

Jutta Eckstein

Agile Softwareentwicklung mit verteilten Teams

dpunkt.verlag

Agile Softwareentwicklung mit verteilten Teams



Jutta Eckstein ist Beraterin und Trainerin für agile Softwareentwicklung, Kommunikations-Managerin sowie Partner von IT Communication. Ihr Wissen über agile Prozesse basiert auf mehr als zehn Jahre langer Erfahrung bei der Entwicklung objektorientierter Anwendungen. In Projekten mit bis zu 300 Personen lag ihr Schwerpunkt auf der Einführung und Umsetzung angemessener (agiler) Prozesse. Sie ist Mitglied der Agile Alliance und unterstützt das Agile Manifest. Sie hält Seminare und ist Vortragende auf mehreren Konferenzen (z.B. OOP, Jax). Mehr Informationen finden Sie unter www.jeckstein.de oder www.it-communication.de.

Jutta Eckstein

Agile Softwareentwicklung mit verteilten Teams



dpunkt.verlag

Jutta Eckstein
jutta@eckstein.com

Website zum Buch:
www.verteilte-teams.de

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Abbildungen: Katja Gloggenjiesser, www.grellgelb.de
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, Paderborn

Die englische Originalausgabe erscheint bei Dorset House Publishing Co., Inc., New York.

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

ISBN:
Buch 978-3-89864-630-7
PDF 978-3-86491-082-1
ePub 978-3-86491-083-8

1. Auflage 2009
Copyright © 2009 dpunkt.verlag GmbH
Ringstraße 19 B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsverzeichnis

1	Einführung	1
1.1	Fokus dieses Buches	2
1.1.1	Projekterfahrung	2
1.1.2	Wem hilft dieses Buch	3
1.1.3	Was dieses Buch nicht beinhaltet	3
1.2	Fahrplan für das Buch	4
2	Agilität und verteilte Projekte	7
2.1	Verteilte Entwicklung verstehen	7
2.1.1	Beteiligung mehrerer Entwicklungsstandorte	8
2.1.2	Verteilte und verstreute Teams	8
2.1.3	Große Projekte	9
2.1.4	Beteiligung verschiedener Firmen	10
2.1.5	Verschiedene Strukturen	11
2.1.6	Weit entfernte Kunden	12
2.1.7	Zentrale Koordinierung oder globale Integration	12
2.1.8	Distanz überwinden	13
2.2	Agilität verstehen	14
2.2.1	Wertesystem	14
2.2.2	Systemische Vorgehensweise	15
2.2.3	Risikoreduzierung	16
2.2.4	Agil: Nicht notwendigerweise produktiver	17
2.2.5	Mehr als nur Praktiken	17
2.2.6	Weder chaotisch noch undiszipliniert	17
2.3	Einfluss agiler Prinzipien auf verteilte Entwicklung	18
2.4	Zusammenfassung	21

3	Teambildung	23
3.1	Featureteams	25
3.1.1	Zusammensitzende oder standortübergreifende Teams	26
3.1.2	Verstreute Teams	28
3.1.3	Ein Team formen	29
3.2	Rollen	32
3.2.1	Konstellation eines Featureteams	32
3.2.2	Architekt und Chefarchitekt	34
3.2.3	Coach	38
3.2.4	Product Owner oder Produktmanager	40
3.2.5	Projektleiter	43
3.2.6	Schlüsselrollen sitzen mit ihren Teams zusammen	44
3.3	Konzeptionelle Integrität sicherstellen	45
3.3.1	Startteam hat Vorbildfunktion	45
3.3.2	Technisches Serviceteam	45
3.4	Zusammenfassung	46
4	Kommunikation und Vertrauen	49
4.1	Vertrauen und gegenseitiger Respekt	50
4.1.1	Grenzwert für vertrauensvolle Beziehungen	51
4.1.2	Wechselnde Treffpunkte	52
4.1.3	Vokabular	52
4.2	Kommunikation	54
4.2.1	Persönliche Teammeetings	54
4.2.2	Persönliche Projektmeetings	55
4.2.3	Mitarbeiterrotation	56
4.2.4	Kommunikation hat ihren Preis	56
4.2.5	Kenntnis des Kommunikationsflusses	57
4.3	Kulturelle Unterschiede	58
4.3.1	Fokus auf Gemeinsamkeiten	59
4.3.2	Gestaltung einer Kultur	60
4.3.3	Realistische Planung	62
4.3.4	Verantwortung übernehmen	62
4.3.5	Probleme ansprechen	63
4.3.6	Ehrliches Feedback geben	64
4.3.7	Lärm	65
4.3.8	Humor	66
4.3.9	Sorgfältige Auswahl der Kommunikationsmedien	66
4.4	Zusammenfassung	68

