

Peter Hruschka / Chris Rupp / Gernot Starke

Agility kompakt

Tipps für erfolgreiche Systementwicklung

2. Auflage

Unter agiler Mitwirkung von Jutta Eckstein, Christiane Gernert,
Thorsten Janning, Markus Reinhold und Beatrice Sablov

Spektrum
AKADEMISCHER VERLAG

Autoren:

Peter Hruschka
E-Mail: hruschka@b-agile.de

Chris Rupp
E-Mail: Chris-Rupp@sophist.de

Dr. Gernot Starke
E-Mail: gs@gernotstarke.de

Wichtiger Hinweis für den Benutzer

Der Verlag und die Autoren haben alle Sorgfalt walten lassen, um vollständige und akkurate Informationen in diesem Buch zu publizieren. Der Verlag übernimmt weder Garantie noch die juristische Verantwortung oder irgendeine Haftung für die Nutzung dieser Informationen, für deren Wirtschaftlichkeit oder fehlerfreie Funktion für einen bestimmten Zweck. Ferner kann der Verlag für Schäden, die auf einer Fehlfunktion von Programmen oder Ähnliches zurückzuführen sind, nicht haftbar gemacht werden. Auch nicht für die Verletzung von Patent- und anderen Rechten Dritter, die daraus resultieren. Eine telefonische oder schriftliche Beratung durch den Verlag über den Einsatz der Programme ist nicht möglich. Der Verlag übernimmt keine Gewähr dafür, dass die beschriebenen Verfahren, Programme usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Der Verlag hat sich bemüht, sämtliche Rechteinhaber von Abbildungen zu ermitteln. Sollte dem Verlag gegenüber dennoch der Nachweis der Rechtsinhaberschaft geführt werden, wird das branchenübliche Honorar gezahlt.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

2. Auflage 2009

© Spektrum Akademischer Verlag Heidelberg 2009
Spektrum Akademischer Verlag ist ein Imprint von Springer

09 10 11 12 13 5 4 3 2 1

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Planung und Lektorat: Dr. Andreas Rüdinger, Barbara Lühker
Umschlaggestaltung: SpieszDesign, Neu-Ulm
Herstellung und Satz: Crest Premedia Solutions (P) Ltd., Pune, Maharashtra, India

ISBN 978-3-8274-2092-3

Inhalt

(Statt einer) Einführung	VI
Warum Agilität?	1
Das agile Manifest – Ein neues Wertesystem.....	2
Angemessenheit	3
Ergebnisorientierung	4
Offen für Änderungen – Iterativ inkrementelle Entwicklung	8
Software von Menschen für Menschen.....	9
Die 1 + 6 Maximen der Agilität.....	10
Die Anwendung agiler Prozesse	12
Agile Systemanalyse.....	12
Agile Architektur	18
Agile Programmierung	29
Agile Dokumentation	33
Agile Rückblicke und Retrospektiven	35
Agiles Projektmanagement	37
Agile Organisation.....	44
Agile Methoden im Überblick.....	52
Die Crystal Methodenfamilie.....	52
Adaptive Software Development (ASD)	58
Scrum	65
ARTE – eine agile Vorgehensweise für technische Systeme	71
Der Rational Unified Process (RUP®)	77
eXtreme Programming (XP)	84
Agile Softwareentwicklung in großen Projekten	92
Agilität und Disziplin.....	97
Wege in die Agilität	101
Die Autoren	108
Literatur.....	110
Index.....	114

(Statt einer) Einführung

*Nicht die schnellsten
oder stärksten überleben,
sondern diejenigen,
die sich am schnellsten auf veränderte
Lebensbedingungen einstellen können.*
Charles Darwin

(Statt einer) Einführung

Flink und behände...

Kennen Sie die Geschichte der Ameisenvölker? Die ersten Ameisenvölker lebten vor Millionen Jahren im feuchtwarmen Klima längst vergangener Zeiten. Sie krabbelten unter den Füßen urzeitlicher Tiere, an den giftigen Stängeln riesiger Farne und an den Ufern reißender Ströme entlang, ständig auf der Suche nach Futter. Damals schmeckten den Ameisen die Überbleibsel von Mammutknochen besonders gut.

