

- das „alles im Griff haben“ ersetzt ist durch ein „vertrauensvolles Loslassen“. Also: „Kontrolle ist gut - Vertrauen ist besser“
- die Überzeugung vorherrscht, dass ein Unternehmen erst durch die fortlaufende Konversation aller das Unternehmen umfassenden Mitarbeitenden und Stakeholder entsteht, nicht durch von oben verkündete Visionen oder verordnete Strukturen.

Diese Voraussetzungen zu erfüllen ist jedoch sehr anspruchsvoll.

Oft wird vorgeschlagen, diese Veränderung „von unten nach oben“ voranzutreiben. Ein soziales System „bottom up“ zu gestalten ist eine bereits Mitte des 19. Jahrhunderts von Michail Alexandrowitsch Bakunin, dem Mitbegründer des kollektivistischen Anarchismus, formulierte Vorstellung: Er lehnt jede Form der von oben nach unten wirkenden institutionalisierten und zentralisierten Autorität ab und akzeptiert statt dessen nur die Autorität von Spezialisten, weil sie nicht aufgezwungen sondern aus eigener Einsicht anerkannt wird. Deshalb fordert er auch den Aufbau der Gesellschaft von unten nach oben auf Basis freier Zusammenschlüsse und Interessensgemeinschaften der Menschen um eine möglichst große Autonomie sicherzustellen. [Ba05]

Aktuelle Erkenntnisse der Organisationsentwicklung widerlegen jedoch die Möglichkeit, ein Unternehmen (als soziales System) ausschließlich „bottom up“ verändern zu können. Essenziell nämlich ist die Unterstützung durch die Entscheidungsträger – bis hinauf zum Top Management auf Geschäftsleitungsebene als „institutionell vorgesetzte“ und nicht „von unten legitimierte“ Autoritäten. Simone Inversini schreibt dazu: *„Es erweist sich als von zentraler Bedeutung, die Entscheidungsträgerinnen und -träger der Organisation verbindlich in die Veränderung einzubinden. Diese Ergebnisse widersprechen klar der von frühen Vertreterinnen und Vertretern der OE (Organisationsentwicklung) aus der gruppendynamischen Tradition geübten Kritik an der Hierarchie an sich, mit welcher rein partizipative Ansätze legitimiert wurden. Die Ergebnisse veranschaulichen, dass eine Veränderung mit mangelhaft agierenden bzw. ohne Machtpromotorinnen und -promotoren wenig Chancen hat.“* [In05]

4 Schlussfolgerungen

In dieser Arbeit wurde gezeigt, dass der Begriff „agil“ ein sehr breites Spektrum an Bedeutungen hat - insbesondere dann, wenn die Domäne der SW-Entwicklung verlassen wird. Die Aussage „wir arbeiten agil“ ist daher eher missverständlich als klärend. Dargestellt wurden auch die heute verbreiteten agilen Konzepte und Praktiken und deren historische Wurzeln und typische Vertreter dieser Praktiken zum Entwickeln von Software, für das Management von Projekten und zur Neu- und Weiterentwicklung und Wartung von Produkten. Dargestellt wurde auch, dass die Gleichsetzung agil = Scrum viel zu kurz greift und damit die Chance vergeben wird, von den anderen derzeit verbreiteten weiteren agilen Verfahren zu lernen und das Risiko besteht eine der gerade populären statt die

im jeweiligen Kontext am besten passende Vorgehensweise zu nutzen. Den Abschluss bildete die kritische Würdigung der Idee einer insgesamt agilen Organisation mit Verweis auf die bereits lange Ideengeschichte in diesem Bereich und mit der Darlegung der Schwierigkeiten der Umsetzung dieser Ideen, die beginnend bei der Unternehmensspitze eine gegenüber dem „Üblichen“ grundlegend veränderten Art des Denkens und Handelns erfordert.

All jene, die vor der Frage stehen, ob und wie das „agile“ Vorgehen ihre aktuelle Berufspraxis verbessern könnte haben in dieser Arbeit Hinweise erhalten, dass diese Frage viel differenzierter zu stellen ist: Was von den unterschiedlichen Bedeutung von „agil“ ist in erster Linie für die Verbesserung der heutigen Situation relevant? Geht es primär um ein inkrementell-adaptives Vorgehen oder primär darum, mehrere Lösungsmöglichkeiten einzusetzen und nahtlos von einer zur anderen überzugehen, auf eine Veränderung der Umgebung rechtzeitig zu reagieren und Arbeitsprozesse und die Organisation zu ändern? Geht es um grundlegende Verbesserungen im Bereich der SW-Entwicklung oder des Projektmanagements oder der kontinuierlichen Weiterentwicklung von Produkten längs ihres gesamten Lebenszyklus? Oder geht es darum, das Unternehmen als Ganzes neu zu gestalten? Die in dieser Arbeit angebotenen zahlreichen Referenzen erlauben dabei das gezielte weitere Eintauchen in spezifische Themenfelder.

Diese Aspekte sollten in naher Zukunft vertieft werden:

- Wie können die heute in der Regel getrennt weiterentwickelten und angewendeten Praktiken kooperieren? Etwa so, wie SAFe auf Teamebene Scrum und XP nutzt oder so, dass PRINCE2 einen PM-Rahmen für mehrere Scrum-Teams bildet.
- Wie kann ein inkrementell-adaptives Vorgehen bei der Produktentwicklung in sehr großen Unternehmen so funktionieren, dass viele Produkte / Produktfamilien und mehrere tausend Entwickler weltweit in hunderten von Teams koordiniert zusammenarbeiten? Wie kann hier eine weitestgehende Selbststeuerung der Teams funktionieren?
- Wie kann es gelingen, dass beginnend bei der Unternehmensspitze das „vertrauensvolle Loslassen“ zum Normalfall wird und „Kontrolle ist gut - Vertrauen ist besser“ zum Leitspruch?

Literaturverzeichnis

- [AH09] Alberts, D. S.; Hayes, R. E.; Honekamp, Wilfried (Übersetzer): Power to the Edge. Re Di Roma Verlag, 2009.
- [An11] Anderson, D. J.: Kanban: Evolutionäres Change Management für IT-Organisationen. Dpunkt Verlag, 2011.
- [Ba05] Bakunin, M.: Die revolutionäre Frage. Föderalismus, Sozialismus, Antitheologismus. Münster, 2005.
- [Be01] Beck et al.: Manifesto for Agile Software Development. 2001. Siehe: <http://agilemanifesto.org/>

- [Be99] Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, 1999.
- [BT03] Boehm, B.; Turner, R.: Using Risk to Balance Agile and Plan-Driven Methods. In: IEE Computer. IIEEE, June 2003; S. 57–66
- [Ca11] Caine, M.: DSDM Atern im Überblick. 2011. Siehe: <http://www.netzwoche.ch/de-CH/News/2011/10/19/DSDM-Atern-im-Ueberblick.aspx>
- [Co02] Coldewey, Jens: Multi-Kulti: ein Überblick über die agile Entwicklung. In: OBJEKTspektrum Ausgabe 01/2002.
- [Co12] Coldewey, J.: Was heißt hier eigentlich „agil“? Kennzeichen agiler Organisationen. In: OBJEKTspektrum Ausgabe 05/2012.
- [Do12] Dobelli, R.: Die Kunst des klugen Handelns. Carl Hanser Verlag, 2012; S. 185
- [DS08] DSDM Consortium: DSDM Atern Handbook. Whitehorse Press Limited, 2008. Siehe: <http://www.dsdm.org/content/dsdm-atern-handbook>
- [Fu08] Fuchs, C.: Internet and Society: Social Theory in the Information Age. Routledge, Reprint edition, 2008.
- [Gi12] Gillies, C.: Cheflose Unternehmen: Ein Hauch von Anarchie. Handelszeitung 01.06.2012, siehe: <http://www.handelszeitung.ch/management/cheflose-unternehmen-ein-hauch-von-anarchie>
- [Ha08] Harwardt, Mark: Wasserfallmodell versus Scrum – Ist der gute Ruf der agilen Methode gerechtfertigt? Masterarbeit. Lehrgebiet Programmiersysteme, Fernuniv. Hagen, 2008.
- [Ha11] Hagel III, J. et al.: The 2011 Shift Index. Deloitte Center for the Edge, http://www.deloitte.com/assets/Dcom-UnitedStates/Local20Assets/Documents/TMT_us_tmt/us_tmt_shiftindex_revised_120512.pdf
- [HL12] Hoffmann, G.; Lee, J. M.: Kontrolle ohne Bürokratie: Projektflexibilität durch PRINCE2. In (Hg.: Kammerer, S.; Amberg, M.; Lang, M. Hrsg.): Führung im IT-Projekt.. Symposion Publishing. 2012.
- [In05] Inversini, S.: Wirkungsvolles Change Management in Abhängigkeit von situativen Anforderungen. Humanwissenschaftlichen Fakultät der Universität Potsdam, 2005. Siehe: <http://opus.kobv.de/ubp/volltexte/2005/549/pdf/inversini.pdf>; Kurzfassung: http://www.iafob.ch/deu/10j/downloads/Teil_3_6.pdf
- [JS12] Janes, Andrea; Succi, Giancarlo: The Dark Side of Agile Software Development. In: Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software. ACM, 2012. Siehe: <http://darkagilemanifesto.org/dark-side-of-agile-janes-succi-splash-2012.pdf>
- [KM12] Kropp, M.; Meier, A.: Swiss Agile Study 2012. Fachhochschulen Zürich und Nordwestschweiz, 2012. Siehe: www.swissagilestudy.ch
- [Ko] Korn, H.-P.: Agile Konzepte. In (Bartsch, O.; Lindinger, M. Hrsg.): IT-Servicemanagement. TÜVRheinland. Siehe: <http://www.tuev-media.de/produkte/91154-it-servicemanagement.php>
- [Ko12] Komus, A.: Ergebnisbericht (Langfassung) Studie: Status Quo Agile. Verbreitung und Nutzen agiler Methoden. Studie des BPM-Labors der Hochschule Koblenz, 2012. Siehe: www.status-quo-agile.de
- [Ko13] Korn, H.-P.: Redesigned Agile in großen Unternehmen. In (Korn, H.-P.; Berchez, J. P. Hrsg.): Agiles IT-Management in großen Unternehmen. Symposion Publishing, 2013.
- [Kr09] Krüger, W.: Organisation als Kunstwerk. Abschiedsvorlesung Justus-Liebig-Universität Gießen, Lehrstuhl für Unternehmensführung und Organisation, 13.02.2009.
- [KS03] Kurtz, C.; Snowden, D.: The New Dynamics of Strategy: Sense-making in a Complex-Complicated World. In: IBM Systems Journal, vol. 42, no. 3. 2003; S. 462-483
- [Le] Leffingwell, D.: Scaled Agile Framework. Siehe: <http://scaledagileframework.com/>
- [Le09] Leffingwell, D.: Thoughts on Lean Thinking. Blog vom 15. Sept. 2009, scalingsoftwareagilityblog.com/thoughts-on-lean-thinking/
- [Le10] Leffingwell, D.: Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise. Addison-Wesley, 2010.

- [Le13] Leffingwell, D.: Agil bis zur Unternehmensstrategie - ein „Big Picture“. In (Korn, H.-P.; Berchez, J. P. Hrsg.): Agiles IT-Management in großen Unternehmen. Symposion Publishing, 2013.
- [Lex10] Lexikon IT-Management. Symposion Publishing, 2010.
- [Lu00] Luhmann, N.: Vertrauen: Ein Mechanismus der Reduktion sozialer Komplexität. UTB, Stuttgart, 2000.
- [Lu84] Luhmann, N.: Soziale Systeme. Suhrkamp, 1984.
- [Ma90] Martin, J.: Information Engineering, Book I/II/III. Prentice Hall, 1990.
- [NM09] Nissen, V.; Mladin, A.: Messung und Management von IT-Agilität. In (Fröschle, H-P Hrsg.): Wettbewerbsfaktor IT. HMD 269, 46. Jahrgang, 2009.
- [OEG01] Olson, E. E.; Eoyang, G. H.: Facilitating Organization Change: Lessons From Complexity Science. Pfeiffer, 2001.
- [Om12] Omann, D.: Quo vadis, Agile? borisgloger newsletter, Oktober 2012. Siehe: http://borisgloger.com/newsletter/oktober-2012/quo-vadis-agile/?utm_source=Newsletter&utm_campaign=89cc7711d1-NL201210
- [Pa12] Parsons, T.: The social system. University of California Libraries, 1951 und Quid Pro, LLC, 2012.
- [Po02] Poppendieck, M.: Lean Design. March 18, 2002, <http://www.leanessays.com/2010/11/lean-design.html>
- [Pr06] Hartmann Preuss, D.: Case Study: DSDM Bridges the Gap Between PRINCE 2 and XP. InfoQ News Oct 06, 2006. Siehe: <http://www.infoq.com/news/XP-DSDM-Prince2>
- [Ra03] Radatz, S.: Evolutionäres Management: Antworten auf die Management- und Führungsherausforderungen im 21. Jahrhundert. Verlag Systemisches Management, 2003.
- [Ru12] Rüssel, F.: Is SAFE unSAFE – My Thoughts. Blog vom 7. Aug. 2012 in <http://www.agilerescue.de/is-safe-unsafe-my-thoughts/>
- [SAa] Scrum Alliance: Why Scrum? Siehe: <http://www.scrumalliance.org/why-scrum>
- [SAb] Scrum Alliance: Core Scrum. Agile Atlas. Siehe: <http://agileatlas.org/atlas/scrum>
- [Sc] Schwaber, K.: What is Scrum? Siehe: <http://www.scrum.org/Resources/What-is-Scrum>
- [Sc07] Schwaber, K.: Agiles Projektmanagement mit Scrum. Microsoft Press Deutschland, 2007. S. 72
- [Se95] Semler, R.: Maverick: The Success Story Behind the World's Most Unusual Workplace. Grand Central Publishing, Reprint edition, 1995.
- [SS] Schwaber, K.; Sutherland, J.: The Scrum Guide. Siehe: <http://www.scrumguides.org/>
- [St12] Stacho, P.: Das beste Managementbuch aller Zeiten: Zusammenfassung Semco-System von Ricardo Semler (Demokratisches Managementmodell). Blog vom 2. Juli 2012, Siehe: <http://geschaeftsmann20.com/2012/07/02/semcosystem/>
- [Sw12] SwissQ: Agile Trends & Benchmarks 2012. Siehe: www.swissq.it/swissq-agile-trends-benchmarks-2012-3, weitere Studien in www.swissq.it/ressourcen
- [SW97] Schuh, G.; Wiendahl, H.-P. (Hrsg.): Komplexität und Agilität: Steckt die Produktion in der Sackgasse? Springer, 1997.
- [Ta09] Tavra, D.: Vertrauen als Mechanismus der Reduktion von Komplexität; Universität Bern Sozialanthropologisches Institut, 2009. Siehe: <http://tinyurl.com/a62hqaw>
- [Tu02] Türcke, C.: Erregte Gesellschaft. Philosophie der Sensation. C.H. Beck, 2002; S. 130, Anm. 11
- [UK09] UK Office of Government Commerce: Erfolgreiche Projekte managen mit PRINCE2. The Stationery Office Ltd, 2009.
- [Wd09a] Wells, D.: Einführung in das Extreme Programming. 2009. Siehe: www.extremeprogramming.org/donwells.html
- [Wd09b] Wells, D.: Extreme Programming. Siehe: www.extremeprogramming.org
- [We12] Wernham, B.: Agile Project Management For Government. Maitland and Strong, 2012.
- [Wo03] Wolf, H.: Partizipatives Management - was bleibt? Expertise für die Hans-Böckler-Stiftung. Soziologisches Forschungsinstitut Universität Göttingen, Juli 2003.