5	Standorte in Verbindung halten	69
5.1	Kommunikationsvermittler	70
5.1.1	Kommunikationsvermittler als Vertrauensperson	71
5.1.2	Kompetenzen eines Kommunikationsvermittlers	71
5.1.3	Management durch Herumfliegen	71
5.2	Botschafter	72
5.2.1	Standorte repräsentieren	73
5.2.2	Eigenschaften und Kompetenzen eines Botschafters	74
5.2.3	Terminplan	75
5.2.4	Konkrete Aufgaben	75
5.3	Soziale Verbindungen	76
5.3.1	Gemeinsam feiern	76
5.3.2	Die Kraft der Bilder	76
5.3.3	Am Leben der anderen teilhaben	77
5.3.4	Reisen angenehmer machen	77
5.4	Werkzeuge	78
5.4.1	Direkte Verbindungen	78
5.4.2	Synchron versus asynchron	79
5.4.3	Audio oder Video	80
5.4.4	Instant Messaging	81
5.4.5	E-Mail	82
5.4.6	Virtueller Raum	83
5.4.7	Gemeinsames Repository	83
5.4.8	Wiki und andere Kollaborationsplattformen	83
5.5	Zusammenfassung	85
6	Entwickeln und Ausliefern	87
6.1	Iterationen	88
6.1.1	Iterationslänge	88
6.1.2	Das Konzept von done-done	89
6.1.3	Iterationen als Herzschlag	90
6.1.4	Verzögern der Auslieferung	90
6.2	Release	92
6.2.1	Release-Iteration	92
6.2.2	Releasestandort	93
6.3	Integration und Build	93
6.3.1	Lokal funktionierende Integration	94
6.3.2	Integration erfordert Aufwand	94
6.3.3	Ein fehlgeschlagener Build stoppt die Produktion	96
6.3.4	Integration als Herz des Projekts	97

6.4	Infrastruktur	98
6.4.1	Prozesse und Werkzeuge für Integration und Build	98
6.4.2	Konfigurationsmanagement	100
6.4.3	Strom	103
6.4.4	Netzwerk	103
6.4.5	Sicherheit	104
6.4.6	Werkzeuge	105
6.5	Zusammenfassung	105
7	Der Geschäftswert ist das Ziel	107
7.1	Steuerung mithilfe wertvoller Funktionalitäten	107
7.1.1	Verbindung zum realen Kunden	108
7.1.2	Iterationsvorbereitung	110
7.1.3	Anforderungen verstehen	112
7.1.4	Benötigte Dokumentation als Anforderung behandeln	113
7.2	Teamgeschwindigkeit	114
7.2.1	Mit unbekannter Teamgeschwindigkeit planen	114
7.2.2	Schätzeinheit	116
7.2.3	Planungspoker	117
7.2.4	Unterschiedliche Teamgeschwindigkeiten	120
7.3	Iterationsplanung	121
7.3.1	Featureteams planen individuell	121
7.3.2	Planungstreffen	122
7.3.3	Termin für das Planungstreffen	123
7.3.4	Handfeste Planungswerkzeuge	124
7.4	Iterationsverfolgung	125
7.4.1	Planungs- und Nachverfolgungswerkzeuge	125
7.4.2	Ziele im Fokus behalten	126
7.5	Umgang mit Änderungen	128
7.5.1	Die Iterationslänge bestimmt das Antwortzeitverhalten	129
7.5.2	Mit Änderungswünschen umgehen	129
7.5.3	Die Teamstruktur verändern	130
7.6	Projektplan	131
7.6.1	Releaseplanung	131
7.6.2	Prognosen	133
7.6.3	Release versus Meilenstein	134
7.7	Zusammenfassung	135

8	Feedback	137
8.1	Den Kunden involvieren	138
8.1.1	Wer ist der Kunde?	138
8.1.2	Kunde auf Distanz	139
8.1.3	Kundenpräsentation	140
8.2	Rückblick	140
8.2.1	Iterationsrückblick	141
8.2.2	Rückblick – verteilt individuell versus gemeinsam persönlich	141
8.2.3	Releaserückblick	143
8.3	Retrospektive	144
8.3.1	Individuelle Featureteamretrospektive	146
8.3.2	Gemeinsame Projektteamretrospektive	147
8.3.3	Gemeinsame Standortretrospektive	149
8.3.4	Durchführung einer Retrospektive	149
8.3.5	Virtuelle Retrospektive	150
8.3.6	Teilnehmer	152
8.3.7	Weniger ist mehr und andere Tricks	153
8.3.8	Gegen die Langeweile – Moderationstechniken	154
8.4	Metriken	157
8.4.1	Fortschritt messen	158
8.4.2	Schätzqualität messen	160
8.4.3	Testbasis vergrößern	161
8.5	Zusammenfassung	161
9	Praktiken	163
9.1	Entwicklungspraktiken	163
9.1.1	Paarweises Programmieren	164
9.1.2	Unit Test	166
9.1.3	Refactoring	166
9.1.4	Gemeinsame Verantwortlichkeit	171
9.1.5	Gemeinsame Programmierrichtlinien	171
9.1.6	Features durch Tests übermitteln	172
9.1.7	Mühe mit Praktiken von der Stange	176
9.2	Prozesspraktiken	177
9.2.1	Tägliche Synchronisation	177
9.2.2	Projektweite Synchronisation	178
9.2.3	Verstreute Synchronisation	180