Dann wurde es kalt. Eisig kalt! Viele der großen Tiere gingen an der zermürbenden Kälte der Eiszeit zugrunde. Die Ameisen nicht – flink und behände hatten sie sich andere Liebesspeisen gesucht, und auch neue Wege. Jahrtausende später taute das Eis – und die Ameisen lernten erneut, sich an der Hitze zu erfreuen. Wieder und wieder wandelte sich die Umwelt – und die Ameisen wandelten sich mit. Zwar änderten sie ihre Form und ihr Aussehen kaum, ihr Verhalten und ihre Vorlieben passten Sie jedoch kontinuierlich den Gegebenheiten ihrer Umwelt an.

Heute gibt es weder starke Mammut noch riesige Giftfarne. Aber Ameisen in rauen Mengen. Und wenn wir sie fragen könnten, würden sie bestimmt sagen, dass sie sich wohl fühlten und Spaß hätten bei ihrer ewigen Krabbelei.

... oder erstarbt in Tradition

Vor langen Jahren begann der Hunger der Menschheit nach Kohle von Jahr zu Jahr zu wachsen. Das führte in vielen Gegenden Europas zur Gründung von Bergwerken – der Abbau in der Tiefe war eine gefährliche Unternehmung, die mit abenteuerlichen Methoden Mutter Erde das dunkle und wertvolle Gut entlockte. Viele Arbeiter ließen ihr Leben dabei, unter Strapazen den Grundstoff der industriellen Revolution aus dem Bauch der Erde zu stemmen.

Die Bergwerke wuchsen zu gigantischen Unternehmen heran, ihre Besitzer häuften fantastische Reichtümer an. Jahr um Jahr fuhren die Bergleute in die dunklen Stollen, Jahr um Jahr blieb alles beim Alten. Die Herren und Arbeiter der Bergwerke glaubten, so würde es endlos weitergehen.

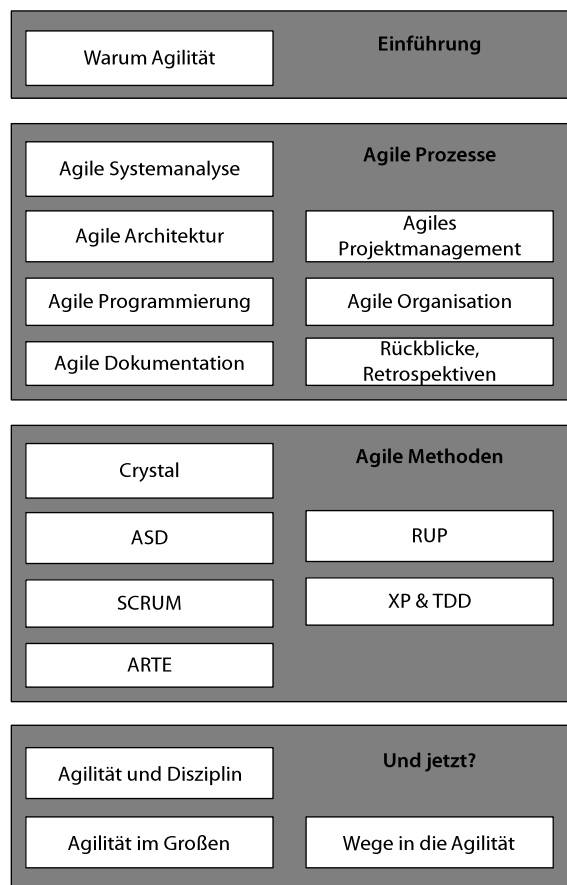
Doch vor einigen Jahren blieb die Nachfrage nach Kohle hinter der geförderten Menge zurück, die Preise sanken weltweit. Die Bergwerke warfen nicht mehr genug Rendite ab. Die Realität holte die langen Träume vom ewigen Reichtum und gesicherten Arbeitsplätzen abrupt ein.

Heute wird immer noch Kohle gefördert. Und einige der Beteiligten glauben immer noch, Bergwerke seien zukunftssträchtige Unternehmen.

(Statt einer) Einführung

Flexibilität

Wir können nicht besonders weit in die Zukunft schauen. Trotzdem versuchen wir immer wieder, IT-Projekte über Monate, teilweise Jahre im Voraus minutiös zu planen und diese Pläne einzuhalten, statt uns kontinuierlich der Realität anzupassen.



Warum Agilität?

*Der Wechsel allein
ist das Beständige*
Arthur Schopenhauer.