Was fehlt Scrum? – ein beispielhafter Lösungsansatz aus der Hochschulpraxis

Jörg Hofstetter, Martin Jud

Hochschule Luzern Technik & Architektur
Technikumstrasse 21, 6048 Horw, Schweiz
joerg.hofstetter@hslu.ch
martin.jud@hslu.ch

Abstract: Der Einsatz eines agilen, iterativ inkrementellen Vorgehensmodells wie Scrum ist heute in der Software-Industrie State of the Art. Wesentliche Aspekte von Softwareprojekten, wie der Umgang mit ändernden Anforderungen oder der Druck auf eine frühe Auslieferung, können damit sehr gut beherrscht werden.

Es gibt aber auch wichtige Aspekte der Software-Erstellung welche in Scrum nur marginal oder gar nicht unterstützt werden. Dies betrifft insbesondere den Umgang mit Architektur-Fragen, komplexen Anforderungen und die Integration in einen Projektmanagement-Gesamtrahmen. Die fehlende Unterstützung durch Scrum für diese Aspekte wird auch von andern Autoren als Mangel empfunden. Sie zeigen aber auch, dass sich eine Unterstützung dieser Aspekte durchaus mit Scrum vereinbaren lässt. Wie dies aussehen kann – ohne die ursprüngliche Kraft und Einfachheit von Scrum zu gefährden – zeigen wir Beispielhaft etwas genauer an einem konkreten für unsere Hochschule entwickelten Vorgehensmodell.

1 Warum Scrum?

1.1 Warum agiles, iteratives Vorgehen in Software Projekten essentiell ist

Ingenieurwissenschaften wie z.B. das Bau- und Maschineningenieurwesen haben über Jahrzehnte hinweg erfolgreiche Vorgehensmodelle geschaffen um höchst komplizierte Probleme zu lösen. In den Anfängen der Informatik orientierte man sich naheliegenderweise an diesen etablierten Modellen, musste aber feststellen, dass eine einfache Übertragung nicht möglich ist.

In den klassischen Ingenieur-Disziplinen kennt man eine klare Trennung zwischen den Bereichen Konzipieren (Entwerfen, Konstruieren) und dem eigentlichen Produzieren (Herstellen, Umsetzen). Neben einer zeitlichen Entkopplung dieser Bereiche, sind es typischerweise auch verschiedene Personen, welche mit unterschiedlichen Skills die entsprechenden Tätigkeiten ausführen. Meist fällt der grössere Teil der Kosten im Bereich der Produktion an.

Die Herstellung von Software unterscheidet sich davon grundsätzlich: Konzipierung und Implementierung werden oft vom gleichen Team aus Personen mit ähnlichen Skills ausgeführt. Die Arbeitsteilung Analytiker / Programmierer hat sich nicht halten können, heute erfolgt die Arbeitsteilung in Software-Projekten eher entlang von Komponenten und Subsystemen als zwischen den Herstellungsphasen. Friedrichsen [Fr10] bezeichnet Konzipieren und Realisieren als Entwurfsphase, die im Gegensatz zu den klassischen Ingenieurdisziplinen relativ teuer ist und grenzt diese gegen die Bauphase – das eigentliche Produzieren – ab, das im Wesentlichen bei der Software ein Kopieren und damit kostengünstig und schnell ist.

Software und damit Software-Projekte umfassen viele Einflussgrößen, welche eng und dynamisch miteinander gekoppelt sind. Eine getroffene Annahme kann das gesamte System stark beeinflussen und zu neuen Randbedingungen und Möglichkeiten führen. Siehe auch „Spezielle Eigenschaften der Software“ in [LL10].

Während es z.B. beim Bau einer Brücke kaum denkbar ist, in einem ersten Schritt eine „normale“ Brücke zu bauen, diese dann in einem weiteren Schritt auf eine 4-spurige Brücke auszubauen, sind solche Dinge in der Software-Erstellung durchaus möglich. Die iterativ- inkrementelle Herangehensweise, d.h. die laufende und parallele Weiterentwicklung von Anforderungen, Architektur/Design und Umsetzung, eröffnet neue Möglichkeiten und Chancen um innovative Softwareprodukte auf dem Markt zu lancieren, mündet aber auch in einer entsprechenden Prozess-Komplexität.

In einem solchen Umfeld bieten klassische, eher lineare Vorgehensweisen kaum Vorteile gegenüber eher empirischen Problemlösungen. Nicht zuletzt weil in der Software-Entwicklung die Kosten für die Realisierung nicht wesentlich höher sind als für die Konzeptionierung, drängt sich ein schrittweises Vorgehen, ein „Herantasten“ auf.

Ein Projekt- und Vorgehensmodell für den Softwarebereich muss aus unserer Sicht zwingend auf diese Eigenheiten eingehen. Es ist sicher kein Zufall, dass iterativ- inkrementelle Vorgehensmodelle vor allem in der Softwareherstellung entwickelt wurden und inzwischen in der industriellen Software-Praxis weit verbreitet sind (siehe dazu auch die Studie [KM12]). Die wachsende Komplexität auch in anderen Bereichen der Technik und die immer besseren Möglichkeiten im Prototypenbau ebnen nun den Weg für den vermehrten Einsatz solcher Vorgehensmodelle auch in anderen Disziplinen.

In der industriellen Software-Praxis verzeichnen wir aktuell einen starken Trend zum Einsatz des Vorgehensmodells Scrum [SAL]. Aus unserer Sicht zeichnen sich Scrum und ähnliche Vorgehensweisen insbesondere durch folgende Erfolgsfaktoren aus:

- Taktung der Arbeit: Durch klar definierte Iterationen ergibt sich im Projekt eine klare Arbeits-Taktung. Die Mitarbeitenden haben klare Ziele vor Augen und diese werden in überschaubaren Intervallen überprüft. Im Idealfall stellt sich nach einigen Iterationen ein produktiver „Arbeits-Flow“ ein.
- Definition of done: Es werden messbare Ziele gesetzt. In der Software-Herstellung kann dies erfahrungsgemäss nur lauffähige, getestete Software sein.

Damit kann der Gefahr von nicht umsetzbaren Konzepten und ewig unfertiger Software begegnet werden.

- Durchgängiger Einbezug der Fachebene in den Entwicklungsprozess: Bei einem agilen Vorgehen werden laufend wichtige Entscheide getroffen. Die Entwickler können und sollen diese nicht alleine fällen. Ein „Produktverantwortlicher“ (Product-Owner) muss entsprechend in die Pflicht genommen werden, d.h. die fachliche Ebene wird in die Entwicklung eingebunden.

2 Was fehlt Scrum?

Obwohl Scrum als Vorgehensmodell in Software-Projekten zunehmend und durchaus erfolgreich eingesetzt wird, sei die Frage erlaubt: Ist mit dem Einsatz von Scrum das Thema Vorgehen und Projektmanagement für ein Projekt abschliessend behandelt? - Wir meinen nein! Scrum stellt ein sehr hilfreiches Instrument dar, welches aber (in vielen Projekten) ergänzende Elemente benötigt.

In den nachfolgenden Kapiteln sollen folgende Aspekte beleuchtet werden, welche Scrum nur marginal abdeckt:

1. neuartige, grosse Systeme, für die zuerst eine Systemarchitektur entworfen werden muss
2. Komplexe Ausgangslagen, die eine vorgängige Situations- und Anforderungsanalyse erfordern
3. Softwareprojekte im Rahmen technischer Produktentwicklung (HW/SW-CoEntwicklung)

Die mangelnde Unterstützung für diese Aspekte hat wohl damit zu tun, dass gemäss Scrum Ur-Paper von Schwaber [Sc95] Scrum sich explizit auf Software-Wartung beschränkt:

Scrum is concerned with the management, enhancement and maintenance of an existing product, (. . .) Scrum is not concerned with new or re-engineered systems development efforts.

Im aktuellen Scrum Guide [Su11] steht genau das Gegenteil:

Scrum wird (. . .) bei der Umsetzung komplexer Produktentwicklungen verwendet.

Allerdings hat sich methodisch in Scrum nichts verändert, was diesen Wandel im Anspruch unterstützen könnte. In der Software Community wollen viele das Potential nutzen, das unbestritten in Scrum steckt und haben entsprechend nach Lösungen gesucht, um die angesprochenen Problemfelder beherrschen zu können.

2.1 Systemarchitektur und Scrum verbinden

Die Stärken von Scrum kommen dann zum Tragen, wenn für eine bestehende Architektur Anpassungen oder Erweiterungen entwickelt werden. Was aber, wenn in einem Projekt eine ganz neue Architektur mit entsprechenden Risiken entwickelt werden muss? Wir stellen fest, dass das Thema Systemarchitektur in Scrum eher am Rande behandelt wird.

Die Frage, wie Anforderungen technisch umgesetzt werden, also die Software-Architektur bzw. der Systementwurf, erfolgen gemäss Scrum Guide in der zweiten Phase der Sprint Planung. Architektur ist damit eine Tätigkeit, welche pro Sprint durchgeführt wird.

In Projekten, in welchen die Systemarchitektur gesetzt ist z.B. bei Software-Erweiterungen, mag dies ein valables Vorgehen sein. Es gibt aber auch Projekte, in denen eine neue Architektur geschaffen werden muss, deren Komplexität und Risiken den Einsatz erfahrener Software-Architekten und möglicherweise eine vorgängige Erstellung und Machbarkeitsabklärung einer Gesamtarchitektur erfordern.

Das „Scaled Agile Framework“ [SAF], [Le11] von Dean Leffingwell erweitert den Einsatz von Scrum auch für grosse Projekte mit komplexen Architektur- und Anforderungsfragen. Dazu wird eine separate Portfolio-Ebene eingeführt, so können Architektur- und Business-Epics über mehrere Iterationen geplant und verfolgt werden. Dabei bleibt allerdings offen, wie „design spikes“ und „value stories“ priorisiert und in den Sprints umgesetzt werden.

2.2 Anforderungsanalyse und Scrum verbinden

In manchen Projekten braucht es eine vorgängige Anforderungsanalyse, in welcher Anforderungen nicht (nur) als User Stories festgehalten werden. Ist damit der Einsatz von Scrum bei der Umsetzung nicht mehr möglich?

„Ein Projekt agil durchzuführen, bedeutet nicht, ohne vorherige Spezifikation des Gesamtsystems mit der Umsetzung zu beginnen“ [Ec10]. Eckkrammer et. al. stellen der Scrum Projektphase eine Vorprojektphase zur Anforderungsdefinition voran. Darin wird eine Risikoanalyse gemacht und nötigenfalls werden Prototypen gebaut. Das schafft die Basis für die Auftragserteilung und liefert einerseits den initialen Produkt-Backlog, andererseits aber auch eine übergeordnete Releaseplanung, Aufwand- und Kostenschätzung.

Insbesondere bei grösseren Projekten gibt es Anforderungen, welche zu gross sind, um sie in einer User Story zu fassen und in einer Iteration umzusetzen:

What happens when a story includes too many unknowns to tell just how big it is? Or what if the story's requirements are known, but its effort is too huge to complete in a single sprint? We call these stories "epics." [SAL]

Epics werden im Product-Backlog festgehalten und während der Spint-Planung in umsetzbare Stories zerlegt. Dabei macht es durchaus Sinn, die Beziehung der abgeleiteten Stories zu den ursprünglichen Epics aufrecht zu erhalten z.B. indem wichtige Anforderungen während einer frühen Konzeptionsphase in einem Konzeptpapier (Anforderungsanalyse) festgehalten werden.

Auch die Firma XING AG ergänzte das agile Vorgehensmodell mit einer übergeordneten Projektführungsschicht (Meta-Ebene) in der die selbstorganisierten Teams als Durchführungsschicht eingebunden wurden. Das Projekt wurde in drei Phasen (Vorbereitungsphase, Phase des verteilten Arbeitens, Konvergenzphase) gegliedert und eine verbindliche Grobplanung (Timebox für das Gesamtprojekt) festgelegt [KM12]. Der XING-Plattform-Relaunch stellte XING vor besondere Herausforderungen, weil die für Enhancement und Maintenance agil aufgestellte Entwicklung nun ein Grossprojekt realisieren musste.

2.3 Interdisziplinarität und Scrum verbinden

Scrum modelliert den Prozess der Software-Erstellung. Oft ist aber die Software-Erstellung nur ein Teil eines Projektes. Sei es, weil im Projekt auch Hardware entwickelt wird oder weil z.B. im Projekt anhand einer Markt-, Kosten- und Machbarkeitsanalyse vorgängig entschieden werden muss, ob das Vorhaben überhaupt umgesetzt wird. Dies bedeutet, dass Scrum in ein Gesamtverfahren einzubetten ist.

Im Kontext von technischer Produktentwicklung, wo die Softwareentwicklung ein Teil des gesamten Entwicklungsprojektes ist, hat z.B. ABB schon sehr früh aufgezeigt, wie die iterativ-inkrementelle Softwareentwicklung in das firmeninterne Gate Model von ABB integriert werden kann [Wal02], [KR06]. Dabei wird das firmeninterne Gate Model von ABB als Projektführungsschicht der iterativ-inkrementellen Softwareentwicklung als Projektdurchführungsschicht übergeordnet.

Im folgenden Kapitel versuchen wir für studentische Projekte aber auch für anwendungsorientierte Forschungsprojekte im Informatikbereich einen Rahmen zu schaffen, der einerseits von der Einfachheit und Kraft von Scrum profitiert, aber andererseits die oben aufgeführten Aspekte bezüglich Architektur und Anforderungsmanagement trotzdem unterstützt.

3 SoDa - Ein beispielhafter Lösungsansatz, angewendet in der Hochschulpraxis

Das Informatikstudium an der Hochschule Luzern – Technik & Architektur vermittelt eine fundierte Fachausbildung für eine professionelle Tätigkeit als Informatikerin oder Informatiker. Das Curriculum ist modular aufgebaut. Die Informatik-Kernausbildung vermittelt essenzielle Kompetenzen in den Bereichen Software-Entwicklung, Systemtechnik, Modellierung und IT-Projektmanagement.

Weil sich der Bereich Software-Entwicklung über das gesamte Studium erstreckt und von verschiedenen Dozentinnen und Dozenten in Modulen unterrichtet wird, die teils mehr theoretisch ausgerichtet sind, teils mehr Projektcharakter haben, haben wir uns entschlossen, ein Projekt- und Vorgehensmodell anzubieten, das den Studierenden als Klammer über alle Module im Bereich Software-Entwicklung dient.

Das Ergebnis heisst SoDa (SoDa steht für Software Development agile). Wir stellen es nachfolgend kurz vor.

3.1 Anforderungen an ein Projekt- und Vorgehensmodell

Ein Projekt- und Vorgehensmodell soll uns in der Lehre in den oben erwähnten Fachmodulen aber insbesondere auch in interdisziplinären Projektmodulen mit den Abteilungen Elektrotechnik und Maschinenbau dienen. Ausserdem soll es uns auch in der Institutsarbeit, d.h. in der angewandten Forschung und Entwicklung bei der Projektzusammenarbeit mit der Industrie unterstützen.

Die folgenden Anforderungen haben wir vor diesem Hintergrund zusammengestellt:

- **Nähe zur Industrie**
das Projekt- und Vorgehensmodell soll strukturell und terminologisch nicht zu stark vom in der Wirtschaft und Industrie praktizierten Vorgehen abweichen.
- **Aktualität**
das Projekt- und Vorgehensmodell soll zeitgemäss sein, d.h. es soll dem Stand der Technik entsprechen.
- **Anwendbarkeit**
das Projekt- und Vorgehensmodell soll bei den in Frage stehenden Projekten (Forschungsprojekte mit der Industrie, schulinterne interdisziplinäre Projekte und Lernprojekte) sinnvoll anwendbar sein.
- **Umsetzbarkeit**
das Projekt- und Vorgehensmodell soll in typischen studentischen Projekten umsetzbar sein (Anzahl und Art der Rollen, zeitliche Aspekte, beurteilbare Artefakte)

In Industrie und Wirtschaft hat Anfang des letzten Jahrzehnts – ausgehend von mittelständischen Unternehmen – schrittweise eine Abkehr von klassischen rigiden Vorgehensmodellen und eine Hinwendung zum agilen Entwickeln eingesetzt [Ju09]. Inzwischen hat sich dieser Trend fortgesetzt, dabei wird offensichtlich, dass unter den agilen Ansätzen Scrum das Rennen macht und sich als de facto Standard – wenn auch nicht immer vollständig umgesetzt – etabliert [KM12].