9.3	Entwicklungskultur	185
9.3.1	Gemeinsame Praktiken erfassen	185
9.3.2	Vereinbarte Praktiken ändern	186
9.3.3	Unterschiedliche Entwicklungskulturen	187
9.3.4	Standortübergreifend den Prozess mit CMMI oder ISO harmonisieren	188
9.3.5	Gleiches Recht für alle	190
9.4	Zusammenfassung	191
10	Agilität in ein verteiltes Projekt einführen	193
10.1	Lokal beginnen – global wachsen	194
10.1.1	Projektstart	195
10.1.2	Frühzeitige Iterationen	196
10.1.3	Ein Team – wechselnde Standorte	197
10.2	Teams und Standorte vergrößern	198
10.2.1	Projektstart in der Timebox	199
10.2.2	Kick-off	199
10.2.3	Die Projektkultur verbreiten	201
10.2.4	Kulturelles Training	201
10.2.5	Neue Mitarbeiter integrieren	203
10.3	Einen agilen Prozess in ein existierendes Projekt einführen	203
10.3.1	Einer nach dem anderen oder alle auf einmal	204
10.3.2	Veränderung der Teamstruktur	204
10.3.3	Mehr und bessere Coachs sind vonnöten	205
10.3.4	Schätzungen und Geschwindigkeit	206
10.3.5	Einzelkämpfer	207
10.4	Zusammenfassung	208
	Nachwort	209
	Glossar	211
	Referenzen	215
	Index	221

Vorwort

*Wir sind alle aufgerufen, pontifices zu sein – Brückenbauer.
Viele Flüsse haben schon ihre Übergänge.
An vielen stehen wir an verschiedenen Ufern
und suchen nach Pontons,
nach Stegen, nach Kommunikation.
Kunst und Technik trennt kein Meer,
oft nur Sprachlosigkeit.*

August Everding

Vor einigen Jahren, als ich mein letztes Buch über Agile Softwareentwicklung im Großen geschrieben habe, gab es nur wenige Menschen, die Agilität für größere Projekte, Teams oder Organisationen einsetzten. In der Zwischenzeit gibt es viele Personen, die festgestellt haben, dass Agilität sich nicht nur für kleine Teams eignet. Das heißt, es existiert zwischenzeitlich durchaus eine ansehnliche Menge an Großprojekten, die einen agilen Ansatz verfolgen.

Es gibt jedoch einen immensen Unterschied zwischen den (meist großen) Projekten, in denen ich früher gearbeitet habe, und den Projekten, in denen ich heute tätig bin. Die größte Diskrepanz besteht darin, dass scheinbar jedes große Projekt über mehrere Standorte verteilt ist. Und selbst kleine Projekte sind häufig nicht mehr nur an einem Standort ansässig.

Obwohl ich feststellte, dass sich ein Großteil meiner Erfahrungen mit der Skalierung agiler Prozesse ebenfalls auf verteilte Projekte übertragen lässt, gibt es jedoch gewisse komplexere Aspekte, die einer speziellen Aufmerksamkeit bedürfen, wenn Sie nicht die gesamte Agilität durch die Verteilung über den Globus verlieren möchten.

Im Zuge meiner Tätigkeit als Change Agent, Projekt-Coach oder Berater bei der Etablierung eines agilen Prozesses in diversen Projekten beobachtete ich bei mir als radikalste Veränderung die hohe Reisenotwendigkeit. Natürlich finde ich Reisen anstrengend, aber gleichzeitig auch spannend – sie bieten mir viele Gelegenheiten, mehr über unterschiedliche Gewohnheiten und Kulturen zu erfahren.

Ich hoffe, Sie finden dieses Buch hilfreich und haben Spaß am Lesen des Inhalts (vielleicht auf einer Ihrer Reisen zu einem Ihrer verteilten Teams). Außerdem möchte ich Sie auf die Webseite des Buches einladen: *www.verteilte-teams.de*. Ich bin gespannt auf Ihr Feedback unter *feedback@verteilte-teams.de*.

Jutta Eckstein
Braunschweig, Juni 2009

Danke

*Bücher liegen in der Luft.
Der Autor ist nur die Brücke
zwischen Stoff und Niederschrift.*

Marguerite Duras

Wenn man ein Projekt, wie das Schreiben eines Buches realisiert, so besteht eines der größten Probleme darin, sich an all die Menschen zu erinnern, die dieses Projekt unterstützt haben. Das ist, meines Erachtens nach, ein unlösbares Problem. Nahezu jeder, mit dem ich mich während oder bereits vor dem Start dieses Projekts unterhalten habe, unterstützte die Entstehung durch seine Erfahrungen oder durch sein Feedback zu meinen Erlebnissen zu diesem Buch. Sich an all diese Personen hier zu erinnern, ist fast unmöglich. Ich möchte mich vorab bei allen entschuldigen, die ich versehentlich hier nicht namentlich genannt habe.

Als Allererstes möchte ich allen Teams danken, mit denen ich die Chance hatte zusammenzuarbeiten. Das Miteinander in diesen Teams gab mir die Gelegenheit, überall auf der Welt viel zu lernen – über agile Softwareentwicklung, die Gemeinsamkeiten und Unterschiede in den Kulturen, Interaktionen und nicht zuletzt die Sitten, Gebräuche sowie die Vielfalt des Essens. Weiterhin möchte ich den zahllosen Menschen danken, die mir halfen zu reflektieren und somit das Erlernte greifbarer und expliziter zu machen. Ganz besonders möchte ich all den Reflektionspartnern danken, die ich als Teilnehmer in diversen Workshops, Vorträgen, Tutorials und Podiumsdiskussionen auf all den verschiedenen Konferenzen kennenlernte – wie ACCU in UK, Agile in Nordamerika, JA00 in Dänemark, Retrospektiven-Gathering in Nordamerika und in Europa sowie XP in Europa, um nur einige zu nennen.