Bei der NATO-Konferenz 1968 in Garmisch-Partenkirchen wurde der Begriff „Software Engineering“ geprägt, um die anwesenden Wissenschaftler und Praktiker dazu anzuregen, als Reaktion auf die Softwarekrise eine ingenieurmäßige Vorgehensweise bei der Entwicklung von Software und Systemen vorzugeben. Viele Methoden und Verfahren wurden seither erfunden; Die Vorschläge, wie Softwareentwickler vorgehen sollen, wurden immer aufwendiger, die Handlungsanweisungen immer umfangreicher und dicker. Aber seien Sie einmal ehrlich: Sind Sie glücklich mit dem erreichten Stand? Bringen die Vorgehensmodelle und Methoden die erhoffte Wirkung? Erhalten Auftraggeber dadurch wirklich das, was sie haben wollen, rechtzeitig und zu akzeptablen Kosten? Und haben Entwickler Spaß daran, so ihre Software zu entwickeln? Werden diese Vorgaben auf breiter Front akzeptiert als die beste Art, Software und Systeme zu entwickeln?

Zu Beginn des neuen Jahrtausends haben Ward Cunningham und Kent Beck [Beck00] mit provozierenden Aussagen zu eXtreme Programming (XP) ein anderes Leitbild für Softwareentwicklung vorgegeben. Dieses neue Leitbild ist scheinbar in vielen Punkten die Abkehr von der Forschung der letzten 30 Jahre. Eine Gruppe von 17 praktizierenden Softwareentwicklern und Methodenforschern hat sich im Februar 2001 in Utah getroffen und das neue Idealbild für Softwareentwicklung in Worte gefasst: das agile Manifest war geboren. Darauf basieren heute eine ganze Reihe neuer Entwicklungsansätze, von denen eXtreme Programming nur einer, wenn auch der bisher bekannteste, aber leider auch der radikalste ist.

Lassen Sie sich durch dieses Buch zum Nachdenken anregen, ob Ihre Art Software zu erstellen diesem neuen Leitbild schon nahe kommt oder ob Sie noch nach Perfektionierung eines Vorgehensmodells streben. Wir haben für Sie in diesem Buch neben den Grundideen agiler Softwareentwicklung auch zusammengefasst, wie sich Agilität auf die unterschiedlichen Rollen in der Softwareentwicklung auswirkt, was es

Warum Agilität?

für Projektleiter, Analytiker, Designer und Programmierer bedeutet, ja vielleicht sogar, wie eine ganze Organisation diesen Kulturwandel vollziehen kann.

Außerdem erläutern wir Ihnen die Kerngedanken einiger populärer agiler Methoden – die Aspekte, die sie von konventionellen Methoden unterscheiden.

Das agile Manifest – Ein neues Wertesystem

Worauf konnten sich die 17 Experten in Utah nach heftigen Diskussionen einigen? Lesen Sie im Folgenden eine Übersetzung der auf den Punkt gebrachten Aussagen des agilen Manifests (Die englische Fassung finden Sie unter www.agilemanifesto.org).

Wir entdecken bessere Wege zur Entwicklung von Software, indem wir Software entwickeln und anderen bei der Entwicklung helfen. Durch diese Tätigkeiten haben wir gelernt, dass uns

- *Menschen und Zusammenarbeit* *mehr als Prozesse und Werkzeuge bedeuten,*
- *lauffähige Software* *mehr als umfangreiche Dokumentation,*
- *Zusammenarbeiten mit Auftraggebern* *mehr als Vertragsverhandlungen und*
- *Reagieren auf Änderungen* *mehr als das sture Befolgen eines Plans.*

Natürlich sind auch die Dinge rechts wichtig, aber im Zweifelsfall schätzen wir die linken höher ein.

Überlegt man sich, was diese vier Wertaussagen für die vielen Tätigkeiten bei der Softwareentwicklung bedeuten und was sie für Konsequenzen auf das Verhalten der am Projekt beteiligten Personen haben, dann ergibt sich daraus ein ganz anderes Weltbild, als es viele Auftraggeber, Projektleiter und die Mitglieder im Projektteam heute gewohnt sind.