3.2 Randbedingungen im schulischen Umfeld

Im schulischen Umfeld dauern Projekte oft nur wenige Wochen (z.B. Lernprojekte während dem Semester) oder Monate (Bachelor Diplomarbeiten). Dabei sind typisch ein bis vier Studierende beteiligt. Die von Scrum vorgegebenen Rollen Product-Owner, Scrum Master und Entwickler (vgl. Scrum Guide [Su11]) können daher nur teilweise ausgefüllt werden.

Die absolutistische Machtstellung des Product-Owners gemäss Scrum Guide „Die Entscheidungen des Product-Owners müssen durch die gesamte Organisation respektiert werden, anderenfalls kann der Product-Owner seine Rolle nicht verantwortlich ausfüllen und auch keinen Erfolg haben.“ lässt sich schlecht mit der Rolle von Studierenden in Lern- oder Wirtschaftsprojekten bzw. Diplomarbeiten in Einklang bringen. Es sei denn der Product-Owner wird als Rolle von Dozierenden bzw. Wirtschaftspartnern wahrgenommen, was diesen nicht unbedingt zuzumuten ist.

Des Weiteren besteht von der Schule der Anspruch, dass die Studierenden sich ernsthaft mit der Architektur der zu entwickelnden Software auseinander setzen und diese, sowie die zu Grunde liegenden Architekturentscheidungen in nachvollziehbarer Form dokumentiert sein müssen. Scrum steht dazu zwar nicht direkt im Widerspruch, marginalisiert aber die Bedeutung der Architektur. Und viele (miss-)verstehen das agile Manifest so, dass „over comprehensive documentation“ dazu legitimiert, die Entwurfsdokumentation wegzulassen.

Last but not least sind schulische Projekte in der Regel zeitlich und organisatorisch in einen starr reglementierten Rahmen der Semesterplanung sowie Testat- und Abgabetermine eingebunden. In zwei oder drei, bei Diplomarbeiten vielleicht höchstens fünf Sprint müssen die Ziele erreicht sein und die zu bewertenden Ergebnisse vorliegen. Andernfalls sind Testat- und Abgabebedingungen nicht eingehalten.

3.3 PM-Rahmen als Ausweg aus dem Dilemma

Aus den obigen Gründen kann im realen schulischen Umfeld Scrum nicht „as is“, jedenfalls nicht praktisch, d.h. in Lern-, Wirtschafts- und Diplomarbeitsprojekten, eingesetzt werden. Wer also unter den oben genannten Rahmenbedingungen behauptet, „wir machen an unserer Schule Scrum“ ist nicht ehrlich und / oder nicht konsequent. An der Abteilung Informatik der Hochschule Luzern – Technik & Architektur, haben wir uns deshalb entschieden, basierend auf Scrum ein eigenes Projekt- und Vorgehensmodell aufzusetzen, das den spezifischen Ansprüchen in unserem Umfeld genügt.

Unseren Überlegungen haben wir das Projektmanagement-Modell von Jenny [Je03] zu Grunde gelegt, welches uns im Hochschul-Alltag gerade in interdisziplinären Diskussionen zu Vorgehensfragen immer wieder klärend zur Seite steht.

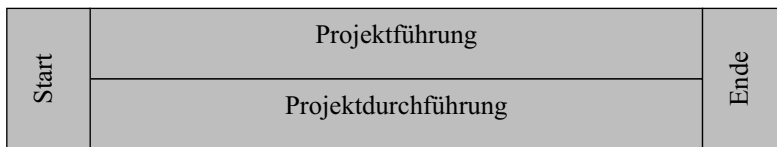


Abbildung 1: Projektmodell nach Jenny (vereinfacht)

Das Modell unterscheidet im Wesentlichen zwei unterschiedliche Ebenen. Einerseits die Projektdurchführung, welche an die jeweilige Disziplin und Projektart angepasst werden muss, und andererseits der Projektführung, welche unabhängig von disziplinären Vorgehensmodellen ist, das Gesamtprojekt umfasst und der Kommunikation und Entscheidungsfindung mit dem Management dient.

Während in der Projektführung Planung und Controlling im Vordergrund stehen, geht es bei der Projektdurchführung um Konzipierung, Realisierung und Einführung. Laut Jenny können in der Projektdurchführung abhängig von den Disziplinen und der Projektart die unterschiedlichsten Vorgehensmodelle zu Einsatz kommen (und bei Projekten mit mehreren Disziplinen auch gemischt werden).

Für unseren Hochschulalltag, aber auch für unsere Projektzusammenarbeit mit der Industrie hat sich das Projektmanagement-Modell von Jenny sehr bewährt. Es schafft einen Rahmen in dem das Projekt abgewickelt wird, dabei verpflichtet sich der Auftragnehmer (Studierende) auf inhaltliche und zeitliche Zielsetzungen, wie sie mit dem Auftraggeber (Lehrpersonen bzw. Wirtschaftspartner) vereinbart wurden – eine Grundvoraussetzung für jegliche Lern- und Wirtschaftsprjekte bzw. Diplomarbeiten.

Die Studierenden stehen damit nach dem Zwei-Ebenen-Modell von Jenny auch in der Projektführungs-Verantwortung. Es gibt eine Projektleiter-Rolle. Das Projekt hat einen definierten zeitlichen Rahmen (Semester-, Testat-, Abgabetermine) völlig losgelöst vom technischen Vorgehen. Der Projektablauf ist konventionell strukturiert, d.h. es gibt einen Projektstart mit Initialisierungsphase und ein Projektende mit einer Einführungsphase.

Auf dieser Basis kann insbesondere auch mit Wirtschaftspartnern, deren Kerngeschäft nicht die Informatik ist – sei dies ein Maschinenbauer, eine Anlagenbau-Firma oder ein Finanzdienstleister – klar und einfach kommuniziert werden.

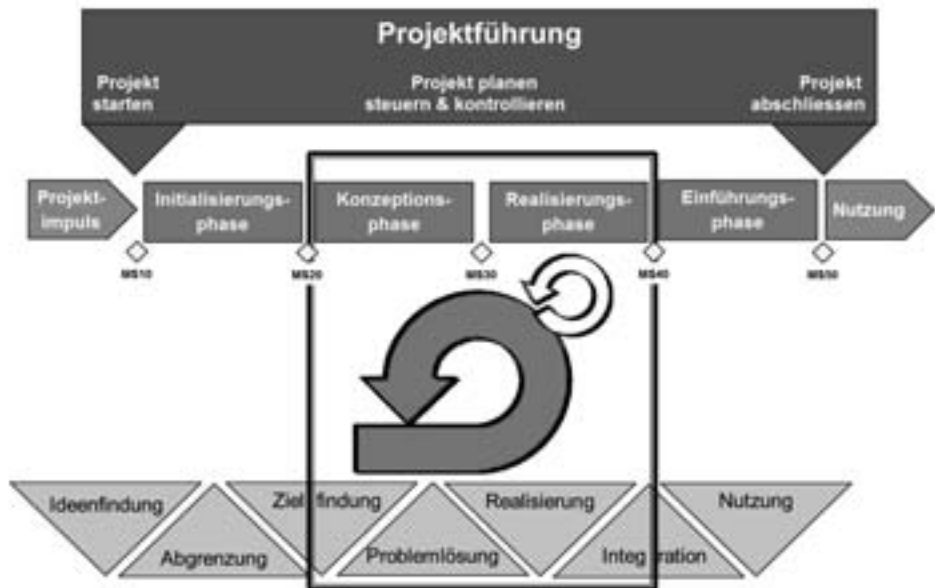


Abbildung 2: Einordnung des iterativ-inkrementellen Vorgehens (angelehnt an [Je03])

Das Projektmodell von Jenny geht explizit davon aus, dass je nach Projektart unterschiedliche Vorgehensmodelle und Charakteristika der Projektphasen anzuwenden sind. Insbesondere ist es bei Software-Entwicklungsprojekten meist nicht möglich bereits in der Initialisierungsphase die Anforderungen vollständig zu erfassen. Zudem ist die Trennung zwischen Konzeption (SW-Design) und Realisierung (Programmierung) bei der Software-Entwicklung eher künstlich.

In SoDa tritt deshalb an Stelle der konventionellen Konzeptions- und Realisierungsphasen ein iterativ-inkrementellen Vorgehen, das sich stark an Scrum anlehnt. Scrum wird hier also im Sinne eines Vorgehensmodells für einen Teil der Projektphasen innerhalb eines konventionellen Projektrahmens genutzt. Die Projektleiter-Rolle mutiert für diesen zentralen Projektabschnitt zum Scrum-Master und unterstützt als «servant leader» das Team und den Product-Owner.

3.4 Planung: lieber grob richtig als genau falsch

Nach wie vor gilt die unter dem Namen Cone of Uncertainty bekannte Beobachtung von Barry Boehm, der bereits 1981 festgestellt hatte, dass Aufwandschätzungen zu Beginn eines Projektes mit einem Unsicherheitsfaktor von mal vier / durch vier behaftet sind [Bo81]

Mit so viel Unsicherheit (25% – 400%) ganz am Anfang, kann weder der Auftraggeber noch der Auftragnehmer vernünftig leben. Ein zweistufiges Vorgehen hilft, mit dieser Situation zurecht zu kommen:

1. Rahmenplan: Der Projektleiter erarbeitet deshalb einen Rahmenplan, der auf der Ebene des Projekt-(Phasen-)Modells eine grobe Vorstellung des übergeordneten Projektablaufs mit den wichtigsten Meilensteinen und Terminen gibt.
2. Rollende Detail-Planung: Der Product-Owner priorisiert laufend die Anforderungen im Backlog nach Geschäftsnutzen. Gemeinsam mit dem Entwicklungsteam wird der Aufwand für die höchst priorisierten Anforderungen abgeschätzt und die jeweils folgenden zwei Iterationen werden detailliert geplant.

Dieses Vorgehen erlaubt, den Ablauf des Projektes schon früh grob richtig festzuhalten. Die Planungsunsicherheit wird durch die Priorisierung des Backlogs aufgefangen: hoch priorisierte Anforderungen werden garantiert umgesetzt und ausgeliefert.

Die 100% vollständigen Requirements, die 100% perfekte Spezifikation sind erst nach unendlich langer Zeit zu haben – «gut genug» ist das Ziel: eine kurze Initialisierungsphase muss genug Klarheit schaffen, um mit der Umsetzung rasch beginnen zu können. Am Ende der Initialisierungsphase, d.h. an der Schnittstelle zwischen Projektmanagement-Rahmen und iterativ-inkrementeller Entwicklung, steht einerseits der initial aufbereitete und priorisierte Product-Backlog, andererseits eine Release-Planung (denn Projektziele und Abgabetermine stehen in Lern- oder Wirtschaftsprojekten bzw. Diplomarbeiten ja fest). Der Product-Owner verteilt die Backlog Items auf die Sprints, er prüft am Ende jedes Sprints was „done“ ist, gibt das Inkrement frei und ermittelt die verbleibende Arbeit.

Dieses Modell trägt einerseits dem Bedürfnis von Auftraggebern und übergeordneter Organisation nach geordneter Projektdurchführung mit standardisierten Phasen und Meilensteinen Rechnung. Andererseits wird sichergestellt, dass die Software iterativ-inkrementell entwickelt wird und sich agil an den sich möglicherweise ändernden Bedürfnissen der Auftraggeber und an den Möglichkeiten des Entwicklungsteams orientiert.

3.5 Was du schwarz auf weiss besitzt

Bleibt noch die Frage der Bewertbarkeit und Nachvollziehbarkeit der geleisteten Projektarbeit. Ein offensichtlicher Anspruch in Schulprojekten, aber natürlich nicht nur da. David L. Parnas, der Pionier der modernen Softwaretechnik, hat einmal gesagt:

At the root of all the problems we have with software lies the failure of software developers to document design decisions in a way that allows those decisions to be reviewed, to guide the implementors, to guide the testers and to guide those who will maintain it in the future. [Pa08]

SoDa kennt zwei Kategorien von Dokumenten: die Entwurfsdokumente, welche im Parnas'schen Sinn die Software dokumentieren und die Planungsdokumente welche den Entwicklungsprozess unterstützen und dokumentieren.

Planungsdokumente gibt es einerseits auf der Projektmanagement-Ebene andererseits auf der Ebene der Sprint-Planung.

Der SoDa Projektplan lehnt sich an den IEEE Software Project Management Plan an (IEEE Std 1058-1998), allerdings im Sinne des agilen Manifests auf das Notwendigste reduziert. Die Terminplanung beschränkt sich explizit auf den Rahmenplan (siehe oben).

Der SoDa Sprintplan enthält Sprint-Ziel, das Sprint-Backlog, das jeweils zu Beginn eines Sprints vom Product-Owner mit dem Team auf Basis der Rahmenplanung und des Product-Backlogs erarbeitet wird. Das Sprint-Backlog ist eine Liste aller Items, die vom Team im Rahmen des nächsten Sprints erledigt werden müssen, um das Sprintziel zu erreichen. Dazu gehören pro Item deren „Definition of done“, eine Aufwandschätzung und falls nötig eine Zerlegung in Personen zugeordnete Tasks.

Typischerweise handelt es sich bei den Backlog-Items um User-Stories (Produkt-Funktionalität). Die Umsetzung einer User-Story umfasst neben dem Codieren und Schreiben der Unittests auch Tätigkeiten wie Requirements verfeinern und dokumentieren, Design-Entscheide ergänzen und dokumentieren, Integrations- und Systemtests planen und durchführen, Entwicklungs- und Testumgebung erweitern und pflegen, Refactoring etc. Daneben erlaubt SoDa auch Backlog-Items, welche nur Teile davon umfassen, wie z.B. die Dokumentation und Diskussion einer Architekturentscheidung.

SoDa Entwurfsdokumente, welche im Parnas'schen Sinn die Software dokumentieren, sind einerseits die Verschriftlichung von Product-Backlog und Sprint-Backlog andererseits die Systemspezifikation und der Testplan. Entsprechend dem iterativ-inkrementellen Vorgehen werden die Dokumente in der Initialisierungsphase erstellt, sind aber naturgemäss am Ende der Phase nicht vollständig und müssen deshalb bei jedem Sprint ergänzt und allenfalls korrigiert werden.

Die SoDa Systemspezifikation ist im weitesten Sinne an die Software Design Description (IEEE Std 1016-1998) angelehnt. Sie umfasst vier Abschnitte. Systemübersicht (1): Übersicht und Begründung des gewählten Lösungsansatzes. Architektur und Design-Entscheide (2): Modell(e) und Sichten, Daten (Mengengerüst / Strukturen), Entwurfsentscheide. Schnittstellen (3): Externe Schnittstellen, wichtige interne Schnittstellen, Benutzerschnittstelle(n). Environment-Anforderungen (4): Anforderungen bezüglich Hardware, Betriebssystem etc.

Ein zentrales Element der Qualitätssicherung in Scrum ist die Definition of done. Dafür erforderliche Integrations- und Systemtestfälle lassen sich oft nicht automatisieren. In SoDa wird deshalb ein Testplan in jedem Sprint mit den neu dazu gekommenen Testfall-Beschreibungen erweitert damit jederzeit reproduzierbare Regressionstests gemacht werden können. Der Testplan ist das Werkzeug des Product-Owners bei der Sprint-Abnahme.

