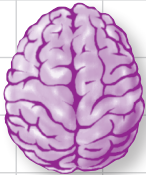


Agile Methoden

von Kopf bis Fuß



Laden Sie sich Agile
Konzepte direkt ins Hirn

Kommandieren
Sie Ihren Code
mit XP herum



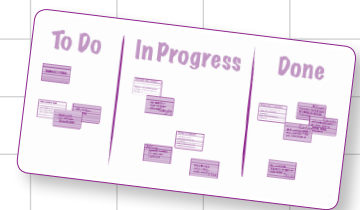
Lösen Sie die Rätsel von
Lean und Kanban



Ein Wegbegleiter für alle,
die Agile Methoden verstehen
und anwenden wollen



Erfahren Sie, wie Amy
ihre Anwender mit
Scrum bei der Stange
hielt



Finden Sie heraus,
wie Taskboards alle auf
Kurs halten

Andrew Stellman & Jennifer Greene

Deutsche Übersetzung von Thomas Demmig

Agile Methoden von Kopf bis Fuß

Wäre es nicht wunderbar, wenn es ein Buch gäbe, das mir dabei hilft, Agile zu erlernen, und das **mehr Spaß macht als ein Besuch beim Zahnarzt?** Aber das wird wohl für immer ein Traum bleiben ...



**Andrew Stellman
Jennifer Greene**

**Deutsche Übersetzung von
Thomas Demmig**

O'REILLY®

Andrew Stelman und Jennifer Greene

Lektorat: Ariane Hesse

Übersetzung: Thomas Demmig

Korrektorat: Sibylle Feldmann, www.richtiger-text.de

Fachgutachten: Silke Notheis

Satz: Ulrich Borstelmann, www.borstelmann.de

Herstellung: Stefanie Weidner

Umschlaggestaltung: Karen Montgomery, Louise Barr, Michael Oréal (www.oreal.de)

Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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.

ISBN:

Print 978-3-96009-079-3

PDF 978-3-96010-207-6

ePub 978-3-96010-208-3

mobi 978-3-96010-209-0

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.

O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

1. Auflage 2019

Copyright © 2019 by dpunkt.verlag GmbH

Wiebinger Weg 17

69123 Heidelberg

Authorized German translation of the English edition of *Head First Agile*, ISBN 978-1-449-31433-0 © 2017 Andrew Stelman and Jennifer Greene.
This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

PMI-ACP, PMP und PMBOK sind eingetragene Marken des Project Management Institute, Inc.

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.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

5 4 3 2 1 0

Papier
plus⁺
PDF.

Zu diesem Buch – sowie zu vielen weiteren O'Reilly-Büchern –
können Sie auch das entsprechende E-Book im PDF-Format
herunterladen. Werden Sie dazu einfach Mitglied bei oreilly.plus⁺:

www.oreilly.plus

Für Nisha und Lisa

Vielen Dank, dass Sie unser Buch gekauft haben! Wir schreiben wirklich gern so ein Zeug und hoffen, das Lesen macht Ihnen Spaß ...

Andrew

Foto von Nisha Sondhe

... denn wir wissen, dass Sie mit Agile tolle Arbeit leisten werden!

Jenny



Andrew Stellman ist Entwickler, Architekt, Redner, Trainer, Agile-Coach, Projektmanager und Experte für das Bauen besserer Software. Andrew ist Autor und ein internationaler Redner, dessen Bücher zu Softwareentwicklung und Projektmanagement sich ausgezeichnet verkaufen, außerdem ist er ein weltweit anerkannter Experte für das Umwandeln und Verbessern von Softwareorganisationen, Teams und Code. Er hat große Softwaresysteme entworfen und gebaut, große internationale Softwareteams geleitet und Firmen, Schulen und Vereinigungen beraten, unter anderem Microsoft, das National Bureau of Economic Research, die Bank of America, Notre Dame und das MIT. Andrew hatte die Ehre, dabei mit einigen wirklich großartigen Programmierern zusammenarbeiten zu können, und glaubt fest daran, dabei das eine oder andere von ihnen gelernt zu haben.

Jennifer Greene betreut Agile-Transformationen in Großunternehmen, arbeitet als Agile-Coach, Entwicklungsmanagerin, Projektmanagerin, Rednerin, und sie ist eine Autorität in Sachen Praktiken und Prinzipien der Softwareentwicklung. Sie baut seit über 20 Jahren Software in unterschiedlichsten Bereichen, zum Beispiel Medien, Finanz und IT-Consulting. Sie hat große Agile-Transformationen von Entwicklungsteams auf der ganzen Welt betreut und Teammitgliedern dabei geholfen, durch Agile-Praktiken so erfolgreich wie möglich zu werden. Jenny freut sich darauf, weiterhin mit talentierten Teams interessante und schwierige Probleme zu lösen.