Als Nächstes möchte ich all jenen danken, die ihre speziellen Erfahrungen in Form einer Expertenbox in diesem Buch zur Verfügung stellten und dadurch einen weiteren Blickwinkel auf das Thema ermöglichten. Danke – David Hussman, Naresh Jain, Nicolai Josuttis, Daniel Karlström, Michael Kircher, Debra S. Lavell, Ainsley Nies, Joseph Pelrine und Linda Rising.

Ich sage Danke an alle Reviewer, die sich enorm dafür eingesetzt haben, mein Erlertes in das Buch zu übertragen. Mein Dank geht an: Jamie Allsop, Joseph Bergin, Magnus Christerson, Lise Hvatum, Carsten Ruseng Jakobsen, Michael Kircher, Yi Lv, Ken Pugh und Bas Vodde.

Weiterhin möchte ich allen Mitarbeitern von dpunkt danken, die wieder einmal ihre Professionalität bei der Durchführung dieses Projekts bewiesen haben. Mein besonderer Dank geht an Christa Preisendanz.

Nicht zuletzt möchte ich meiner Familie Dank aussprechen, der es scheinbar nie langweilig wird, sich meine verschiedenen (Reise- und Arbeits-)Geschichten anzuhören und die mir auf diese Weise hilft, die Unterschiede und Gemeinsamkeiten der Menschen und deren Verhalten auf der ganzen Welt zu erkennen. Allen voran möchte ich meinem Partner Nico Josuttis danken, der mich nicht nur in meinem Berufsleben herausragend unterstützt, sondern, was noch wichtiger ist, mein Leben auf so eine wunderbare Art bereichert. Meiner Cousine Katja Gloggiessner gilt mein Dank, die mit diesen wunderschönen Illustrationen das Buch lebhafter gestaltet hat. Meiner Freundin Susanne Hornig möchte ich danken, die fast schon zur Familie gehört und die in ihrem »früheren« Leben Germanistik studiert hat. Dieses Wissen brachte sie in unermüdlicher Kleinarbeit ein und machte so mein oft umgangssprachliches Deutsch lesbar. Und schlussendlich danke ich meiner Schwester Eva Eckstein – erneut (wie bereits beim Schreiben meines ersten Buches) befolgte ich ihren Rat, mich auf der gleichen kleinen Insel (Hiddensee in der Ostsee) zu verstecken. Die Abgeschiedenheit und Stille dieses wunderschönen Orts haben sich auch dieses Mal positiv auf die Entstehung des Buches ausgewirkt.

1 Einführung

*Die Erfahrung
kommt den Lehren zuvor.*

Jean-Jacques Rousseau

Heutzutage gibt es nicht mehr viele Projekte, die an einem Standort, das heißt ohne Outsourcing, Offshoring oder eine sonstige Form von Verteilung, entwickelt werden. Jede Art von Verteilung impliziert aber, dass eine Distanz zwischen den Projektmitarbeitern existiert. Diese Distanz kann sich auf die Geografie, Zeit, Kultur, Sprache, Politik oder Geschichte beziehen. Viele verteilte Projekte vereinen sogar eine Kombination der genannten Distanzen.

Ein anderer Trend befindet sich momentan im Aufschwung – die agile Softwareentwicklung. Mehr und mehr Projekte betrachten Agilität als kritischen Erfolgsfaktor für die Softwareentwicklung. Agile Entwicklung betont die Kommunikation von Angesicht zu Angesicht sowie eine enge Zusammenarbeit zwischen allen Projektbeteiligten.

Distanz einerseits und enge Zusammenarbeit andererseits – wie passt das zusammen?

Trotz scheinbarer Inkompatibilität verteilter Softwareentwicklung und agiler Methoden haben einige Projekte versucht beides zu kombinieren. Das liegt auf der Hand, denn Kommunikationseinschränkungen und -hindernisse müssen bei verteilter Entwicklung erst recht adressiert werden, und agile Methoden fokussieren sich genau darauf. Sie legen besonderen Wert auf die Bedeutung von Kommunikation und Interaktion unter allen an der Entwicklung beteiligten Personen.

Das Agile Manifest (siehe [AgileManifesto]) und die zugrunde liegenden Prinzipien stellen die Durchführbarkeit von Agilität in einer verteilten Umgebung auch gar nicht infrage. Sie werden für den Einsatz agiler Entwicklung in einem verteilten Projekt vermutlich keine spezifische agile Methode finden, die Sie direkt »wie im Buch« verwenden können, aber die agilen Prinzipien können Ihnen helfen, den notwendigen Fokus in Ihrer Vorgehensweise zu behalten. So bleiben Sie agil und können das Leistungsvermögen der Ihnen zur Verfügung ste-

henden Kommunikationsmedien maximieren, unabhängig davon, wie eingeschränkt diese sind.

Dieses Buch basiert auf Erfahrungen in mehreren großen und verteilten agilen Projekten, deren Umfeld sich von eingebetteten bis zu kommerzieller Softwareentwicklung erstreckt. Diese Erfahrungen haben mich zu der Schlussfolgerung veranlasst, dass sowohl große als auch verteilte Teams von einem Wertesystem profitieren können, das für kleine und lokale Teams nützlich ist.