4. Fazit

Agiles, iteratives Vorgehen ist in der Software-Entwicklung essentiell. Als Vorgehensmodell setzt sich Scrum in der Industrie zunehmend durch. Für Neuentwicklungen, in denen Requirements-Engineering, Architektur und Systementwurf ein höheres Gewicht haben, aber auch für Produktentwicklungen mit mehreren Disziplinen deckt Scrum nicht alle projektrelevanten Aspekte zufriedenstellend ab.

In manchen Projekten braucht es daher eine übergeordnete Planungsebene. Verschiedene Modell-Ansätze ordnen Scrum in ein mehrschichtiges Modell ein, es schafft Rollen und Ansatzpunkte um den in Scrum fehlenden Aspekten gerecht zu werden. Dies darf aber keineswegs dazu führen, dass Scrum in seiner Einfachheit und Kraft eingeschränkt wird. Für unser Hochschulumfeld haben wir diesen Weg mit SoDa exemplarisch umgesetzt.

Bei der Beurteilung der ersten Erfahrungen mit SoDa im Unterricht vergleichen wir das neue Modell mit dem ebenfalls an unsere Schule geschaffenen und über mehrere Jahre eingesetzten Vorgänger „HTAgil“¹, welcher sich stark am Vorgehensmodell RUP (Rational Unified Process) orientierte.

Wir können folgendes, erstes Fazit ziehen:

Auf Ebene der einzelnen Sprints (Mikroplanung) konnte eine klare Verbesserung des Vorgehens der einzelnen studentischen Gruppen erreicht werden. Die Arbeit ist besser getaktet, die Definition of done wird klarer formuliert weil adäquatere Strukturen die Aufgabe für die Studierenden greifbarer machen. Zudem ist es einfacher geworden klare Sprintziele einzufordern.

Insbesondere bei Projektbeginn macht es Sinn auch Architektur- und Infrastruktur-Fragen als Stories zu formulieren (Sichtbarkeit/Transparenz), die ebenfalls auf Sprints heruntergebrochen werden müssen. Um einen Rückfall in eine technisch getriebene Entwicklung zu verhindern, unterscheiden wir klar zwischen User-Stories und z.B. Qualitäts-&Architektur-Stories („design spikes“ und „value stories“ nach Leffingwell [Le11]). Hier werden wir weiter experimentieren, im Bewusstsein eines gewissen Konfliktes mit der Rolle des Product-Owners.

Der gleichzeitige Einsatz von Dokumenten (in welchem z.B. Kontext-Diagramm, Use-Cases und die Architektur beschrieben werden) und Stories hat sich grundsätzlich bewährt, stellt aber einen dauernden (und wohl grundsätzlichen) Drahtseilakt dar.

Die Planung auf zwei Ebenen (Projektführung und Projektdurchführung) ist für die Studierende herausfordernd, sie sind tendenziell in der Projektdurchführung verhaftet. Es ist an uns Coaches, die Ebene der Projektführung aktiv einzufordern.

Insgesamt sind wir auf gutem Weg und überzeugt, dass sich den kommenden Semestern einige noch offene Punkte klären werden.

¹ <https://wiki.enterpriselab.ch/edu/publication:htagil:htagil>

Literaturverzeichnis

- [Bo81] Boehm, B.: Software Engineering Economics. Prentice-Hall, 1981.
- [Ec10] Eckkrammer, T.; Eckkrammer, F.; Gollner, H.: Agiles IT Projektmanagement im Überblick. In (Tiemeyer, E. Hrsg.): Handbuch IT-Projektmanagement. Hanser, 2010.
- [Fr10] Friedrichsen, U.: Agil oder ingenieurmässig? Business Technology Heft Nr. 2.2010, 2010.
- [Je03] Projektmanagement – Das Wissen für eine erfolgreiche Karriere, ISBN 978-3-7281-3218-2, vdf Hochschulverlag, Zürich, 2003.
- [Ju09] Jud, M.: Kosten und Nutzen von Vorgehensmodellen – eine vergleichende Studie, ObjektSpektrum 1/2009, siehe:
http://www.sigs.de/publications/os/2009/01/jud_OS_01_09.pdf
- [KM12] Kaiser, T. ; Menden, F.: Ein Modell zur agilen Realisierung von Grossprojekten, Objekt Spektrum 2/2012. Siehe:
http://www.sigs.de/publications/Jahres_PDFs/Gesamtheft_OS_02_12_PPuH.pdf
- [KM12] Kropp, M.; Meier, A.: Swiss Agile Study 2012 - Schweizer Informatik wird agil. Swiss IT Magazin 12/2012.
- [KR06] Karlström, D.; Runeson, P.: Integrating agile software development into stage-gate managed product development. Empir Software Eng (2006) 11: 203–225, Springer Science + Business Media, 2006.
- [Le11] Leffingwell, D.: Agile Software Requirements - Lean Requirements Practices for Teams, Programs, and the Enterprise, Addison-Wesley, 2011
- [LL10] Ludewig, J.; Lichter, H.: Software Engineering – Grundlagen, Menschen, Prozesse, Techniken. dpunkt, 2010.
- [Pa08] Parnas, D. L.: Disciplined Quality Software Construction, Vorlesung an der Università della Svizzera Italiana USI, 2008.
- [SAF] Scaled Agile Framework, siehe: <http://scaledagileframework.com/>, Zugriff: 23.5.2013
- [SAL] Scrumalliance: Was ist Scrum? http://www.scrumalliance.org/pages/what_is_scrum, Zugriff: 23.5.13
- [Sc95] Schwaber, K.: Business Object Design and Implementation Workshop, OOPSLA '95 Position Paper, siehe: <http://www.jeffsutherland.org/oopsla/schwaber.html>
- [Su11] Sutherland, J.; Schwaber, K.; Scrum Guide – Der gültige Leitfaden für Scrum, Scrum.org, Oktober 2011, siehe:
<http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20DE.pdf>
- [Wal02] Wallin, C.; Ekdahl, F.; Larsson, S.: ABB, Integrating Business and Software Development Models, IEEE SOFTWARE Nov/Dec 2002.

Einführung von Scrum in einem Software-Entwicklungsprojekt der ContiTech AG

Roland Petrasch¹, Torben Franzke², Songsak Rongviriyapanich³

¹ Institute for Interactive Software and Media Engineering
Berlin, Germany

² ContiTech Conveyor Belt Group
ContiTech AG
Northeim, Germany

³ Thammasat University
Faculty of Science and Technology
Department of Computer Science
Bangkok, Thailand

Abstract: Die Einführung des agilen Ansatz nach Scrum bei der Entwicklung einer Kalkulations-Software bei der ContiTech AG wurde mit Hilfe externer Experten begleitet. Dieser Beitrag stellt einige grundlegende Überlegungen bzgl. Vorgehensmodelle vor und geht dann auf die Spezifika und besonderen Herausforderungen ein, die es im Rahmen des Projektes zu meistern galt. Im Ergebnis hat sich dabei - wenig überraschend - die Erkenntnis bestätigt, dass allein mit Scrum der Erfolg nicht garantiert ist und Scrum nur einen Teil der Aspekte eines Entwicklungsprojektes in Bezug auf den Erfolg abdecken kann. Die individuelle Ausgestaltung des agilen Ansatzes ist ebenso wichtig wie eine schrittweise Einführung.

1 Einleitung

Bei der IT-Projektierung stellte sich in diesem Praxisfall die grundlegende Frage nach dem Vorgehen bzw. dem Vorgehensmodell. Um diese Vorüberlegungen nachvollziehen zu können, greift die Einleitung die Grundsatzfrage nach der Zweckmäßigkeit von Vorgehensmodellen auf und grenzt klassische Vorgehensmodelle und Agilität gegeneinander ab. Das Kapitel 1 soll keine wissenschaftliche Analyse von Vorgehensmodellen darstellen (was den Rahmen sprengen würde), sondern aus praktischer Sicht einige Aspekte betrachten, die die Grundlage für den praktischen Einsatz von Scrum (s. Kapitel 2) darstellen. Exemplarisch seien hier das V-Modell XT als schwergewichtiges Vorgehensmodell und Scrum als agiler Projektmanagementansatz ausgewählt, um konkrete Aussagen und Beispiele vorstellen zu können.

1.1 Reifegrad- und Vorgehensmodelle: irrelevant für den Projekterfolg

Zunächst seien zwei Aussagen aus der OFFIS-Studie [BEJ06] bzgl. der Fragestellung, ob der Einsatz eines Reifegrad- oder Vorgehensmodells bestimmend sind für den Erfolg eines Projektes sind, zitiert:

- “Das Projektergebnis ist unabhängig von der Verwendung eines Reifegradmodells.“ [BEJ06, S. 280 f.]
- “Die Verwendung eines Vorgehensmodells und der Projekterfolg scheinen unabhängig.“ [BEJ06, S. 285]

Die Studie bestätigt, was wir seit dem Aufkommen der (formalen) Vorgehensmodelle in Theorie und Praxis beobachten können: Bei den Diskussionen um das passende oder „bessere“ Vorgehensmodell werden viele Meinungen und sehr wenig gesicherte Erkenntnisse ausgetauscht und die Relevanz aus praktischer Sicht ist zu oft unbeachtet geblieben. Daher sei auf die OFFIS-Studie verwiesen, in der auch andere Studien, z.B. der Chaos Report [Stan13], betrachtet aber als „nicht relevant“ eingestuft wurden. Die Gründe sind wissenschaftlich dargelegt. Daher wurde auf eine (erneute) Analyse der Studien zu diesem Thema im Rahmen dieses Beitrages verzichtet.

Vermutlich lohnt es sich nicht, eingehend auf Reifegrad- oder Vorgehensmodelle einzugehen, was am Beispiel des V-Modell XT deutlich wird: Die Quantität des Werkes ist zwar durchaus beeindruckend, aber aus praktischer Sicht taugt das Werk wohl nur als eine simple Checkliste bzw. Sammelsurium für Anregungen (Motto: “Was so alles im Projekt beachtet werden könnte ...”). Im Großen und Ganzen stehen dort selbstverständliche Dinge (um nicht Trivialitäten sagen zu müssen), z.B. dass ein Prüfer für die Prüfspezifikation verantwortlich ist. Die hauptamtliche Beschäftigung mit dem V-Modell XT ist (wohl nicht nur für Software-Entwickler) verständlicherweise keine wirkliche Herausforderung. Das allein würde jedoch nicht die fehlende Relevanz von Vorgehensmodellen in der Praxis erklären - schließlich muss wohl jeder im Projektgeschäft zuweilen auch den simplen und ungeliebten Tätigkeiten nachgehen.

Interessant ist daher die Frage, was dazu geführt haben könnte, dass sich klassische oder schwergewichtige Vorgehensmodelle wie das V-Modell XT nicht durchsetzen konnten und in der Praxis keine bzw. eigene Verfahren präferiert werden. Die o.g. Studie [BEJ06, S. 221 f.] führt auf, dass 40 % der Befragten Unternehmen gar kein Vorgehensmodell einsetzen und von den anderen 60 % hat die überwiegende Zahl ein eigenes Vorgehen. Eine weitere Studie [Ko12] bescheinigt den agilen Methoden einen deutlichen Aufschwung und eine erfolgreiche Anwendung in der Praxis. Das folgende Kapitel versucht sich an einigen praxisorientierten Reflexionen in Hinblick auf diese Tendenz.

1.2 Abgrenzung: Klassische Vorgehensmodelle und Agilität

Hier seien einige Faktoren genannt, die zu der Verbreitung von agilen Verfahren und zum Niedergang klassischer Ansätze geführt haben könnten, wobei beispielhaft auf das

V-Modell XT (als nicht-agiles und schwergewichtiges Vorgehen) und Scrum Bezug genommen wird:

Top-Down-Ansatz: Das V-Modell XT beispielsweise mit über 100 Produkten (der Begriff Dokument wird tunlichst vermieden), über 30 Rollen und entsprechend vielen Aktivitäten muss erst einmal angepasst werden – vielleicht sogar bis ins Extreme. Dieses „extreme Tailoring“ führt aber weder zu einem standardisierten Vorgehen (was noch als Vorteil bei Auftraggeber-/Auftragnehmer-Schnittstellen zu werten wäre), noch zu einem methodisch durchdachten Konzept für die Entwicklung, denn zentral wichtige Fragen, z.B. die methodische Ausgestaltung oder die Kommunikation im Team, werden hier (noch) gar nicht umfassend beantwortet, aber immerhin gibt es Methodenreferenzen. Aus praktischer Sicht stellt sich daher die Frage, warum ein ohnehin klares Vorgehen, z.B. objektorientiert und iterativ-inkrementell, damit begonnen muss, ein umfangreiches Vorgehensmodell einem Tailoring zu unterziehen, um ein Ergebnis zu erhalten, welches dann doch wieder nur zum Teil Klarheit schafft. Im Gegensatz dazu bieten sich agile Verfahren an, die zunächst wenig(er) Anpassung erforderlich machen. Ein Tailoring finden dann quasi sukzessive während der Anwendung statt (Retrospective nach jedem Sprint).

Kommunikation: Während schwergewichtige Vorgehensmodelle das Thema Kommunikation im Team praktisch völlig ausblenden, stellt Scrum beispielsweise mit dem Daily und dem Taskboard zwei einfache und auch sehr wirksame Techniken zur Verfügung, damit sich das Team abstimmen kann und sich der Status Quo der Arbeit auch „nach außen“ kommunizieren lässt. Derartige operativ wirkende und pragmatische Maßnahmen, die aus praktischer Sicht als erfolgreich angesehen werden, sind also im Vorgehen nach Scrum integriert – bei Vorgehensmodellen wie dem V-Modell XT fehlen sie.

Requirements-Engineering: Untrennbar mit dem Projekt verbunden ist das Anforderungsmanagement, wobei auch hier Scrum ein integriertes Konzept mit dem Product und Sprint Backlog sowie den Sprint Meetings vorliegt. Auch bei der Aufwänden liegt mit den Story Points oder den Ideal Man Day ein direkt anwendbares und schlüssig mit dem Vorgehen verbundenes Konzept vor. Das V-Modell XT bietet an dieser Stelle keine Lösungen an und erschöpft sich in der Nennung von Einzelmethoden im Anhang, wobei völlig offen bleibt, wie diese einzelnen methodischen Elemente in einem Gesamtverfahren zu kombinieren sind – zumal Methoden wie das Planning Poker gänzlich fehlen.

Auch wenn es also (theoretisch) möglich wäre, mit dem V-Modell XT ein iterativ-inkrementelles und vielleicht sogar agiles Verfahren durch Tailoring und Methodenzuordnung zu „erzeugen“, stellt sich ganz praktisch die Frage, wozu all der Aufwand, wenn Scrum bereits „out-of-the-box“ Verwendung finden kann.

Allerdings muss klar sein, dass es auch bei der Agilität nicht ohne Anpassungsaufwand geht, aber im Gegensatz zum V-Modell-Tailoring befindet sich der anzupassende bzw. zu erweiternde Scrum-Prozess bereits auf einer sehr praxisnahen Ebene.

1.3 Auch Agilität braucht „Tailoring“

Bei der Einführung von Scrum ist die schrittweise Ausgestaltung und Anpassung ein wichtiges Thema – eine dogmatische Sicht auf eine Anwendung von Scrum strikt nach Lehrbuch würde sich vermutlich kontraproduktiv auswirken. Aber genau hier zeigt sich wieder der Unterschied zu schwergewichtigen Verfahren wie dem V-Modell XT: Während der ScrumMaster bei der Einführung von Scrum im verteilten Team darüber nachdenken kann, wie der Daily Scrum durchführbar wird (z.B. mit Video- oder Telefonkonferenz), steht beim V-Modell XT nach einem (umfangreichen) Tailoring noch nicht einmal fest, ob es überhaupt einen solchen Daily gibt.