Mit dem Einleitungssatz positioniert sich die Gruppe von Experten: „Wir sind keine Methodenentwickler aus dem Elfenbeinturm. Wir stehen alle mitten in Softwareprojekten. Wir praktizieren diese Tätigkeiten aktiv, geben unser Wissen aber gerne auch an Kollegen und Kunden weiter. Und wir denken auch darüber nach, was wir hier

Angemessenheit

tun und ob es hilft oder eher schadet. Die Ergebnisse dieses Nachdenkens und Abwägens konnten wir in vier Grundsätzen zusammenfassen.“ Der Einleitungssatz fasst damit eine Grunderkenntnis agilen Handelns zusammen: Hören Sie auf diejenigen, die sich aus eigener Erfahrung mit Methoden auskennen. Vertrauen Sie erprobten, erfolgreichen Praktiken mehr als abstrakten Ideen von Theoretikern, die selbst schon lange keine Softwaresysteme mehr entwickelt haben. Der Nachsatz des Manifests schlägt die Brücke zu den Fortschritten in den letzten 30 Jahren. Die waren nicht alle falsch: der Versuch, den Entwicklungsprozess immer besser zu verstehen, zu definieren und Werkzeuge dafür zur Verfügung zu stellen; der Versuch, die „Zwischenergebnisse“ auf dem Weg zu auslieferbaren Systemen sauber zu dokumentieren; das Absichern von Projektzielen durch Verträge und das Vorausdenken, um in Form von Plänen dem Team Handlungsanweisungen mitzugeben. Aber vielleicht haben wir in all diesen Punkten übertrieben und damit die wirklich wichtigen Aspekte der Softwareentwicklung vernachlässigt.

eXtreme Programming war eine der ersten Methoden, die die abstrakten Ideen des agilen Manifests durch konkrete Vorschläge präzisiert hat. XP ist aber – wie der Name schon andeutet – sicherlich auch die extremste Interpretation dieses neuen Wertesystems. Etwas überspitzt formuliert wird darin u. a. Folgendes gefordert: Das einzige Ergebnis, das zählt, ist lauffähige Software. Wir entwickeln in so kurzen Zyklen (wenige Stunden bis maximal einige Tage), dass wir außer lauffähiger Software keine Dokumente brauchen. Auch keine Spezifikationen über die Problemstellung, denn der Kunde ist immer griffbereit im gleichen Raum, sodass Unklarheiten sofort beseitigt werden können (Mehr zu XP finden Sie im Abschnitt „eXtreme Programming (XP)“, Seite 84).

Angemessenheit

XP ist vielleicht unter gewissen Randbedingungen ein ideales Vorgehen – wenn es sich um relativ kleine Projekte handelt, mit einem einzelnen, hoch verfügbaren Kunden, der weiß, was er will, und einem Entwicklungsteam, das im selben Raum mit dem Kunden direkt Lösungen finden kann. Diese Randbedingungen sind aber nicht in jedem Projekt gegeben. Deshalb ist XP definitiv nicht immer und für jeden die richtige Antwort.

Warum Agilität?

Andere agile Methoden setzen die Grundsätze des agilen Manifests in weniger radikale Handlungsanweisungen um. Statt „nur der Source Code reicht“ hören Sie z.B.: Natürlich ist unser Hauptziel einsetzbare, lauffähige Software, aber wir müssen uns auch richtig aufstellen für den nächsten Schritt. Und richtig aufgestellt sein bedeutet, dass den Entwicklern von Folgeversionen vielleicht neben dem Source Code auch ein ausgewogenes Maß an Zusatzdokumenten zur Verfügung stehen sollte (z.B.: Architekturdokumente, Regressionstestdaten, ...), sodass sie nicht alles wieder durch Reengineering herausfinden müssen.

Aber wie viel Zusatzaufwand und Zusatzdokumentation soll ein Projekt erzeugen? Wie viel ist ausreichend und wo beginnt die bürokratische Übertreibung? Die Antwort muss jeder für jedes Projekt selbst geben. Der Schlüssel dazu liegt in einer Risikoabwägung: Was gewinnen wir, wenn wir die Schritte durchführen und die Ergebnisse erzeugen, bzw., was verlieren wir, wenn wir es nicht tun? Jeder Projektbeteiligte beurteilt daher für seinen Zuständigkeitsbereich ständig die Risiken und entscheidet dann über die geeigneten Maßnahmen. Agil heißt beweglich und flexibel sein, mitdenken statt „Dienst nach Vorschrift“ und Dogma.

Somit wird für agile Entwicklung die folgende Maxime zum Ausgangspunkt für alle Entscheidungen über Aktivitäten, Ergebnisse und Rollen in der Softwareentwicklung:

„eher Angemessenheit als Extremismus!“.