Jenny und Andrew haben Software gebaut und darüber geschrieben, seit sie sich das erste Mal 1998 trafen. Ihr erstes Buch *Applied Software Project Management* wurde von O'Reilly im Jahr 2005 veröffentlicht. Ihre ersten Bücher der »Von Kopf bis Fuß«-Reihe – *Head First PMP* und *Head First C#* (in Deutsch: *C# von Kopf bis Fuß*) erschienen beide in 2007. Diese Bücher gibt es mittlerweile in dritter und bald in vierter Auflage. Ihr viertes Buch, *Beautiful Teams*, erschien 2009, gefolgt vom fünften Buch *Learning Agile* im Jahr 2014.

2003 gründeten sie Stellman & Greene Consulting – ihr erstes Projekt als Beratungsfirma war ein wirklich faszinierendes Softwareprojekt für Wissenschaftler, die die Herbizid-Belastung bei Vietnam-Veteranen untersuchten. Und wenn sie keine Software schreiben oder an Büchern arbeiten, halten sie viele Vorträge auf Konferenzen und Meetings von Softwareentwicklern, Architekten und Projektmanagern.

Werfen Sie einmal einen Blick auf ihre Website »*Building Better Software*« unter <http://www.stellman-greene.com>.

Der Inhalt (im Überblick)

	Einführung	xv
1	Was ist Agile? <i>Prinzipien und Praktiken</i>	1
2	Agile Werte und Prinzipien: <i>Mindset trifft Methode</i>	23
3	Projekte mit Scrum managen: <i>Die Scrum-Regeln</i>	71
4	Agiles Planen und Schätzen: <i>Allgemein anerkannte Scrum-Praktiken</i>	117
5	XP (Extreme Programming): <i>Veränderungen willkommen heißen</i>	177
6	Lean/Kanban: <i>Waste ausmerzen und den Flow managen</i>	245
7	Vorbereitung auf das PMI-ACP® Exam: <i>Testen Sie Ihr Wissen</i>	307
8	Berufliche Verantwortung: <i>Gute Entscheidungen</i>	377
	Index	391
<i>Bonuskapitel</i> zum Download	Ein PMI-ACP-Examen zum Üben (Sie finden das PDF auf <i>oreilly.de</i> auf der Webseite zum Buch und dort unter <i>Zusatzmaterial</i>)	

Inhalt (jetzt ausführlich)

Einführung

Ihr Gehirn in agil. Sie versuchen, etwas zu lernen, und Ihr Hirn tut sein Bestes, damit das Gelernte nicht hängen bleibt. Es denkt nämlich: »Wir sollten lieber ordentlich Platz für wichtigere Dinge lassen, z. B. für das Wissen darüber, welche Tiere einem gefährlich werden könnten, oder dass es eine ganz schlechte Idee ist, nackt Snowboard zu fahren.« Tja, wie schaffen wir es nun, Ihr Gehirn davon zu überzeugen, dass Ihr Leben davon abhängt, zu wissen, wie Sie wirklich »agil« werden – und vielleicht sogar das PMI-ACP®-Examen bestehen?

Für wen ist dieses Buch?	xvi
Und wir wissen, was Ihr Gehirn gerade denkt.	xvii
Metakognition: Nachdenken übers Denken	xix
Und das können SIE tun, um sich Ihr Gehirn untertan zu machen	xxi
Das Team für das technische Review	xxiii
Danksagungen	xxiv

Was ist Agile?

1

Prinzipien und Praktiken

Es ist so eine spannende Zeit, agil zu sein! Zum ersten Mal hat unsere Branche einen echten, nachhaltigen Weg gefunden, Probleme zu lösen, mit denen Generationen von Softwareentwicklungsteams gekämpft hatten. Agile Teams verwenden einfache, gradlinige **Praktiken**, die sich schon in realen Projekten bewährt haben. Aber Moment mal ... wenn Agile so toll ist, warum nutzt es dann nicht schon jeder? Nun, in der Welt da draußen kann es passieren, dass eine Praktik, die für ein Team gut funktioniert, bei einem anderen Team zu echten Problemen führt – der Unterschied liegt im **Mindset** des Teams. Machen Sie sich also bereit, Ihre Denkweise über Projekte zu verändern!