Wie bereits eingangs erwähnt sollten hier nur einige Aspekte aufgezeigt werden, die dazu führten, dass in diesem betrachteten Praxisfall Scrum zum Einsatz kam. Das folgende Kapitel stellt nun das Praxisprojekt näher vor, bei dem Scrum eingeführt wurde. Die Ausführungen erheben keinen Anspruch auf Allgemeingültigkeit, sondern wollen exemplarisch zeigen, dass die Scrum-Einführung alles andere als trivial ist und Anregungen für die richtige Einführungsstrategie geben.

2. Einführung von Scrum im Cobra2-Projekt

2.1 Vorstellung des Projektes

„ContiTech ist im Non-Tire-Rubber-Bereich der größte Spezialist für Kautschuk- und Kunststofftechnologie weltweit. Das Unternehmen – eine Division der Continental AG – entwickelt und produziert Funktionsteile, Komponenten und Systeme für die Automobilbranche und andere wichtige Industrien.“ [Co13]. Dazu gehört auch die ContiTech Conveyor Belt Group, die Transportbandsysteme herstellt.

Die Software „Cobra“ dient der Kostenkalkulation von Standard- und Speziallösungen im Bereich Transportbandsysteme. Die Zielgruppen sind technische Experten und der Vertriebsmitarbeiter. Cobra wurde in den letzten Jahren hausintern entwickelt und musste nun umfangreich angepasst und weiterentwickelt werden, da das System sukzessive für alle Werke nutzbar sein muss und neue Schnittstellen vorhanden sein werden, z.B. zu SAP Sales on Demand.

Für diesen Zweck wurde das Team erweitert und besteht zurzeit aus 5,5 Entwicklern, die über 3 Standorte verteilt sind (Notheim, Berlin und Szeged in Ungarn). Seit Jahresbeginn erfolgt die Einführung von Scrum, die durch einen ScrumMaster begleitet wird. Ein Mitarbeiter kümmert sich hauptamtlich um die Anforderungen und eine weitere Person ist für die Tests zuständig.

2.2 Projektorganisation

Den klassischen Product Owner gibt es nicht, was darin begründet ist, dass die infrage kommenden Personen erst noch in diese Rolle hineinwachsen müssen und sich das

Fachwissen auf mehrere Beteiligte verteilt. So kann man von sog. „Domain Product Owners“ sprechen: Experten z.B. für den anwendungstechnischen Bereich, für die vertriebsorientierten Themen und für R&D (Research and Development). Ein Mitarbeiter ist für die Koordination der Domain-Experten zuständig; er fungiert – um es klassisch auszudrücken – quasi als Gesamtprojektleiter und könnte im Scrum-Kontext als Chief Product Owner bezeichnet werden. Dieses „Product Owner Gremium“ hat noch eine Assistentenkraft, die bei der Anforderungserhebung und -analyse hilft. U.a. anhand dieser Personalbesetzung wird deutlich, welch wichtigen Stellenwert die Anforderungen haben.

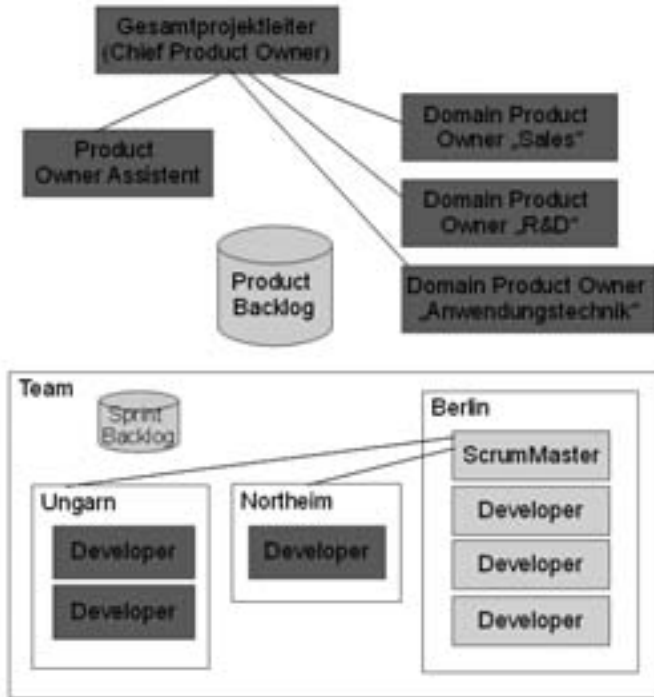


Abbildung 1: Projektorganisation

Im Bild 1 sind die dunkelgrauen Rollen von ContiTech-Mitarbeitern besetzt und die hellgrauen Rollen stellen extern besetzte Rollen dar. Die Linien deuten Hauptkoordinationsbeziehungen an, d.h. der Gesamtprojektleiter koordiniert die Beteiligten auf der Anforderungsseite und der ScrumMaster koordiniert die Entwicklung.

Einen hauptamtlichen ScrumMaster gibt es allerdings nicht: Er ist in der Übergangsphase eine Mischung aus technischem Projektleiter, Software-Architekten und ScrumMaster, was unter normalen (Scrum-)Umständen nicht zielführend wäre. Während der Einführung von Scrum ergeben sich allerdings einige Vorteile, da das Team nicht sofort mit der Autonomie allein gelassen wird – besonders wenn das Vorgehen nach Scrum noch nicht so bekannt ist, technologische bzw. engineering-orientierte Herausforderungen zu meistern sind und die Verteilung des Teams eine besondere Sorgfalt bei der Koordination erforderlich macht.

Ein besonders wichtiger Erfolgsfaktor ist die enge Kommunikation zwischen allen Beteiligten – besonders zwischen dem Chief Product Owner und dem ScrumMaster. Die Tatsache, dass beim Chief Product Owner vertiefte Kenntnisse über Software-Systeme und Entwicklungsprozesse vorliegen, ist essentiell, damit das Vorgehen abgestimmt wird, ein gemeinsames Verständnis über den Projektstatus und die -aktivitäten entsteht und die Entscheidungen fundiert erfolgen - gerade bei Themen wie Software-Architektur, Schnittstellen und der Qualitätssicherung.

2.3 Anforderungsmanagement

Die Anforderungen müssen teilweise vom bestehenden Cobra-System übernommen und überarbeitet werden oder es entstehen neue Anforderungen. Um hier eine schnelle und unbürokratische Möglichkeit der Erfassung zu finden, kommt ein Ticketsystem zum Einsatz. Jeder Stakeholder kann ein Ticket einstellen, welches zu einer Diskussion führt. Ist ein Ticket inhaltlich soweit geklärt und die sich daraus ergebenden Anforderungen klar, so findet eine Übernahmen in das Product Backlog in Form einer User Story (ggf. mit Einzelanforderungen) statt. Die Product Backlog Items sind dann bekanntermaßen wiederum die Grundlage für das Sprint Backlog.

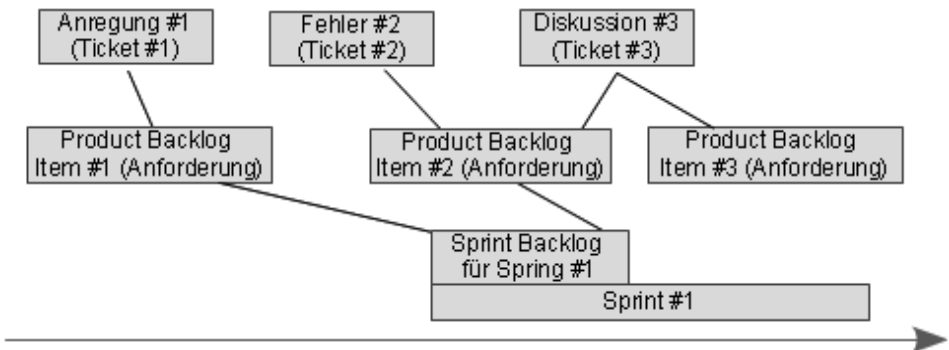


Abbildung 2: Tickets und Anforderungen Product und Sprint Backlog

Der Daily Sprint findet jeden Tag mit Hilfe eines Videokonferenzsystems statt und hat sich als außerordentlich hilfreich erwiesen. Dabei wurde in der Anfangsphase bewusst in Kauf genommen, dass die Dauer oftmals die Grenze 15 Min. weit überstieg: Wichtige grundlegende Entscheidungen, z. B. für die Software-Architektur müssen gefällt und neue Mitarbeiter in das Team integriert werden. Auch ist bei einem räumlich verteilten Teams der Austausch während des Tages vermindert. Mittlerweile normalisiert sich die Dauer: Nach 2-3 Sprints ist das Team eingespielt.

Bei der Einführung war zunächst wichtig, das Anforderungsmanagement von der Entwicklung zu trennen und einen iterativ-inkrementellen Rhythmus zu finden. Dinge wie die Aufwandsschätzung oder die „saubere“ Durchführung der Sprint Planning Meetings stehen dabei erst einmal nicht im Vordergrund. Es ist zu erwarten, dass der grundlegenden Scrum-Prozess nach einem halben Jahr stabil und im Team verankert ist.

2.4 Entwicklung

Bei der Entwicklung hat sich der Gedanke des Common Ownerships bewährt, d.h. nicht einzelne Entwickler sind zuständig und verantwortlich für (selbst) erstellte Code-Module, sondern das gesamte Team ist in der Verantwortung. Der Idealzustand, in dem jedes Teammitglied alle Bereiche der Softwareentwicklungstechnisch abdecken kann, ist allerdings nicht erreicht worden, d.h. es wurden an verschiedenen Stellen Kompromisse eingegangen.

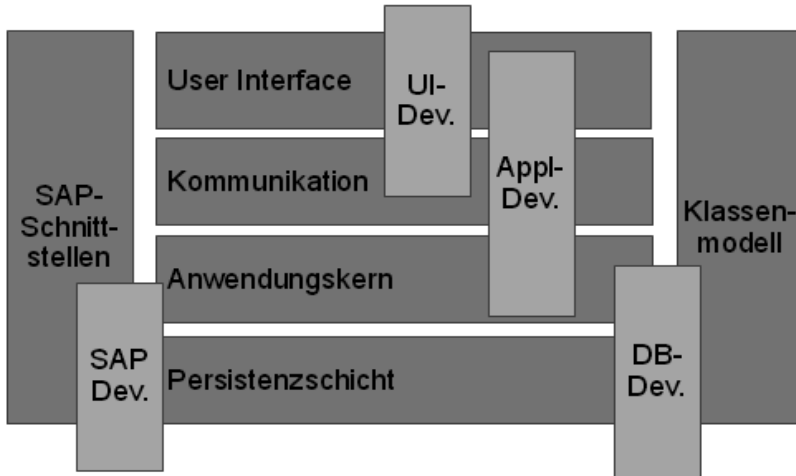


Abbildung 3: Schematische Architektur und Teammitglieder

Das o.g. Bild zeigt grob die Schichten der Software und die Entwicklertypen: Es wird zwar stets versucht, das Wissen so breit wie möglich zu streuen, aber vertiefte Kenntnisse und Erfahrungen sind zuweilen bei einzelnen Entwicklern notwendig, ohne dass eine Streuwirkung auf alle Entwickler erfolgen kann. So gibt einen Mitarbeiter, der sich eingehend mit der Persistenzschicht auseinandersetzt. Ein anderer Entwickler ist hauptsächlich mit dem User Interface betraut und pflegt den Style Guide. Auch für die SAP-Schnittstellen gibt es einen Entwickler. Die Applikationslogik wird in großen Teilen von 2 Entwicklern programmiert. Aber es wurde darauf geachtet, dass auch jenseits dieser Schwerpunkte zumindest kleinere Aufgabenpakete übernommen wurde, damit eine gewisse Breite bei der Einsetzbarkeit der Entwickler gegeben ist.

Ein weiterer wichtiger Baustein ist die Auseinandersetzung mit den Domain-Experten: Jeder Entwickler kann und soll direkt mit den Experten kommunizieren und bei der Erstellung der Product Backlog Items (Epics, User Stories, Teilanforderungen etc.) helfen. Die tiefe Kenntnis der Anwendungsdomäne – in diesem Fall die Kalkulation von Förderbändern – ist unabdingbar für die Software-Entwicklung.

2.5 Scrum-Einführungsprozess

Die sukzessive Einführung in Teilschritten vermeidet Überforderung und Überlast bei den Beteiligten und ermöglicht eine zeitnahe Justierung des Prozesses noch während der Einführungsphase. Statt eines umfassenden Prozessmodells bzw. Tailorings wurde hier eine Art Minimal-Scrum für die ersten Sprints praktiziert, so dass Erkenntnisse daraus bereits nach kurzer Zeit vorlagen.

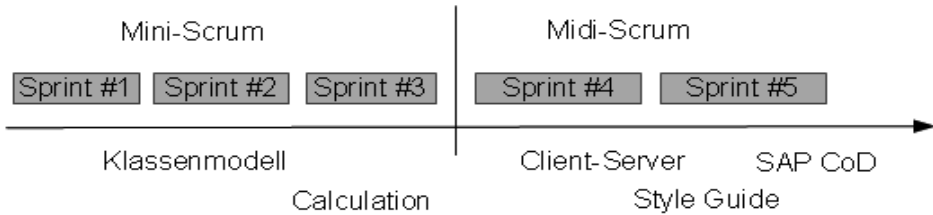


Abbildung 4: Einführung von Scrum (Mini- und Midi-Scrum)

Mini-Scrum umfasste die Aufstellung eines Sprint-Backlogs, die Sprint-Planung und die koordinierte Zusammenarbeit, die u.a. mit dem Daily erfolgt. Insofern stellte während der Einführung das Mini-Scrum den Startschuss für die iterativ-inkrementelle Projektarbeit mit der Timebox (Sprint) dar. Im unteren Bereich der Grafik sind einige wichtige entwicklungstechnische Meilensteine aufgeführt, z.B. das Klassenmodell oder das Style Guide.

Beim Midi-Scrum wurden die Anforderungen formal über das Ticketsystem in das Product Backlog übernommen und dort verfeinert. Auf der Basis von testbaren Anforderungen, dem Product und Sprint Backlog sowie dem Task Board war es möglich den Projektfortschritt tagesaktuell zu visualisieren (Burndown-Chart) und Probleme frühzeitig zu erkennen bzw. zu diskutieren. Alle Beteiligten haben dadurch ein gemeinsames Verständnis von Entwicklungsaktivitäten und -aufwänden. Allerdings wurde während der Einführungsphase auf eine formale Aufwandsschätzung verzichtet, was überraschend mag. Angesichts der Tatsache, dass sowohl technologisch als auch organisatorisch viel Veränderungen notwendig waren, z.B. die neue Client-Server-Architektur oder das Prinzip der Domain Product Owner, erschien ein vorläufiger Release-Plan sinnvoller als eine feingranulare Aufwandsschätzung auf der Ebene der User Stories.

Die Einführungsphase erstreckt sich über 3 Sprints mit Mini-Scrum und weiteren 3 Sprints für Midi-Scrum über ca. ein halbes Jahr, wobei damit die Einführung noch nicht ganz abgeschlossen ist. Weitere Aspekte, z.B. die Verbesserung des DoD (Definition of Done) oder das Review sind noch offen. Daher sollte von einem Dreivierteljahr ausgegangen werden.

Rückschläge gab es durchaus auch während der Einführungsphase. So ist es beispielsweise schwer, die Qualitätssicherung in den Entwicklungsprozess zu integrieren. Entwickler wollen Software erstellen und den Sprint erfolgreich abschließen. Das Testen in Hinblick auf den Komponenten-, Integrations- und Systemtest ist dabei oftmals eine

ungeliebte Tätigkeit. Daher wurde eine weitere Mitarbeiterin eingeplant, die zumindest auf der Systemtestebene unterstützend tätig wird.