In den folgenden Abschnitten diskutieren wir, was diese risikogetriebene Angemessenheit für drei wichtige Faktoren in der Systementwicklung konkret bedeutet:

- Für das, was wir erreichen wollen: → Ergebnisse
- Für das, was wir (nicht) haben: → Zeit
- Für die, die an der Entwicklung beteiligt oder davon betroffen sind: → Menschen

Ergebnisorientierung

Überlegen Sie einmal Folgendes: Können wir ein erfolgreiches Softwareentwicklungsprojekt haben, wenn wir das Vorgehensmodell der Firma perfekt eingehalten haben, aber keine brauchbare Software geliefert haben? Wenn wir alle Dokumentationen abgegeben haben, aber keine laufende Software?

Ziel eines Softwareentwicklungsprojekts ist es, laufende Software zum Nutzen des Unternehmens in Betrieb nehmen zu können. Deshalb geht an dem Ergebnis Source Code in IT-Projekten kein Weg vorbei. Nutzen bringende Software sollte unser Hauptmaß für den Projektfortschritt sein. Ab einer gewissen Größe des Source Codes, wenn die Software langfristig für eine ganze Produktfamilie eingesetzt wird oder wenn an der Beauftragung und an der Entwicklung viele Personen an vielen Standorten beteiligt sind und sich abstimmen müssen, dann werden wir auch andere Projektergebnisse benötigen, z. B. Zielvereinbarungen, Architekturdokumente, Geschäftsprozessmodelle, Qualitätsrichtlinien und Spielregeln für die Projektorganisation und die Kommunikation im Projekt.

Die konkreten Gegebenheiten in Projekten, die Ziele und die Randbedingungen bestimmen, wie viel wir davon jeweils brauchen und in welcher Form. Je kleiner und lokaler das Projekt ist, desto weniger Vorschriften sind notwendig und desto weniger Formalismus muss dafür aufgewandt werden.

Sie sehen, dass „Agilität“ keineswegs die Erlaubnis zur Anarchie einschließt. Agile Vorgehensmodelle haben Ergebnisse, nur unterscheiden sich diese in unterschiedlichen Branchen und Projekten in Anzahl, Tiefgang und Formalität.

Der Unterschied zu „schweren“ Vorgehensmodellen besteht also hauptsächlich darin, dass es keine allgemeingültigen Vorschriften oder „Zwischenergebnisse“ auf dem Weg zum Projektziel (= lauffähiges, Nutzen bringendes Softwaresystem) gibt, sondern dass konkret für jedes Projekt Kosten und Nutzen von Zwischenergebnissen abgewogen werden. Was auch immer das Ergebnis ist, die Devise dafür muss sein: so wenig wie möglich, so viel wie nötig. Welchem Risiko setzen wir uns aus, wenn wir das Ergebnis nicht (oder nicht formal genug oder nicht umfangreich genug) erzeugen? Sie sollen in einem agilen Projekt keine Ergebnisse erzeugen, die mehr kosten als sie Nutzen bringen! Das Ergebnis muss jemandem etwas Wert sein. Sie sollten diejenigen kennen, denen das Ergebnis so viel Wert ist, wie Sie für die Erstellung oder Pflege aufwenden müssen. Suchen Sie diese „Sponsoren“ für jedes Ergebnis. Wenn Sie keinen Sponsor dafür finden, sollten Sie auch den Mut haben, das Ergebnis nicht zu produzieren. Auf keinen Fall sollten Sie sich hinter einem anonymen Vorgehensmodell verstecken, das die Erstellung bestimmter Ergebnisse verlangt. Als Maxime formuliert: Agile Projekte arbeiten

Warum Agilität?

„eher ergebnisorientiert als prozessorientiert“.

Wir konzentrieren uns auf das gewünschte Ergebnis mit seinen geforderten Qualitätsaspekten. Das Erreichen der Ergebnisse ist uns wichtiger als der Weg dorthin. Sie wissen: Viele Wege führen nach Rom. Um ein bestimmtes Ergebnis zu erreichen, ist uns (fast) jeder Weg recht. Sie sollten daher – im Gegensatz zu manchen Vorgehensmodellen – die Wege zur Zielerreichung nicht zu starr vorschreiben, sondern durchaus Alternativen zulassen und deren Auswahl in die Verantwortung der Projektbeteiligten verlagern.