1.1 Fokus dieses Buches

In diesem Buch geht es um die Entwicklung von Software mit verteilten Teams; alternativ mit einem Team, das in sich verteilt ist (mehr zu dieser Unterscheidung im Abschnitt 2.1.2). Es geht nicht darum, einen Dienst oder eine Infrastruktur eines Offshore-Standortes anzubieten und diesen von einem Heimatstandort aus zu verwalten, wie dies beispielsweise für das Aufsetzen eines Callcenters notwendig wäre. Die Herausforderungen für verteilte (und noch extremer für Offshore-) Entwicklung bestehen im Vereinen von Teams aus den unterschiedlichsten Gegenden. Dabei muss sichergestellt werden, dass alle Teammitarbeiter standortunabhängig an einem Strang ziehen, das gleiche Ziel anvisieren, und zwar trotz physischer Distanz und kultureller Unterschiede. Dieses Buch geht genau auf diese Herausforderungen ein und zeigt auf, wie man die Probleme, die verteilte Teams haben, lösen kann.

1.1.1 Projekterfahrung

Ich habe meine hauptsächlichen Erfahrungen in *großen* verteilten Projekten gesammelt. Typischerweise arbeiten in meinen Projekten fünfzig bis dreihundert Mitarbeiter, die über drei bis fünf Standorte verteilt sind. Einige meiner Projekte waren nur mäßig verteilt zum Beispiel über vier Standorte, aber alle innerhalb Europas. Andere waren stark verteilt, indem sie sich beispielsweise von Südamerika über Europa bis nach Asien erstreckten. Die Domänen dieser Projekte variierten stark – einige konzentrierten sich auf kommerzielle Anwendungen, die meisten aber auf technische, zum Beispiel auf eingebettete Systeme.

Sie werden feststellen, dass Sie selbst dann die Hinweise des Buches nutzen können, wenn Ihr Projekt weniger verteilt und kleiner ist. Das heißt, selbst dann, wenn Ihr Projekt beispielsweise aus zehn Mitarbeitern besteht, die von ihrem Home-Office aus arbeiten, oder aus zwanzig Mitarbeitern, die über mehrere Etagen verteilt sind, finden Sie hier wertvolle Tipps.

1.1.2 Wem hilft dieses Buch

Dieses Buch richtet sich an Personen, die nach Möglichkeiten suchen, flexibler und agiler zu sein, obwohl sie in einer verteilten Umgebung arbeiten. Ich biete zwar einige Grundlagen zu agiler Entwicklung an, es ist jedoch hilfreich, wenn Sie sich das Grundwissen auf diesem Gebiet angeeignet haben.

Für diejenigen, die bereits mit agiler Entwicklung vertraut sind und diese in einer lokalen Umgebung anwenden, aber unsicher sind, wie Agilität in einer verteilten Umgebung gelebt werden kann – präsentiert dieses Buch eine Leitlinie zur Anpassung der Agilität an solch eine Umgebung.

Ich gehe davon aus, dass Sie ein sogenannter Change Agent sind. Also jemand, der Veränderungen forcieren und von beiden Welten profitieren möchte – der agilen und der verteilten. Als Change Agent arbeiten Sie vermutlich in einem verteilten Team als Projektleiter, Prozess-Coach, Kundenvertreter, Berater oder Entwickler. Dieses Buch wird darüber hinaus für Sie von Interesse sein, wenn Sie:

- agile Methoden in verteilten Projekten erfolglos ausprobiert haben,
- agile Methoden in verteilten Projekten erfolgreich eingesetzt haben,
- agile Methoden bisher nicht in verteilten Projekten verwendet haben, dies aber planen,
- fest an die Agilität glauben, jedoch starke Zweifel an deren Erfolg in einer verteilten Umgebung haben.

1.1.3 Was dieses Buch nicht beinhaltet

In vielen bislang erschienenen Büchern finden Sie Ratschläge zu Sinn und Unsinn des Einsetzens von z.B. Offshoring. Dieses Buch behandelt dieses Thema nicht. Es setzt Verteilung als gegeben voraus¹ und stellt sie nicht infrage, auch aus demselben Grund, den Jan-Erik Sandberg und Lars Arne Skår in [SandbergSkår07] angeben:

Offshoring ist hier und bleibt hier. Wir müssen damit leben und versuchen das Beste daraus zu machen.

Eine derartige Entscheidung lässt Sie mit der Frage zurück, wie Sie trotz eines verteilten Teams agil sein können. Die Antworten auf diese Frage werden Sie in diesem Buch finden.

Auch werde ich nicht über Ängste des Arbeitsplatzverlustes sprechen, die einige Menschen an ihren Heimatstandorten aufgrund von Offshoring bewegen. Ich werde jedoch zur Sprache bringen, inwiefern sich die Arbeit an den verschie-

1. Bücher, die solche Ratschläge erteilen, sind zum Beispiel [Corbett04], [Friedman05], [LacityWillcocks01], [Nicklisch08] und [ThondavadiAlbert04].

denen Standorten ändern wird und wie man über mehrere Standorte (und eventuell sogar Kontinente) hinweg effektiv zusammenarbeiten kann.