3. Fazit und Zusammenfassung

Die Scrum-Einführung erfolgte in Stufen über einen Zeitraum von 6 Monate und ist damit noch nicht abgeschlossen. Der Gesamtprojektleiter (Chief Product Owner) verbindet die fachliche Welt der Domain-Experten (Domain Product Owner) mit den Entwicklungsaktivitäten, die vom ScrumMaster begleitet werden. Der ScrumMaster fungiert auch als Software-Architekt und berät das Team bei allen entscheidenden Fragen des Software-Systems. Die Teammitglieder handeln selbständig und kommunizieren mit den Domain-Experten. Anforderungs-Workshops, Sprint-Planungsmeetings, Daily Scrum, Review und Retrospective tragen zur intensiven Kommunikation bei. Bestimmte Aspekte wie die Aufwandsschätzung spielen zu Beginn eine untergeordnete Rolle zugunsten qualitativ hochwertiger Anforderungen und der Streuung des Wissens. Eine Schwerpunktbildung im Team ist zwar zuweilen notwendig (UI-Entwickler, Persistenz-Experte etc.); es sollte aber tendenziell zumindest versucht werden, das Wissen zu streuen.

Im Ergebnis ist der steinige Weg zu einem von allen Beteiligten akzeptierten Vorgehen dann erfolgreich, wenn die Akteure persönlich und fachlich überzeugen und die postulierten Methoden und Techniken selbst vorleben, z.B. aktive Beteiligung beim Daily Scrum. Externe Prozessberater, die die erdachten Konzepte nicht auch selbst einführen und praktizieren, sind in diesem Kontext eher kontraproduktiv. Die Einführung eines Vorgehens nach Scrum benötigt je nach Vorkenntnissen der Beteiligten und je nach Projektart und Umfeld mindestens sechs Sprints und sollte mit externer Unterstützung auch auf der operativen Ebene erfolgen.

Abschließend sei kurz die „Grundsatzfrage“ wieder aufgegriffen: Scrum hat seinen Schwerpunkt eher auf dem operativen Projektmanagement, der Integration des Requirements-Managements in das iterativ-inkrementelle Vorgehen und der Kommunikation (genau diese Aspekte sind jedoch bei schwergewichtigen Vorgehensmodellen wie dem V-Modell XT nicht oder nur marginal ausgestaltet). Wenn es „eine Abhängigkeit zwischen dem Aufwand für Kundenkommunikation und dem Projektergebnis“ [BEJ06, S. 258] gibt und ein „Zusammenhang zwischen Teamkommunikation und Projektergebnis“ [BEJ06, S. 270] existiert, dann macht der Einsatz von Scrum sinn. Auch beim Thema „Abhängigkeit zwischen der Motivation des Teams und dem Projektergebnis“ [BEJ06, S. 266] ist zu überlegen, ob durch agile Verfahren wie Scrum und die damit verbundene höher Autonomie des Teams die Motivation steigt und damit das Projektergebnis positiv beeinflusst. Das „Projektergebnis ist abhängig von der Projektkontrolle“ [BEJ06, S. 278], so dass einem einfachen, aber wirksamen Kontrollmechanismus eine wichtige Rolle zuteil wird: In Scrum gilt das Burndown-Chart als zentrales Instrument dafür: Der Fortschritt des Projektes wird anhand der implementierten User Stories verdichtet und visuell dargestellt. Insofern ist Scrum als Projektmanagement-Framework mit integriertem Requirements-Management zu bezeichnen (und nicht als Vorgehensmodell).

Fazit: Die Beschäftigung mit Reifegrad- und Vorgehensmodellen mag für manche Menschen aus theoretischer Sicht interessant sein, relevant für die Praxis ist dies wohl eher nicht – zumindest noch nicht. Projekt- und Anforderungsmanagement sowie Kommunikation im Projekt hingegen sind essentielle „Bausteine“ für den Projekterfolg. Scrum greift diese Aspekte auf und fordert zur individuellen Ausgestaltung und Anpassung auf. Gleiches gilt für die Motivation und das Knowhow der Projektmitglieder: Hier hilft Scrum durch kurze Zyklen der Erfolgskontrolle frühzeitig Schwachstellen aufzudecken. Nicht zu vergessen: „Das Projektergebnis ist abhängig vom Komplexitätsgrad der zu entwickelnden Hard- und/oder Software“ [BEJ06, S. 246 f.] – daher sollte der Fokus bei Software-Entwicklungsvorhaben auch darauf liegen, wie das auszuliefernde Produkt zustande kommt: Gutes Software-Engineering ist die erste Wahl in Bezug auf die Komplexitätsbewältigung und damit das Projektergebnis.

Literaturverzeichnis

- [BEJ06] Buschermöhle, R.; Eekhoff, H.; Josko, B.: SUCCES - Erfolgs- und Misserfolgsk Faktoren bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland 2006. Report: VSEK/55/D, Oldenburger Forschungs- und Entwicklungsinstituts für Informatik Werkzeuge und -Systeme (OFFIS), Version: 1.1, 28.09.2006.
- [Co13] ContiTech AG, <http://www.contitech.de>, 2013.
- [Ko12] Komus, A.: Status Quo Agile. Studie zur Verbreitung und Nutzen agiler Methoden. Hochschule Koblenz, 2012.
- [Stan13] The Standish Group: <http://blog.standishgroup.com/>

Teil III

Eingeladene Beiträge der Session „Future Track“

Human Factors and SCRUM

Uwe Henker

medatiXX GmbH & Co. KG
Im Kappelhof 1
65343 Eltville/Rhein
u.henker@medatixx.de

Abstract: Nachfolgend werden die Einführung von agiler Softwareentwicklung und die Erfahrungen innerhalb einer Abteilung mit ca. 100 Mitarbeitern mit dieser Methode über einen Zeitraum von fünf Jahren beschrieben. Dabei wird besonderer Wert auf die menschliche Seite der Veränderungen durch die Umstellungen gelegt. Es wird gezeigt, welche Gratwanderung ein solcher Umbruch sein kann, mit welchem Widerstand innerhalb und außerhalb des Bereiches umzugehen war.

An Beispielen wird gezeigt, wie Transparenz und Offenheit die Veränderungen im Unternehmen unterstützen können und wie wichtig die Mitarbeit der Beteiligten an der Gestaltung des Prozesses ist.

1 Die Einführung von SCRUM

Das Vertrauen in unsere Software und die Leistungen der Entwicklungsabteilung hätten vor etwa sechs Jahren kaum schlechter sein können. Auch das Vertrauen der Mitarbeiter in die eigene Leistung war zerstört. Mit viel Mühe und Frust wurde alle drei Monate eine neue Software-Version erstellt und an die Kunden verschickt. Fast jedes Mal war ein hohes Maß an Nacharbeit nötig, um ein Mindestmaß an Stabilität herzustellen.

Es musste etwas geschehen, um die Zufriedenheit der Kunden wiederherzustellen, die Motivation der Mitarbeiter wieder zu stärken und letzten Endes den Fortbestand der Firma zu sichern. Dem allgemeinen Trend folgend, wurde SCRUM als Vorgehensmodell eingeführt. Daran war zuerst ein Standort mit etwa 50 Mitarbeitern beteiligt. Schnell wurde klar, dass erste Erfolge nur eine neue Sicht auf weitere Probleme ergaben.

2 Die Mitarbeiter

Die Mitarbeiter waren es gewohnt, nach unrealistischen Plänen zu arbeiten und ertrugen mit viel Frust und Resignation alle Vorgaben. Die Qualität der Software war ein direkter Indikator der Stimmung in der Firma. Wahrscheinlich war jeder Einzelne froh über die zu erwartenden Änderungen, auch wenn bei weitem nicht jeder wusste, was ihn erwartet. Vereinzelt waren Nervosität und Angst vor den Veränderungen zu spüren.

Schnell wurde klar, dass die erwartete Selbstorganisation und Eigenverantwortung sich nicht von allein einstellte. Viele Trainings, Diskussionen waren nötig – aber auch erster Frust machte sich breit, dass sich die erwartete heile, schöne Welt nicht von allein einstellte. Vieles erinnerte an eine Umstellung auf gesunde Ernährung. Jeder weiß, wie es geht, und trotzdem ist das Einhalten der neuen Gewohnheiten mit viel Disziplin und Ehrgeiz verbunden.

Trotz vieler Änderungen und neuer Abläufe stellten sich – schneller als erwartet – Verbesserungen, insbesondere bei der Qualität der Software, ein. Ebenfalls sehr schnell identifizierten sich die Mitarbeiter mit ihren neuen Teams. Doch gleichzeitig wurde klar, dass jetzt auch sie persönlich gefordert waren, Veränderungen und weitere Verbesserungen einzuführen. Diese Erkenntnis, verbunden mit vielen Fragen und noch mehr Missverständnissen, sollte die Mitarbeiter die nächsten Jahre immer wieder beschäftigen.

Zusätzlich zu den Sprintretrospektiven der einzelnen Teams wurde mit gemeinsamen Retrospektiven einmal im Quartal begonnen. Hier wurde immer wieder auf die Selbstorganisation und die Möglichkeit, aber auch auf die Verantwortung jedes Einzelnen bei der Gestaltung des neuen Vorgehens hingewiesen. Insbesondere diese quartalsweisen Retrospektiven haben sich zu einem mächtigen Werkzeug bei der Umgestaltung erwiesen.

3 Transparenz

Die vielen Veränderungen haben die Leistungsfähigkeit, die Qualität und die Zufriedenheit der Mitarbeiter erhöht. Gleichzeitig sind jedoch Ruhephasen im Entwicklungsprozess verschwunden. Viele öffentlich dargestellten Informationen erhöhen die Transparenz und damit auch den Druck auf die Mitarbeiter. Außerdem machen sie Schwächen des Bereiches öffentlich, was mögliche Konflikte mit anderen Bereichen ehrlicher, aber auch herausfordernder in der Auseinandersetzung gestaltet. Dies zeigt, welche Gratwanderung Transparenz sein kann. Jedoch führt sie auf ein höheres Level in der Zusammenarbeit und ebenfalls zu besserer Qualität und Verantwortung für die eigene Arbeit.

Ohne Transparenz sind die vielen nötigen Veränderungen und deren Nachhaltigkeit nicht umzusetzen.

4 Diskussion

Insbesondere die Transparenz und die Delegation von Verantwortung in die Teams verstärken die Diskussion im Unternehmen über die Softwareentwicklung. Viele Experten geben Ratschläge und erkennen Fehler. Mit diesen Experten muss man sich auseinandersetzen, was aber auch zu neuen Ideen und zu einem neuen Umgang miteinander führen kann. Darin stecken Chancen, jedoch auch Risiken, die am Beginn unterschätzt wurden, inzwischen aber einen Wert darstellen.

5 Zusammenfassung

Inzwischen sind seit der Einführung agiler Entwicklung, insbesondere SCRUM, fünf Jahre vergangen. Es zeigte sich, dass die ursprünglich bekannten Probleme am ehesten mit schmutzigen Gardinen zu vergleichen sind. Diese wurden gewaschen und lassen heute deutlich mehr Licht in den Raum. Nachdem nun auch die Fenster geputzt wurden, sind jetzt Flecken im Raum, an den Wänden und auf dem Boden zu erkennen.

Unsere Mitarbeiter haben sich in den Jahren zu einer deutlich besseren „Putzkolonne“ entwickelt, die an vielen Stellen neuen Schmutz entdeckt und selbstständig beseitigt. Die Erwartungen an jeden Einzelnen und die Verantwortung der Teams werden, dank vieler erfolgreich durchgeführter Retrospektiven, deutlicher. Transparenz ist an vielen Stellen unverzichtbar geworden und wird immer mehr auch von den Mitarbeitern eingefordert.

Das Vorgehensmodell führt nicht nur zu besserer Qualität, zu höherer Leistung und zu zufriedeneren Mitarbeitern, sondern erhöht ebenfalls die Chancen der Firma bei der Suche nach neuen, engagierten und gut ausgebildeten Mitarbeitern. Ein Weg zurück ist heute für viele Mitarbeiter nicht mehr vorstellbar.

Was ist dran an Agilen Mythen?

Jane Trümner

Immobilien Scout GmbH
Andreasstr. 10
10243 Berlin
jane.truemner@immobilienscout24.de

Abstract: In Zeiten immer dynamischer und komplexer werdender Märkte kommt an agilen Methoden kein Unternehmen mehr vorbei. Heute entwickelt weltweit die Mehrheit von Unternehmen damit Produkte und Unternehmensapplikationen, die wirklichen Mehrwert für ihre Kunden bringen. Doch wie bei jeder neuen Methode sind Skeptiker nicht weit und es ranken sich hartnäckig Mythen darum, z.B. ...

- „Agile“ ist nur für die IT
- Agile Methoden haben zu unflexible Regeln
- Agile Teams sind „Black Boxes“
- Planung ist vollkommen unmöglich.
- Agile Methoden schaffen Unruhe und Probleme
- Mit „Agile“ braucht es kein Management mehr

Solche Mythen werden in diesem Vortrag beleuchtet, Antworten darauf gegeben und Lösungsansätze aufgezeigt. Diese Ansätze basieren auf der fünfjährigen, praktischen Erfahrung in agiler Produkt- und Organisationsentwicklung mit 28 Teams bei ImmobilienScout24.

1 Agile Methoden als Antwort auf Marktanforderungen

In Zeiten immer dynamischer und komplexer werdender Märkte kommt an agilen Methoden kein Unternehmen mehr vorbei, um flexibel auf Veränderungen zu reagieren, Kundenbedürfnisse zu befriedigen und schnelles Kundenfeedback zu bekommen. Heute entwickelt weltweit die Mehrheit von Unternehmen damit Produkte und Unternehmensapplikationen, die wirklichen Mehrwert für ihre Kunden bringen. Doch wie bei jeder neuen Methode sind die Skeptiker nicht weit und es ranken sich hartnäckig Mythen darum.

2 Agile Mythen – Antworten & Lösungsansätze

2.1 „Agile“ ist nur für die IT

Wie auch viele andere Unternehmen haben auch wir mit Einführung von Scrum, der am häufigsten eingesetzten agilen Methode in der IT-Abteilung gestartet. Hauptziele dabei war den Output der IT zu erhöhen und die Time-to-Market signifikant zu verringern.

Aber Scrum kann nicht alleine in der IT überleben, denn die Methode zielt darauf ab, mit jeder Iteration Kundennutzen zu liefern. Dazu muss ein Team seine Kunden sehr gut kennen sowie priorisieren, was wirklich wichtig ist und diese Funktionalitäten dann entwickeln. Dafür braucht es echte crossfunktionale, co-located Teams aus Product Owner, Designer, Entwickler, Scrum Master, Betriebsexperten, Quality Engineers, usw. Diese übernehmen gemeinsam die Verantwortung für ein Produkt – von der Idee bis zum Livebetrieb neuer Funktionalitäten.

2.2 Agile Methoden haben zu unflexible Regeln

Das oberste Ziel crossfunktionaler agilen Teams ist es, mit jeder Iteration Value für die Kunden zu liefern. Dazu braucht es v.a. zwei Dinge: ungestörte Iterationen, in denen sich das Team auf die Entwicklung neuer Funktionen fokussieren kann und ein enges, selbstorganisiertes Zusammenspiel von Product Owner, Team und Scrum Master. Diese Rollenteilung, die Übernahme von Produktverantwortung durch dieses Dreigestirn sowie die angestrebte Selbstorganisation gepaart mit strengen Regeln für eine Iteration (Sprint) vermitteln oft den Eindruck, Teams sind „Black Boxes“ und zu unflexibel.

Hier ist Umdenken nötig – sowohl von der Organisation als auch von den Teams.