Haben Sie nicht auch Ihre persönlichen Vorlieben dafür, wie Sie eine bestimmte Aufgabenstellung anpacken? Wir gehen davon aus, dass Sie Ihre Erfolgsrezepte dafür, wie man etwas anpackt, auch guten Freunden weitergeben. Andererseits hören Sie sicherlich auch zu, wenn gute Bekannte Ihnen ihre Tipps und Tricks verraten, wie sie schneller und effizienter zum Ziel kommen. Und Sie probieren das vielleicht auch bei nächster Gelegenheit aus.

Die Grundbotschaft von agilen Prozessen ist also: Arbeiten Sie mit einer Sammlung von „Best Practices“, die Sie aus eigener Erfahrung für wertvoll halten oder die Sie von Personen übernehmen, die Ihr Vertrauen genießen. Picken Sie aus diesen Best Practices immer das heraus, was Sie Ihrer Meinung nach am schnellsten und besten zum Ergebnis führt.

Es kann sein, dass es aus Projektgründen notwendig ist, nicht jedem Einzelnen den Weg zum Ziel komplett freizustellen. Vereinheitlichen oder standardisieren Sie Vorgehensweisen aber nur so weit, wie es notwendig ist. Man muss die Vorgehensweise zur Erreichung bestimmter Ziele nicht notwendigerweise für eine ganze Firma angleichen, manchmal reicht es auch im Team oder in der Abteilung. Lassen Sie Varianten in den Methoden zu, wenn sie zum gleichen Ergebnis führen. Die besten Anforderungen, Architekturen und Entwürfe entstehen in selbst organisierenden Teams, die ihren Arbeitstil gefunden haben und die Methoden und Verfahren freiwillig anwenden, weil sie von deren Wirksamkeit überzeugt sind. Geben Sie diesen Teams die Chance, in regelmäßigen Abständen zu reflektieren, wie sie noch effektiver werden können. Dadurch kommen immer neue Best Practices in den Vorrat dazu, bzw. existierende Praktiken werden optimiert oder ausgetauscht.

Ergebnisorientierung

Auf den Punkt gebracht bedeutet das: Wir ersetzen große, starre Vorgehensmodelle mit wenigen Freiheitsgraden durch kleine, essenzielle Vorgehensmodelle kombiniert mit vielen erprobten Praktiken. Die Maxime ist:

„eher best practices aus Erfahrung als verordnete Vorgaben“.

Die effizienteste und effektivste Art der Informationsvermittlung innerhalb eines Teams ist von Angesicht zu Angesicht. Im Gegensatz zur Weitergabe von Informationen in Dokumentenform haben Sie bei direkter Kommunikation den Vorteil, nicht nur den Inhalt übermittelt zu bekommen, sondern auch den Tonfall zu hören, die Körpersprache der anderen zu sehen und die Stimmung aufzuschnappen. Dies alles fehlt Ihnen bei ausschließlich schriftlicher Kommunikation über Aspekte des Projekts. Nutzen Sie diese Stärken der persönlichen Kommunikation! Im Zweifelsfall sollten Sie lieber miteinander diskutieren, statt zu versuchen, etwas ganz präzise und formal zu beschreiben. Selbstverständlich müssen in jedem Projekt auch einige Ergebnisse schriftlich festgehalten werden. Nur so lässt sich oft das Erreichte und Beschlossene für viele Personen zugänglich machen. Suchen Sie aber nach Gelegenheiten, Schriftgut zu vermeiden und Kontakt zu anderen zu suchen. Ein kurzes, offenes Gespräch über die Ergebnisse und Probleme der Woche ist oft mehr Wert als die routinemäßig kopierten Wochenberichte; ein persönliches Treffen von verteilt arbeitenden Teams oder eine Videokonferenz ist im Endeffekt billiger (und förderlicher für die Zusammenarbeit) als der elektronische Austausch von Dokumenten. Unsere Maxime zu diesem Thema lautet:

„eher miteinander reden als gegeneinander schreiben“.

Wiederum bestimmt das Risiko im Projekt, wie viel wir an Dokumentation schriftlich festhalten müssen. Wir sind jedoch sicher, dass Sie bei einer kritischen Betrachtung Ihres derzeitigen Vorgehens und einer kritischen Würdigung aller Dokumente, die Sie heute erzeugen, zahlreiche Kandidaten finden, die vielleicht vermieden werden können und sollten, wenn Sie stattdessen mehr „Kultur des miteinander persönlich Kommunizierens“ einführen.