1.2 Fahrplan für das Buch

Zur optimalen Nutzung dieses Buches habe ich einen Überblick zusammengestellt, der Ihnen als Fahrplan dienen kann:

- **Kapitel 2, Agilität und verteilte Projekte,**
legt die Basis für das Verständnis von sowohl verteilter als auch agiler Softwareentwicklung. Dieses Kapitel erläutert, wodurch sich ein verteiltes Projekt auszeichnet und was ein agiles Projekt charakterisiert. Des Weiteren zeigt es, welchen Einfluss die agilen Prinzipien auf die verteilte Entwicklung haben.
- **Kapitel 3, Teambildung,**
bietet Erläuterungen, wie Teams in einer verteilten agilen Umgebung strukturiert sein sollten. Ich werde erklären, inwiefern Featureteams Ihr Projekt in die Lage versetzen, immer den höchsten Geschäftswert während der Projektlaufzeit zu liefern. Die Gewährleistung von konzeptioneller Integrität mit Featureteams und Projektrollen stellt in diesem Kapitel das Hauptthema.
- **Kapitel 4, Kommunikation und Vertrauen,**
spricht über das Entstehen vertrauensvoller Beziehungen zwischen den Standorten und die Wichtigkeit des gegenseitigen Respekts unter allen Projektmitarbeitern. Da Vertrauen Kontakt erfordert, werde ich aufzeigen, wie Sie durch standortübergreifende Kommunikation trotz der Distanz Nähe schaffen können. Schlussendlich betrachtet dieses Kapitel, welche Auswirkungen die kulturellen Unterschiede auf Kommunikation und Vertrauen haben.
- **Kapitel 5, Standorte in Verbindung halten,**
konzentriert sich auf die speziellen Rollen und Verantwortlichkeiten, die zur Wahrung einer guten Arbeitsbeziehung zwischen allen Standorten beitragen. Sehr hilfreich sind dafür Personen, die die Rolle des Kommunikationsvermittlers übernehmen und zu allen Standorten reisen, und Botschafter, die ihren Heimatstandort an einem fremden Standort repräsentieren. Sozialkontakte und virtuelle Werkzeuge ermöglichen eine dauerhafte Verbindung zwischen den Standorten.
- **Kapitel 6, Entwickeln und Ausliefern,**
erläutert die agilen Entwicklungszyklen, die Planung kurzfristiger Iterationen und langfristiger Releases. Da der Fokus auf der Auslieferung eines kohärenten Systems liegt, werden in diesem Kapitel neben dem Thema Infrastruktur auch die Themen Integration und Build in verteilten Umgebungen vertieft.

- **Kapitel 7, Der Geschäftswert ist das Ziel,**
präsentiert, wie die Entwicklung über Features angesteuert und dabei immer der höchstmögliche Geschäftswert zu jedem Zeitpunkt priorisiert wird. Sie müssen die Projekt- bzw. Entwicklungsgeschwindigkeit kennen, um eine Aussage über zeitlich präzise Lieferungen treffen zu können. Diese kann zur Planung und Nachverfolgung sowohl der Iterationen als auch des Projektplans verwendet werden. Darüber hinaus thematisiert dieses Kapitel, wie wir mit geänderten Anforderungen umgehen und trotzdem in der Lage sind, regelmäßig zu liefern und dabei weiterhin den höchsten Geschäftswert im Fokus zu behalten.
- **Kapitel 8, Feedback,**
stellt den Kunden und sein Feedback in den Vordergrund. Es wird aufgezeigt, wie der Iterationsrückblick zur Erhaltung von Feedback über das echte System verwendet werden kann und inwiefern Retrospektiven zur ständigen Prozessverbesserung beitragen. Außerdem werden verschiedene Metriken vorgestellt, die Ihnen im Hinblick auf Fortschritts- und Qualitätsmessung Hilfestellungen geben.
- **Kapitel 9, Praktiken,**
unterstreicht Entwicklungs- und Prozesspraktiken, die Ihnen helfen, agil zu bleiben. Ebenso werden Techniken dargelegt, die Sie dabei unterstützen, Praktiken an Ihre Bedürfnisse anzupassen bzw. neue Praktiken zu entwickeln. Beides wird dazu beitragen, dass Sie eine Entwicklungskultur in Ihrem Projekt etablieren und bewahren. Inwiefern CMMI dabei behilflich oder hinderlich ist, klärt dieser Abschnitt.
- **Kapitel 10, Agilität in ein verteiltes Projekt einführen,**
erklärt, wie man mit Agilität in einer verteilten Umgebung startet. Wenn Sie mit einem neuen Projekt beginnen, empfehle ich, lokal zu starten und global/verteilt zu wachsen. Wir werden betrachten, wie Sie sowohl die Teams als auch die Projektstandorte vergrößern können. Abschließend finden Sie eine Erläuterung zur Einführung einer agilen Vorgehensweise in ein bereits laufendes Projekt.

Am Ende des Buches finden Sie ergänzend ein Glossar und Referenzen mit Hinweisen zu weiterführender Literatur in Form von Artikeln, Büchern und URLs.

2 Agilität und verteilte Projekte

Alle Dinge sind miteinander verbunden.

Rede des Indianerhäuptlings Seattle

2.1 Verteilte Entwicklung verstehen

Mein Supermarkt an der Ecke warb neulich mit der folgenden Anzeige:

Computer-Spezialisten findet man in Indien.

Lebensmittel-Spezialisten direkt um die Ecke.