Scrum ist eine Methode mit engen Regeln für die einzelnen Sprints, mit kurzen Feedbackzyklen, „bottom-up“ Engagement und Disziplin, um höhere Produktivität zu erzielen. Die Sprintregeln stellen sicher, dass das Team fokussiert arbeiten kann und so qualitativ hochwertige Arbeit für die Kunden liefert. Gegensätzlich zur häufigen Kritik, ist Change ein gewollter Aspekt der agilen Entwicklung, aber nicht mittendrin in einem laufenden Sprint. Die meisten Dinge können diese kurze Zeit warten und das Team hat die Chance, vor dem Start der Entwicklung das Problem zu verstehen, sinnvolle Lösungsansätze zu entwickeln und durch möglichen Wissensaufbau Risiken zu minimieren. Durch das direkte Kundenfeedback nach jedem Sprint lernt das Team iterativ und kann Funktionalitäten zeitnah verbessern.

2.3 Agile Teams sind „Black Boxes“

Erfolgreiche, reife agile Teams stellen von sich aus die Transparenz ihrer Arbeit sicher, z.B. mittels ihrer physischen und elektronischen Boards, regelmäßigen übergreifenden Reviews usw. Eine Herausforderung ist dabei das Wachstum. Je mehr Teams „agile“ arbeiten, bei uns mittlerweile 28, desto mehr müssen wir den Überblick behalten, die Inter-Team-Kommunikation fördern und gemeinsame Standards einhalten. Für den

Überblick nutzen wir gemeinsame Boards und Tools (Jira, Confluence, ...). Die übergreifende Kommunikation und die gemeinsamen Standards forcieren wir über „Communities of Practise“.

Die agile Entwicklung gibt den Mitarbeitern mehr Freiheit, aber auch mehr Verantwortung. Das motiviert. Unsere Teams gehen heute viel mutiger technische Probleme an, deren Lösung großen Kundennutzen bringt. Darauf sind wir stolz.

2.4 Planung ist vollkommen unmöglich

Auch in „Agile“ wird geplant, aber anders. Beim Einsatz agiler Methoden wird auf die vollumfängliche Upfront-Planung verzichtet, weil diese in der Realität selten lange Bestand hat. Deshalb wird in der agilen Welt öfter und auf verschiedenen Levels geplant, um die Probleme und Anforderungen zu durchdringen, Verständnis dafür zu entwickeln und Risiken zu minimieren. Dabei liegt der Fokus auf "just in time and just enough". Statt monatelang zu designen und zu planen, wird eine Vision erarbeitet und Funktionen, die den höchsten Wert für den Kunden liefern als erstes und intensiv durchdacht, geplant, umgesetzt und Feedback von den Kunden eingeholt. Funktionen hingegen, die noch in weiter Ferne liegen werden nur grob geplant, um ggf. einen Einfluss auf das Design oder die Architektur zu berücksichtigen. Dadurch ist es auch leichter möglich, kurzfristig Erkenntnisse aus dem Kundenfeedback in die laufende Entwicklung einfließen zu lassen.

2.5 Agile Methoden schaffen Unruhe und Probleme

Agiles Arbeiten ist nicht die Lösung aller Probleme sondern macht organisatorische und technische Probleme durch kontinuierliches „Inspect & Adapt“ schmerzhaft sichtbar. Da gilt es, die Augen nicht davor zu verschließen, sondern diese Themen gezielt anzugehen. Und das liegt in der gemeinsamen Verantwortung von Management und Teams.

Es ist ein Irrglaube, dass der agile Veränderungsprozess irgendwann aufhört. Es gilt immer wieder auf allen Ebenen zu hinterfragen, was das Unternehmen am Wachsen blockiert und welche Schritte ausprobiert werden, um besser zu werden.

2.6 Mit „Agile“ braucht es kein Management mehr

Nein, gerade in agilen Unternehmen braucht es ein starkes Management, denn dieses schafft genau die Rahmenbedingungen, die selbstorganisierte Teams brauchen. Das sind u.a. Entwickeln und Kommunizieren einer motivierenden Vision, das Ableiten von Zielen, das Setzen von Grenzen, die Verfügungsstellung notwendiger Ressourcen und das Zusammenbringen der verschiedenen crossfunktionalen Teams. Dabei setzt das agile Management nicht auf „Command-and-Control“ der Hierarchie sondern entwickelt ein Umfeld, in dem offen, respektvoll, mutig, fokussiert und committed gearbeitet werden kann.

Wenn wir auf unser Unternehmen schauen, gibt es besondere Herausforderungen bei der Skalierung, also dem Zusammenspiel der 28 Teams und damit ein wichtiges Betätigungsfeld für das Management. Teams können nur bis zu dem Punkt wachsen, welchen der Rahmen im Unternehmen hergibt. Themen bei der Skalierung sind:

- Wie skalieren wir das, was wir tun? Welche Ziele verfolgen wir und wie sieht damit das Produkt- und Projekt-Portfolio aus?
- Wie behalten wir über alle Teams Transparenz, wissen wo sie stehen, wie sie vorankommen, wo sie geblockt sind und wie die Qualität aussieht?
- Wie gestalten wir individuelles und kollektives Lernen, die Vertiefung von Skills und Methoden, z.B. agile Engineering Practises?
- Wie rollen wir Veränderung auf alle aus?

Diese Aspekte zeigen die Wichtigkeit von sehr guten Managern im agilen Umfeld.

Die Antworten auf die agilen Mythen und mögliche Lösungsansätze für die Überwindung dieser basieren auf der fünfjährigen, praktischen Erfahrung in agiler Produkt- und Organisationsentwicklung mit 28 Teams bei ImmobilienScout24, dem erfolgreichsten Immobilienportal. Das Unternehmen hat es mit der Anwendung agiler Methoden geschafft Output, Qualität und Mitarbeiterzufriedenheit signifikant zu steigern.

Warum Projekte scheitern

Rolf Voller

Trivadis AG
Elisabethenanlage 9
4051 Basel
rolf.voller@trivadis.com

Abstract: Im IT-Umfeld scheitern mehr als 60% der angegangenen Projekte. Und dies, obwohl es stets weiterentwickelte Vorgehensmodelle für Projektmanagement gibt und auch die Anzahl von zertifizierten Projektmanagern seit 2003 deutlich steigt. Haben Vorgehensmodelle zu viele „Nebenwirkungen“? Wäre der Einsatz von mehr „gesundem Menschenverstand“ anstelle vorgefertigter Prozesse erfolgversprechender?

Der Autor ist seit mehr als 20 Jahren im Projektmanagement tätig und stellt im Beitrag Vorschläge aus der eigenen Erfahrung zur Verbesserung der Ergebnissituation von IT-Projekten vor. Dabei werden sowohl Beispiele aus agilem Projektmanagement, klassischen Methoden und dem Mix aus beiden Welten aufgezeigt.

1 Einführung

Trotz einer Vielzahl von Projektmanagement-Methoden und genügend Projektmanagern am Markt, die für mindestens eine dieser Methoden zertifiziert sind, scheitern im IT-Umfeld über 60% der initiierten Projekte.

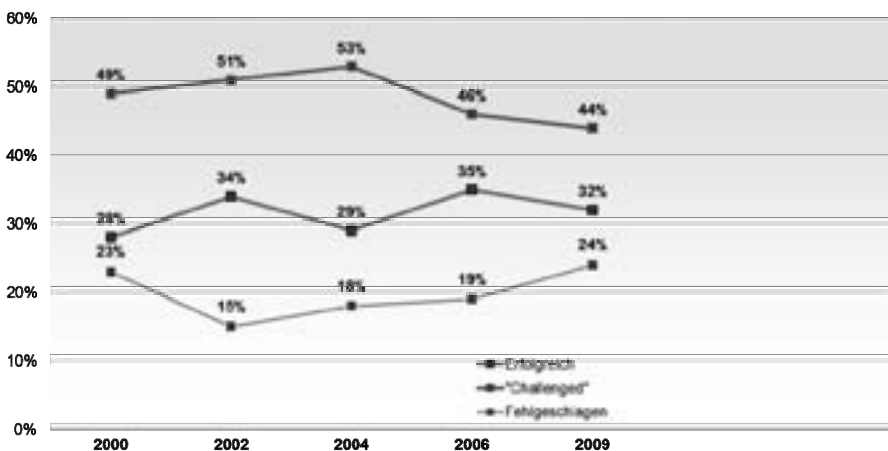


Abbildung 1: Projektausgang nach den CHAOS-Reports [St09]

Projektteam, Projektmanagement, Zielgruppe und Senior Management müssen an einem Strang ziehen, aktiv involviert sein und regelmäßig zusammenkommen (virtuell oder real), um das Scheitern eines Projekts zu verhindern.

Das klassische Face-to-Face Meeting hat nicht ausgedient, allerdings müssen Projektbeteiligte heutzutage auch in der Lage sein, Kommunikation über elektronische Hilfsmittel ebenso effizient und zielführend einzusetzen.

Die Erkenntnis, dass nicht jede Aufgabe zwangsläufig ein Projekt werden muß, dass Projekte unterschiedlich priorisiert werden müssen, dass Rahmenbedingungen über den gesamten Projektverlauf möglichst unverändert beibehalten werden sollten und dass jederzeit für jeden Projektbeteiligten der Fertigstellungsgrad seiner Teilaufgabe einsehbar sein muß, hat sich noch nicht überall durchgesetzt.

2 Vorgehensmodelle als Allheilmittel?

Der Einsatz von Vorgehensmodellen für Projektmanagement kann positive Auswirkungen auf den Projektverlauf haben. Insbesondere dann, wenn der Projektmanager es versteht, je nach Projektsituation tiefer oder weniger tief in definierte Prozesse innerhalb der Vorgehensmodelle einzutauchen.

Denn Vorgehensmodelle und die darin beschriebenen Prozesse generieren bei falscher „Dosierung“ schon auch mal „Projekt-Roboter“, Projektmanager folgen dann stur dem in den Prozessen festgelegten Ablauf.

Ein damit einhergehendes Phänomen ist die „Zertifikatsgläubigkeit“ von Unternehmen. Firmen gehen allzu gerne davon aus, dass der Toolset dem Projektmanager die Fähigkeit verleiht Entscheidungen zu treffen, korrekte Priorisierungen vorzunehmen, Risk & Issue Register zu verwalten und schlussendlich das Projekt mit gutem Ausgang zu Ende zu bringen.

3 Lösungsansätze

Der Autor hat die Erfahrung gemacht, dass Vorgehensmodelle dann sehr hilfreich sind, wenn der Projektmanager über die Fähigkeit verfügt, die Eindringtiefe in die jeweilige Methodik aus der Erfahrung heraus festzulegen. „Welches Werkzeug verwende ich, welchen Prozessen folge ich in einer bestimmten Projektsituation?“ sind Entscheidungen, die man schwerlich in Zertifikatsschulungen erlernen kann.

So werden sich erfahrene, gute Projektmanager nicht hinter Vorgehensmodell-Prozessen verstecken, sondern eine unternehmerische, kreative Denkweise an den Tag legen, wenn es um die schnellstmögliche Lösung eines Konflikts oder die Herabstufung eines Risikos geht.

Der Autor empfiehlt zudem die Vermischung von Vorgehensmodellen. Insbesondere die Kombination von SCRUM (einer neueren, agilen Projektmanagement-Methode, die ohne die Rolle des Projektmanagers auskommt) mit PRINCE2, einer „alteingesessenen“, immer noch schlanken Methode mit vorhandenen klassischen Projektinstanzen wurde als sehr effektiv und flexibel für fast jede Unternehmenskultur erlebt.

So paßt der Phasenansatz von PRINCE2 gut zum iterativen Vorgehen von SCRUM. Beide Verfahren können als „produktorientiert“ bezeichnet werden.

Grosse Unterschiede gibt es dagegen im Bereich Projekt-Organisation. Hier stehen sich ein extrem schlankes SCRUM-Gebilde, bestehend aus gerade mal 3 Rollen (SCRUM-Master, Product Owner und Team) und ein deutlich umfangreicheres, auch mehrere Geschäftsebenen abdeckendes, prozessorientiertes Konstrukt bei PRINCE2 gegenüber. Beide „Gesichter“ dieser Mischform können im Projektverlauf sehr gut vor den jeweils passenden Projektbeteiligten zum Einsatz kommen.

Im Vortrag berichtet der Autor hierzu aus selbst geleiteten Projekten und zeigt unterstützend Auswertungen einer Analyse der Standish Group [St09] zu Erfolgs- bzw. Misserfolgsfaktoren in Projekten auf. Der Beitrag endet mit der Projektion zukünftiger Arbeitsweltmodelle [DI11] auf die Tätigkeit von Projekt-Teams.

Literaturverzeichnis

- [DI11] Dell & Intel: The Workforce Perspective, Report #2, 2011.
- [St09] Standish Group: The Chaos Reports, 2009.

Jenseits der Vorgehensmodelle - Projektteams wirksam in Organisationen verankern, das Viable System Model und MAP

Jan Wehinger

Mieschke Hofmann und Partner
Gesellschaft für Management- und IT-Beratung mbH
A Porsche Company
Office Wolfsburg
Major-Hirst-Straße 5 | Autovision ICW Gebäude 7
D-38442 Wolfsburg
jan.wehinger@mhp.com

Abstract: Es existiert eine Vielzahl an Vorgehensmodellen um Projekterfolge zu sichern oder augenscheinlich zu ermöglichen. Tatsache ist jedoch, dass sich keines dieser Vorgehensmodelle nachhaltig in der Praxis bis heute durchgesetzt hat. Vielmehr ist in Organisationen häufig zu beobachten, dass Projekte trotz der angewandten Vorgehensmodelle in Schieflage geraten. Daher gilt es jenseits der Vorgehensmodelle eine Organisation zu befähigen, nicht den Projektverlauf über eine Modell vorzugeben, sondern sowohl das Projektteam wirksam zusammenzusetzen (MAP) als auch dieses Projektteam wirksam in der Organisation zu verankern (VSM) und damit auf Basis einer wirkungsvollen, eigenverantwortlichen Projektorganisation die Projektziele zu erreichen.

1 Einleitung

Zwar sind die Herausforderungen innerhalb von Projektmanagementthemen in vielen Unternehmen ähnlich, die jeweiligen Organisationsformen der Unternehmen jedoch recht heterogen. Daher bedarf eines organisationalen Metarahmens, der eine hinreichende Isomorphie mit der Realität bietet und als allgemeines Strukturmodell für Projekte und deren umgebende Unternehmensorganisation dienen kann. Dafür bietet sich das Viable System Model an.

2 Viable System Model

Das Viable System Model (VSM), oder Strukturmodell lebensfähiger Systeme, wurde in seiner Ursprungsform von Stafford Beer [Be79], [Be81] auf Basis der Ausführungen von [As70, Fo62, Pa61] erdacht. Es ist bis heute das weltweit anerkannt am weitesten entwickelte systemisch-kybernetische Modell zum Abbilden von sozialen Systemen. Das VSM basiert auf dem Postulat der Lebensfähigkeit, das nach Beer dann erfüllt ist, wenn folgende fünf Systeme vorhanden sind und miteinander interagieren:

- System 1: die operativen wertschöpfenden Subsysteme
- System 2: Koordinationsfunktion der Systeme 1
- System 3: operatives Management
- System 3*: Audit- und Monitoringaufgaben
- System 4: Strategisches Management und Umweltinteraktion
- System 5: Normatives Management des Gesamtsystems

Jedes Defizit in dieser Grundstruktur führt zu Einschränkung der Systemfunktion und mindert die Lebensfähigkeit des Gesamtsystems. Vertiefend zum VSM sei u.a. auf die Originalliteratur von Stafford Beer sowie auf [Sc00], [We12] verwiesen.