Warum Agilität?

Offen für Änderungen – Iterativ inkrementelle Entwicklung

*Lang ist der Weg durch Vorschriften,
kurz und wirkungsvoll durch Beispiele.*
Seneca

Vorhersagen ist schwierig, besonders für die Zukunft! Versuchen Sie daher erst gar nicht, ein sehr weit in der Zukunft liegendes Ziel mit sehr großer Präzision vorhersagen zu wollen. In den letzten Jahren sind die Zeiträume, die einem Entwicklungsteam bis zur Fertigstellung Nutzen bringender Software gegeben werden, deutlich kürzer geworden. Mitte der 1980er-Jahre galt noch die Spielregel: Kein Softwareprojekt darf länger als 12–18 Monate dauern, bevor es lauffähige Ergebnisse produziert. Heute liegen die Zeiten in vielen Branchen oft (weit) darunter. Sie sollen durchaus Visionen haben, die längere Zeiträume umfassen. Aber diesen Visionen sollten Sie schrittweise konkret näher kommen. In kurzen Iterationen werden Inkremente des endgültigen Systems entwickelt, die für sich gesehen schon geschäftlich Wert bringend genutzt werden können. Sehen Sie es einmal von dieser Seite: Solange nichts von Wert für das Geschäft freigegeben wird, ist das bis dahin in der Entwicklung verbrauchte Geld totes Kapital. Ihr Ziel sollte es sein, die „Lagerkosten“ für nicht freigegebene Software zu reduzieren und dieses tote Kapital zu minimieren.

Betrachten Sie, wie lange Sie „Geld verbrauchen“ bevor Geld zurückkommt. Suchen Sie die Gründe, die dazu führen, dass der Zeitraum so lange ist und räumen Sie sie aus dem Weg. Setzen Sie sich als Ziel, den heute gebrauchten Zeitraum zu halbieren oder zu dritteln. Die Erfahrung zeigt, dass das meistens gar nicht so schwer erreichbar ist, wenn man es nur will und bewusst daraufhin arbeitet. Für viele Softwareprojekte gilt heute: Zyklen von 2–4 Monaten sind durchaus machbar, manchmal sogar wesentlich kürzere (wie bei XP), manchmal auch etwas längere (6 Monate). Zykluszeiten von mehr als 6 Monaten sollten sehr kritisch hinterfragt werden.

Die Releasezyklen nach außen (an Kunden oder Anwender) müssen nicht unbedingt den Iterationen angepasst werden. Kürzere Zykluszeiten beruhigen vielleicht auch die Nerven von Managern und Auftraggebern, die jetzt das Endergebnis konkret wachsen sehen – auch

Software von Menschen für Menschen

wenn es nicht in Betrieb geht. Auch die Entwickler arbeiten nicht mehr so lange „nur mit Papier“. Ihre Erfolgserlebnisse bekommen Sie jetzt mehrfach täglich, wöchentlich oder zumindest in wenigen Wochen, und nicht erst in 1–3 Jahren. Projektmitarbeiter werden dem Projektleiter für die nächsten Tage, Wochen oder wenige Monate sehr präzise sagen können, was machbar ist und was nicht. Schätzungen für eine noch weit entfernte Zukunft sind für alle Beteiligten viel schwieriger.

Kürzere Zykluszeiten haben noch einen anderen wesentlichen Vorteil. Wir wissen inzwischen, dass sich zwar die Ziele und Visionen langfristig festlegen lassen, dass sich aber die genaueren Anforderungen an Software um ca. 1–2 % pro Monat ändern. Eine grobe Planung für ein Gesamtprojekt, gepaart mit Detailplanungen in viel kürzeren Abständen, gibt Ihnen die Möglichkeit, neue oder geänderte Anforderungen bei jeder neuen Iteration zu begrüßen. Agile Prozesse nutzen Änderungen zum Wettbewerbsvorteil für den Kunden. Der Kunde bekommt in jeder Iteration das, was er kurzfristig am Dringendsten benötigt. In kurzen Abständen kann man die Ziele für die nächste Iteration wieder neu diskutieren und neue Prioritäten setzen. Daraus ergibt sich eine der wichtigsten Maximen der Agilität:

„eher offen für Änderungen als starres Festhalten an Plänen“.