Daraus kann man die Vorstellung (oder das Missverständnis) ableiten, dass man hier für IT keine Experten benötigt, jedoch aber für Lebensmittel. Ich möchte gar nicht für irgendeine Seite Partei ergreifen (zumal es vermutlich auch in Indien Lebensmittel-Spezialisten gibt), auf alle Fälle ist es um einiges einfacher, lokal nach Experten Ausschau zu halten als global. Natürlich ist gerade das häufig ein Grund für das Aufsetzen eines verteilten Projekts, da man oftmals sogar darauf angewiesen ist, an mehr als einer Stelle nach talentierten Leuten zu suchen. Will man jedoch mit einem verteilten Projekt erfolgreich sein, gilt es, mehr zu tun, als sich lediglich Experten von überall zusammenzusuchen. Klar, des Öfteren wird auch die Meinung vertreten, dass die Entscheidung für ein (global) verteiltes Projekt lediglich bedeutet, dass dieses Projekt im Falle eines Scheiterns einfach ein billigerer Fehlschlag wird, als wenn dieses Projekt lokal durchgeführt werden würde. Wenn das tatsächlich Ihr Ziel sein sollte – dann reicht es aus, sich lediglich nach (günstigen) Arbeitskräften umzuschauen.

Wie im Supermarkt brauchen Sie auch als Kunde zumindest einen lokalen Kontakt, um tatsächlich an Ihr Essen zu kommen. Und ebenfalls vergleichbar mit der Nahrungskette sind immer mehrere Orte involviert. Jedoch bedeutet Verteilung, dass im Prinzip nichts mehr lokal ist. Verteilte Softwareentwicklung kann mehrere Standorte, in sich verteilte – verstreute – Teams, Beteiligung mehrerer Firmen und Off-site-Kunden implizieren.

2.1.1 Beteiligung mehrerer Entwicklungsstandorte

Typischerweise sind bei verteilter Entwicklung mehrere Entwicklungsstandorte beteiligt. Deshalb ist es nicht damit getan, dass man einfach ein paar Experten irgendwo lokalisiert, sondern man muss stattdessen überall nach Experten suchen und dann deren Wissen virtuell zusammenbringen. Diese Experten sind meist über mehrere Länder verstreut.

Die meisten Schwierigkeiten sind genau auf diesem Szenario begründet. Die Experten müssen über verschiedene Standorte hinweg zusammenarbeiten. Doch diese Kooperation wird umso schwieriger, je mehr dieser Standorte sich auch noch über mehrere Länder verteilen, da dann die unterschiedlichen Kulturen, Zeitzonen, Sprachen usw. die Zusammenarbeit zusätzlich erschwert.

2.1.2 Verteilte und verstreute Teams

Verteilte Entwicklung bedeutet im Kern, dass mehrere Teams über mehrere Standorte verteilt sind. Teams können dabei auf zweierlei Arten verteilt sein:

- **Verteilte Teams** (englisch: distributed team)

Das bedeutet, dass sich zum Beispiel ein Team in Indien befindet und ein anderes in den USA. In diesem Fall würde die Entwicklung von zwei Teams bewerkstelligt werden, wobei die Teamgrenzen durch die unterschiedlichen Standorte definiert werden. An jedem Standort arbeitet beispielsweise ein Team an einem Teilsystem.

- **Verstreute Teams** (englisch: dispersed team)

Ein Team kann auch in sich verteilt sein. Dies ist dann der Fall, wenn sich beispielsweise ein Teammitglied in Indien befindet, ein anderes in Nordirland, ein drittes in den USA, und das vierte sitzt in Russland, aber alle vier arbeiten zusammen in einem Team. Im vorliegenden Fall wird die Entwicklung von einem Team durchgeführt, dessen Teammitglieder über mehrere Standorte verstreut sind. Weder werden Teamgrenzen noch Teilsysteme über die Standorte definiert.

Meistens findet man in großen verteilten Projekten einen Mix von einigen klassisch verteilten Teams, wobei komplette Teams an verschiedenen Standorten sitzen, und einigen verstreuten Teams, bei denen die Teammitglieder über mehrere Standorte verstreut sind.

Zur Sicherstellung der Kommunikation unter allen Teammitarbeitern verlangt ein verstreutes Team einen hohen Koordinierungsaufwand, selbst wenn es sich dabei um ein kleines Team handelt. Der größte Aufwand muss in das Selbstverständnis des Teams gesteckt werden – das heißt, dass sich die individuellen Teammitglieder tatsächlich als ein Team verstehen und gemeinsam das gleiche Ziel anvisieren.