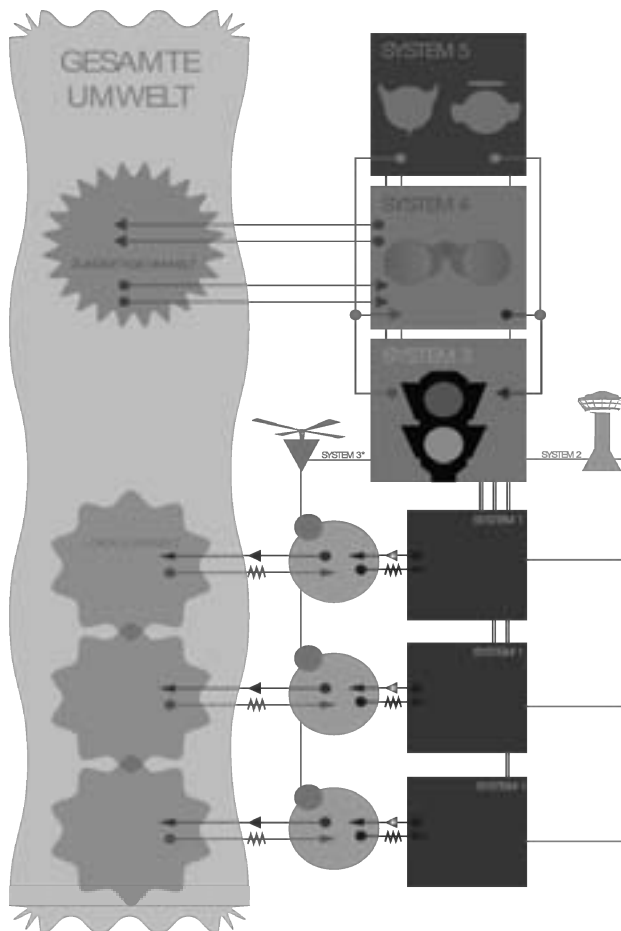


Abbildung 1: VSM Darstellung mit erweiterter Symbolsprache [We12]

3 Integration von MAP ins VSM

In [Ha12] wurde ausführlich hergeleitet, wie sich der MAP-Ansatz (siehe [Ha10]) für Projektteams in das VSM integriert/ integrieren lässt. Mit dem Viable System Model (VSM) von Stafford Beer steht nach [Ha12] somit ein geeignetes neurokybernetisches Modell zur Verfügung, dass das Projektteam „systemisch“ beschreibt und in die höheren Rekursionsebenen des Unternehmens integriert. [Ha12] stellt fest, dass die MAP-Teams eines Unternehmens sich wie andere agile Projektteams in das Unternehmen eingliedern und in der detaillierten Ausgestaltung in tiefere Teamebenen ebenfalls die VSM-Postulate erfüllen (siehe hierzu auch [HW12]).

4 Grenzen klassischer Vorgehensmodelle

Das Wirkfeld klassischer Vorgehensmodelle ist insbesondere auf Arbeitsabläufe in einem Projektteam bzw. auf die Gremien- und Kommunikationsstruktur der Projektleitung fokussiert. Dies ist in Abbildung 2 dargestellt. Es bedarf in Organisationen gerade im Bereich nicht linear planbarer Projekte, komplexen Aufgabenstellungen oder agilen Vorgehensweisen, Ansätze um das System Unternehmen auf unterschiedlichen Ebenen zu befähigen mit der jeweiligen Projektlandschaft umgehen zu können. Die Leistungsfähigkeit eines Zusammenspiels des VSM und MAP stellt Abbildung 3. dar.



Abbildung 2: Wirkfeld klassischer Vorgehensmodelle

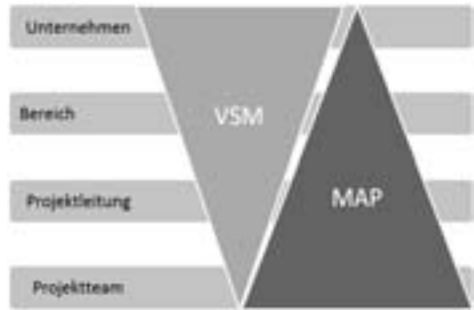


Abbildung 3: Wirkfeld VSM und MAP

5 Zusammenfassung

Unter der Annahme, dass Vorgehensmodelle nicht ausreichend das Projektmanagement der Zukunft stützen, bedarf es für insbesondere agile und komplexe Projekte einer Erweiterung der klassischen Projektmanagementansätze in der Praxis. Hier sollte der Fokus auf Projektebene auf der wirksamen Zusammenstellung des Projektteams über MAP stehen. Weiterhin muss in den nächsten höheren Rekursionsebenen des Unternehmens aus kybernetischer Sicht für die wirksame Ausgestaltung der Systemfunktionen des Unternehmens über die Anwendung des VSM gesorgt werden. Dieses Zusammenspiel

von MAP und VSM sollte in weiteren Praxisanwendungen näher validiert und untersucht werden.

Literaturverzeichnis

- [As70] Ashby, W. R.: An introduction to cybernetics. Chapman and Hall, 1970.
- [Be79] Beer, S.: The Heart of Enterprise. Chichester: Wiley, 1979.
- [Be81] Beer, S.: Brain of the Firm. 2nd ed. Chichester: Wiley, 1981.
- [Fo62] Foerster, H. von: Principles of self-organization. Transactions of the University of Illinois Symposium on Self-Organization, Robert Allerton Park, 1961. Pergamon Press, Oxford, 1962.
- [Ha10] Hanser, E.: Agile Prozesse: Von XP über Scrum bis MAP, Springer-Verlag, 2010.
- [Ha12] Hanser, E.: Erweiterung agiler Prozesse durch sozietäre Vorgehensbausteine, WI-MAV Rundbrief, 2012.
- [HW12] Hanser, E.; Wehinger, J.: Integration MAP in VSM. Private Notizen, 2012.
- [Pa61] Pask, G.: An approach to cybernetics. Hutchinson, 1961.
- [Sc00] Schwaninger, M.: Das Modell Lebensfähiger Systeme - Ein Strukturmodell für organisationale Intelligenz, Lebensfähigkeit und Entwicklung, Universität St. Gallen, (2000). Unter: <http://www.tranquilla.ch/serveattachment/984e4202df9fe30e5459fa815a35c6dc/D35.pdf> (Letzter Zugriff August 2013)
- [We12] Wehinger, J.: Management lebenszyklusorientierter Innovationssysteme, Vulkan-Verlag, 2012.

Vorgehensmodellentwurf für Forschungsprototypen

Sven Niemand, Sven Feja, Sören Witt, Andreas Speck

Christian-Albrechts-Universität zu Kiel

Institut für Informatik

24098 Kiel

{svn|svfe|swi|aspe}@informatik.uni-kiel.de

Abstract: Die Auswahl eines geeigneten Vorgehensmodells für Software-Entwicklungsprojekte von Forschungsprototypen gestaltet sich schwierig: Die klassischen Standardmodelle sind nicht an die speziellen Rahmenbedingungen eines solchen Entwicklungsprojekts angepasst. Lösungsansätze können die Modifikation bestehender Modelle und die Definition eigener Vorgehensmodelle sein. Solche an die spezifischen Ansprüche angepassten Vorgehensmodelle können bei der Bewältigung von Herausforderungen im Entwicklungsprojekt, beispielsweise bei der Umsetzung von Maßnahmen aus dem Qualitätsmanagement, unterstützen.

In diesem Beitrag wird anhand eines konkreten Beispiels das spezielle Entwicklungsumfeld von universitärer Forschungssoftware vorgestellt, aus dem Designentscheidungen zur Entwicklung eines angepassten Vorgehensmodells abgeleitet werden können.

1 Besonderheiten und Herausforderungen im Entwicklungsumfeld von Forschungssoftware

Das Entwicklungsumfeld von (universitärer) Forschungssoftware kann sich durch Besonderheiten auszeichnen, die es beispielsweise von den mitunter sehr strukturierten Vorgehensweisen in Unternehmen, aber auch von kleineren und agilen Entwicklungsprojekten mit einem festen Entwicklerteam unterscheiden.

Am Beispiel des Entwicklungsprojekts des Business Application Modelers (kurz BAM) – einem Forschungsprototypen zur Modellierung von Geschäftsprozessen und Umsetzung von Konzepten wie Sichtenbildung in Prozessen und Validierung anhand grafischer Validierungsregeln – können verschiedene Besonderheiten herausgestellt werden. Bezeichnend im Entwicklungsprojekt des BAMs ist die fortlaufende Durchführung der Entwicklungsarbeit in separaten Teilprojekten auf Basis studentischer Arbeiten: Ein erheblicher Teil der Entwicklungsarbeit wird im Rahmen von Bachelor-, Diplom- und Masterarbeiten erbracht. Häufige Entwicklerwechsel, eine geringe Zahl simultaner Projektteilnehmer und das generelle Fehlen eines klassischen Entwicklerteams sind Beispiele für besondere Merkmale dieses Projektvorgehens. Die anfangs vorgegebenen Ziele ihres Teilprojekts sollen die Studierenden weitestgehend selbstständig ausgestalten, konkretisieren und weiterentwickeln. Dabei werden sie von Lehrstuhlmitarbeitern betreut.

Für die möglichst effektive Projektdurchführung und insbesondere bei der Umsetzung von Maßnahmen aus dem Qualitätsmanagement ergeben sich in diesem Umfeld besondere Herausforderungen. Beispielsweise sind die kurzen Teilprojektlaufzeiten und regelmäßigen Entwicklerwechsel Ursache für einen relational hohen Einarbeitungsaufwand und eine hohe Diffusion von Wissen. Die Ergebnisse der Teilprojekte sollten trotz der fest eingegrenzten Bearbeitungszeit produktiv nutzbar sein, da eine Nacharbeit in der Regel in neuen Teilprojekten erfolgen müsste, wohingegen die dafür durchzuführenden Tätigkeiten eine entsprechende Durchführung nicht immer rechtfertigen. Aber auch einer relativen Unerfahrenheit der Projektteilnehmer im Bereich der Softwareentwicklung und des Qualitätsmanagements muss angemessen begegnet werden.

2 Ansprüche an Vorgehensmodelle

Durch ihre strategische Gestaltungsmöglichkeit der Projektdurchführung [Sa10] erscheinen Vorgehensmodelle prinzipiell als ein geeignetes Werkzeug, um den besonderen Herausforderungen im Entwicklungsprojekt von Forschungsprototypen zu begegnen. Wird ein definiertes Vorgehensmodell verfolgt, könnten den einzelnen Phasen gezielt Vorgaben und Maßnahmen etwa zur Erreichung bestimmter Qualitätskriterien zugeordnet werden.

Aus den Herausforderungen im Entwicklungsprojekt des BAMs können spezielle Zielsetzungen und Voraussetzungen für den Einsatz eines Vorgehensmodells abgeleitet werden. Beispielsweise sollte das Vorgehensmodell zum Erreichen einer verbesserten Software-Qualität und zur Verkürzung der Einarbeitungszeit beitragen. Es soll regelmäßige Abstimmungen zwischen Entwickler und (internem) Auftraggeber begünstigen, muss allerdings flexibel in Bezug auf die Zielvorgaben des Teilprojekts sein und die inkrementelle Entwicklung von einsatzfähigen Zwischenständen ermöglichen.

Ein besonderer Anspruch ist jedoch die Passgenauigkeit mit der spezifischen Projektstruktur: Bei der Zuordnung von Aufgaben und Verantwortlichkeiten muss ein angepasstes Vorgehensmodell beispielsweise auf zwei Rollen –Studierender und Lehrstuhlmitarbeiter – beschränkt sein. Neben einer angemessenen Steuerung der Teilprojekte sollte ferner auch das Vorgehen im Gesamtprojekt berücksichtigt werden – beispielsweise ist darzulegen, wie simultan durchgeführte Teilprojekte koordiniert und abschließend in einen gemeinsamen Prototypen integriert werden können.

3 Lösungsansätze

Die Evaluation klassischer Vorgehensmodelle hinsichtlich ihrer Eignung für die Anwendung im BAM-Entwicklungsprojekt offenbart vor dem Hintergrund der Zielvorgaben unterschiedliche und zumeist unzureichende Passgenauigkeiten. Viele der monumentalen Modelle werden beispielsweise in [Ba08] als schwergewichtig und schwierig zu durchschauen beschrieben. Für eine Anwendung im Entwicklungsprojekt erscheinen sie oftmals aufgrund von hohem Planungsaufwand oder geringer Flexibilität ungeeignet.

Die Konzeption der agilen Modelle zeichnet sich demgegenüber durch ein höheres Maß an Flexibilität aus [Ho08]. Ansätze dieser Modelle und verschiedene ihrer Grundprinzipien erscheinen für das Entwicklungsprojekt des BAM gut geeignet. Allerdings sind auch für diese Modelle umfangreiche Anpassungen erforderlich, die über das klassische Tailoring hinausgehen. Beispielsweise steht vielfach die Teamarbeit im Vordergrund, die im BAM-Entwicklungsprojekt durch die Fokussierung der Tätigkeiten beim Studierenden nur von geringer Bedeutung ist. Werden in einem Standard-Modell Rollen definiert, müssten diese im Falle der BAM-Entwicklung im Wesentlichen von einer einzigen Person (dem entwickelnden Studierenden) erfüllt werden.

Eine Alternative zur umfangreichen Adaption eines Standardmodells kann der Entwurf eines speziell an die Rahmenbedingungen angepassten Vorgehensmodells darstellen. Ein solches Modell könnte auf dem inkrementellen Basismodell [So11] basieren und auf zwei Ebenen für Gesamt- und Teilprojektdurchführung jeweils die Zuständigkeiten von Studierenden und Lehrstuhlmitarbeitern abgrenzen. In das Modell können eine Reihe geeigneter Ansätze aus agilen Vorgehensmodellen integriert werden, ebenso wie angepasste Zeitvorgaben für die Inkremente und Auslöser für Abstimmungen zwischen Studierenden und Lehrstuhlmitarbeiter. Zur Überbrückung von Wissensdefiziten kann der Detaillierungsgrad des Modells – beispielsweise zur Anwendung der bestehenden Projektinfrastruktur – an den Wissensstand der Studierenden angepasst sein.

4 Fazit

Aus verschiedenen Standard-Vorgehensmodellen – insbesondere den agilen Vorgehensweisen – steht eine Reihe geeigneter Ansätze für das BAM-Entwicklungsprojekt zur Verfügung, beispielsweise die Forderung kurzer Entwicklungszyklen und flexibler Zielvorgaben. Aufgrund der unzureichenden Passgenauigkeit eines einzelnen Modells vor dem Hintergrund der speziellen Rahmenbedingungen müssten diese Ansätze allerdings in einem speziell angepassten Vorgehensmodell zusammengeführt werden.

Literaturverzeichnis

- [Ba08] Balzert, H.: Lehrbuch der Softwaretechnik: Softwaremanagement. Spektrum Akademischer Verlag, 2008.
- [Ho08] Hoffmann, D. W.: Software-Qualität. Springer-Verlag, 2008.
- [Sa10] Schatten, A. et al.: Best Practice Software-Engineering. Spektrum Akademischer Verlag, 2010.
- [So11] Sommerville, I.: Software Engineering. Pearson Education, 2011.

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. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, 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 Magenheimer (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 Magenheimer, 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 Brömmke, 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 Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (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 and 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 Ranneberg, 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, Wolfried 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 Röbling, 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 Walther (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 health 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 Dreö 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 and 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 Brömmе, Christoph Busch, Detlef Hühnlein (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 Dreö 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 Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software
- P-152 Jan Mending, 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 Brömmе, 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. Claupein, 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 Brömmе, 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 Dreö 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 und 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 Brömmel, 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 Fährnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1
- P-176 Klaus-Peter Fährnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2
- P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fährnrich, 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 Wirtschaftsinformatik (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 Dreö 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 Landbewirtschaftung 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 Dreö 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 und 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 Brömm, 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 Dreö 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, Efthimios 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

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