Software von Menschen für Menschen

„Softwareentwicklung ist ein kooperatives Spiel“, sagt Alistair Cockburn, einer der Gründer der Agile Alliance. Ein Spiel, in dem nach Win-Win-Situationen gestrebt werden soll. Betrachten wir ein paar von den oben diskutierten Ideen nochmals aus Sicht der beteiligten Personen. Statt den Mitarbeitern zu viele Vorschriften über die Vorgehensweise zu geben, sollten Sie eher motivieren, bestimmte Dinge zu tun:

- Mehr Ausbildung über das „Warum“, weniger Vorschriften über das „Wie“.
- Alternative Wege zulassen, solange das Ergebnis erreicht wird – wenn nicht externe Zwänge dagegen stehen.
- Mehr Ideen aus der Praxis standardisieren, statt aus dem „Elfenbeinturm“ vorzuschreiben.
- Regelmäßige Feedbackrunden über die Prozessqualität.

Als Manager sollten Sie den Mut haben, Kontrolle abzugeben, rigide Kontrollmechanismen wegzulassen, und mehr Vertrauen zu einem

Warum Agilität?

gut motivierten Team haben. Im Zusammenspiel mit den raschen Feedbackzyklen ist die Kontrolle ohnehin oftmals gegeben (oder sogar besser als früher, denn laufende Software ist aussagekräftiger als Tonnen von Papierspezifikationen!)

Kunden sollten Sie weniger als Auftraggeber sehen, sondern als Projektpartner, die am gleichen Strang ziehen, die Visionen teilen und die Details am Anfang genauso wenig kennen wie Sie. Sie kooperieren ständig und vertrauensvoll miteinander, auch über den Abschluss eines Vertrages hinaus. Sie tauschen regelmäßig, in kurzen Zyklen die neuesten Erkenntnisse aus, die dazu beitragen, dass im nächsten Schritt immer die wichtigsten Teile der Software geliefert werden können. Diskutieren Sie langfristige Visionen und was es dem Auftraggeber Wert ist und den Auftragnehmer kostet. Verhandeln Sie Details aber nicht am Anfang, sondern präzisieren Sie diese, sobald es notwendig ist (Variable Scope Contracts). Auch für den Auftraggeber gilt das Gleiche wie im Team: Geben Sie Kontrolle ab und suchen sich stattdessen lieber einen vertrauenswürdigen Lieferanten, der auch an einer langfristigen Partnerschaft interessiert ist und Sie deshalb nicht über den Tisch ziehen wird. Diese hier nur kurz angerissenen Ideen gehören zu den Kernbotschaften der Agilität und schlagen sich in den folgenden beiden Maximen nieder:

*„eher Menschen und Kommunikation als Prozesse und Tools“,
„eher Vertrauen als Kontrolle“.*

Projekte werden rund um motivierte Individuen geschaffen. Stellen Sie ein Arbeitsumfeld zur Verfügung und die Unterstützung, die die Beteiligten brauchen, und vertrauen Sie darauf, dass alle ihren Job machen werden. Achten Sie darauf, dass Sponsoren, Entwickler und Anwender das Entwicklungstempo auf Dauer mitgehen können.

Die 1 + 6 Maximen der Agilität

So wenig wie möglich, so viel wie nötig! Streben Sie nach Einfachheit. Maximieren Sie die Menge der Arbeit, die Sie *nicht* machen müssen und der Dokumente, die Sie *nicht* schreiben, lesen und pflegen müssen. Aber in allen Fällen gilt die oberste Maxime: angemessen statt extrem! Für sehr komplexe Projekte mit vielen Randbedingungen und Vorschriften von außen unterscheidet sich agile Entwicklung vielleicht

Die 1 + 6 Maximen der Agilität

nur wenig von den als „schwergewichtig“ gescholtenen Vorgehensmodellen. Für die Majorität der IT-Projekte sollte agile Vorgehensweise jedoch deutlich weniger Bürokratie und Papierflut liefern – allerdings bei etwas erhöhtem Risiko. Beachten Sie bei den weiteren sechs Maximen, die wir in den letzten Abschnitten diskutiert haben, dass die jeweils rechte Seite durchaus ihre Berechtigung hat, dass Sie aber eher nach der linken Seite streben sollten.