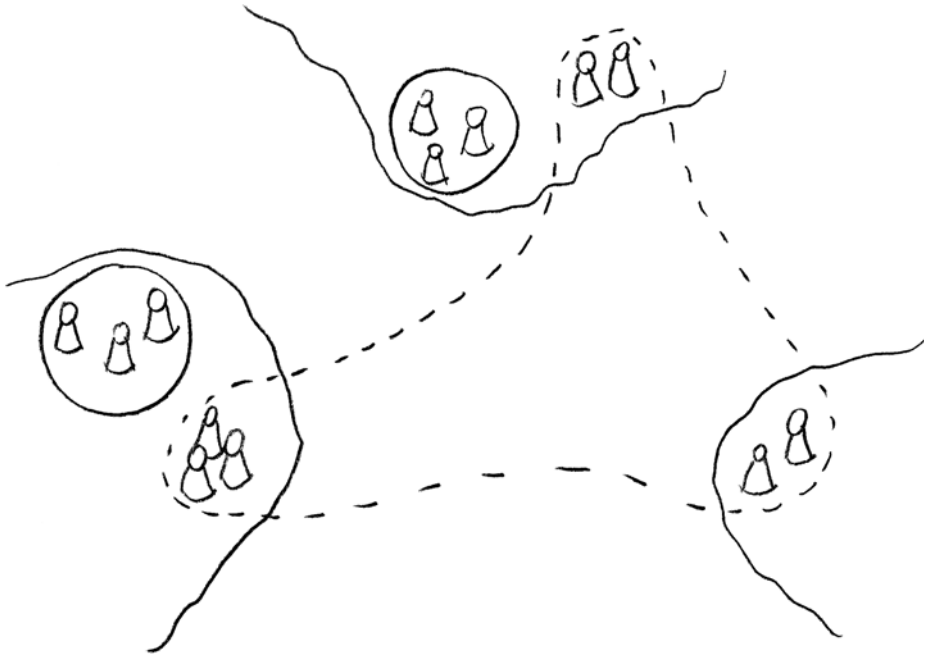


Abb. 2-1 Verstreute Teams

2.1.3 Große Projekte

Die meisten verteilten Projekte sind etwas größer. Wie ich in [Eckstein04] erläutere, gibt es mehrere Dimensionen, die die Größe eines Projekts beeinflussen. Diese Dimensionen spielen in verteilten Projekten eine wichtige Rolle: zum Beispiel die Dimension der Personenanzahl – in den meisten verteilten Projekten ist mehr als ein Team involviert. Oder die Dimension der Dauer – es gibt kaum ein verteiltes Projekt, das lediglich drei Monate dauert. Der Aufwand, eine verteilte Umgebung aufzusetzen, ist dafür einfach zu hoch.

Allerdings gibt es auch einige verteilte Projekte, die nicht groß, aber dafür sehr stark verstreut sind. In diesen Projekten sitzt nahezu jedes Teammitglied weit voneinander entfernt an einem anderen Ort der Welt.

Große Projekte implizieren immer schon eine Menge Risiken. Eines davon ist die Koordination der verschiedenen Mitarbeiter und der Teams. Ein anderes Risiko ist, dass das Projekt trotz seiner Größe und dem damit verbundenen Overhead flexibel und effizient bleibt. Dazu gehört, dass alle Projektmitarbeiter das gleiche große Gesamtbild über das Projekt im Kopf haben.

2.1.4 Beteiligung verschiedener Firmen

In vielen verteilten Entwicklungen sind mehr als eine Firma beteiligt. Einige Firmen kaufen dabei die Firma des anderen Entwicklungsstandorts, mit der sie zusammenarbeiten wollen, oder sie gründen direkt ein Tochterunternehmen dort. Auf diese Weise setzen diese Unternehmen auf eine dauerhafte Kooperation und betrachten verteilte Entwicklung als eine langfristige Strategie. Andere Firmen sehen in der verteilten Entwicklung eine Chance, sich auf ihre Kernkompetenzen zu konzentrieren und peripheres Fachwissen an andere Firmen auszulagern. Für wieder andere ist die verteilte Entwicklung ganz einfach eine Möglichkeit, besser an verschiedene Fachkenntnisse heranzukommen und sich nicht nur auf fähiges Personal in ihrer Region zu beschränken.

Die Beteiligung verschiedener Firmen erfordert eine Menge Arbeit hinsichtlich des Aufbaus einer guten und vor allem vertrauensvollen Beziehung zwischen den verschiedenen Unternehmen. Obwohl natürlich auch bei verschiedenen Tochterunternehmen gegenseitiger Respekt fehlen kann, so ist es doch hauptsächlich bei unterschiedlichen Unternehmen zu beobachten. Normalerweise werden Verträge verwendet, um das Vertrauen zwischen den beteiligten Firmen zu untermauern. Der Vertrag dient dazu, im Detail zu klären, was die zwei (oder mehr) Parteien voneinander erwarten, speziell im Fall, dass etwas schief läuft.

Ein Vertrag ist jedoch als formales Mittel nur beschränkt tauglich, da er nur einen Schutz vor einem möglichen Vertrauensverlust für die beteiligten Parteien suggeriert, indem für den entsprechenden Fall mit Strafen gedroht wird. Obwohl die Absicht des Vertrages die Begründung einer erfolgreichen und vertrauensvollen Geschäftsbeziehung ist, kann ein Vertrag nie das Mittel sein, um tatsächlich solch eine vertrauensvolle Verbindung *herzustellen und zu bewahren*, wie sie für eine erfolgreiche Zusammenarbeit notwendig ist.

Auf keinen Fall habe ich die Absicht vorzuschlagen, dass Sie ohne Vertrag solch eine Zusammenarbeit starten sollten, vielmehr möchte ich hervorheben, dass es viel mehr als einen Vertrag braucht, um eine erfolgreiche Beziehung zu etablieren. Lean (z.B. [Liker04] oder [Poppendieck03]), eine der Wurzeln agiler Entwicklung, favorisiert partnerschaftliche Beziehung, wie auch Carsten Ruseng Jakobsen von Systematic anmerkt:

Toyota hat z.B. bewiesen, dass es auf lange Sicht profitabler ist, Subunternehmer als Partner zu behandeln, als ständig den günstigsten Preis zu erstreben. Ich glaube, dass viele Outsource-Projekte die gleiche Herausforderung haben. Obwohl man einen Vertrag zwischen den verschiedenen Teilen des Projekts hat, möchte man trotzdem, dass das gesamte Projekt gemeinsam auf die gleiche Vision hinarbeitet.