Das neue Feature hört sich toll an ...	2
... aber es läuft nicht immer wie erwartet	3
Agile eilt zur Hilfe!	4
Kate versucht, ein Daily Standup abzuhalten	5
Verschiedene Teammitglieder verhalten sich unterschiedlich	6
Ein besseres Mindset führt zu einer besser funktionierenden Praktik	8
Was ist denn dann nun Agile?	10
Scrum ist das am häufigsten genutzte agile Konzept	12
Werden Sie agiler mit der PMI-ACP-Zertifizierung	18

In einem Daily Standup stehen alle Teammitglieder während des gesamten Meetings. So bleibt es kurz, nett und fokussiert.



Aber achtet der da wirklich darauf, was seine Kollegen sagen?

2 Agile Werte und Prinzipien

Mindset trifft Methode

Es gibt kein »perfektes« Rezept, um großartige Software zu bauen.

Manche Teams haben nach der Übernahme agiler Praktiken, Methoden und Vorgehensweisen viel Erfolg, während andere Probleme bekommen. Wir haben gelernt, dass der Unterschied im Mindset der Teammitglieder liegt. Was tun Sie also, wenn Sie diese großartigen agilen Ergebnisse in Ihrem eigenen Team erreichen wollen? Wie stellen Sie sicher, dass Ihr Team die richtige Denkweise besitzt? Hier kommt das **Agile Manifest** ins Spiel. Wenn Sie und Ihr Team dessen **Werte und Prinzipien** verinnerlichen, beginnen Sie, anders über die agilen Praktiken und ihre Funktionsweise zu denken, und diese werden nun *viel effektiver*.

In Snowbird ist etwas Wichtiges passiert	24
Das Agile Manifest	25
Es kann eine Herausforderung sein, Praktiken in der Realität umzusetzen	26
Individuen und Interaktion mehr als Prozesse und Werkzeuge	27
Funktionierende Software mehr als umfassende Dokumentation	28
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung	31
Reagieren auf Veränderung mehr als das Befolgen eines Plans	32
Frage-Sprechstunde: Die Frage »Which is the BEST«	36
Sie glauben, einen Volltreffer gelandet zu haben ...	38
... aber es ist ein Flop!	39
Die Prinzipien hinter dem Agilen Manifest	40
Die agilen Prinzipien helfen Ihnen dabei, Ihr Produkt auszuliefern	42
Die agilen Prinzipien helfen Ihrem Team dabei, zu kommunizieren und zusammenzuarbeiten	52
Das neue Produkt ist ein Hit!	56
Prüfungsfragen	58



Projekte mit Scrum managen

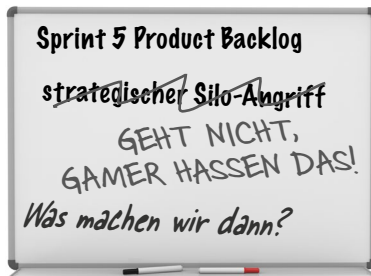
3

Die Scrum-Regeln

Die Scrum-Regeln sind einfach. Sie effektiv einzusetzen, ist nicht so einfach. Scrum ist die am häufigsten genutzte Vorgehensweise von Agile, und das aus gutem Grund: Die **Scrum-Regeln** sind einfach und leicht zu erlernen. Die meisten Teams brauchen nicht viel Zeit dafür, die **Ereignisse, Rollen und Artefakte** zu verstehen, die die Scrum-Regeln ausmachen. Aber damit Scrum richtig effektiv ist, müssen sie die **Werte von Scrum** und die Prinzipien des Agilen Manifests wirklich verstehen, wodurch sie leichter ein effektiveres Mindset entwickeln. Denn auch wenn Scrum einem einfach erscheint, ist die Art und Weise, wie ein Scrum-Team durchgehend **untersucht und anpasst**, eine ganz neue Art, über Projekte zu denken.



Lernen Sie das Team von Ranch Hand Games kennen	73
Die Scrum-Ereignisse helfen Ihnen dabei, Ihre Projekte zu erledigen	74
Mit den Scrum-Rollen verstehen Sie, wer was tut	75
Die Scrum-Artefakte halten das Team informiert	76
Die Scrum-Werte machen das Team effektiver	82
Frage-Sprechstunde: Die »Was kommt als Nächstes«-Frage	90
Eine Aufgabe ist erst erledigt, wenn »Done« erledigt ist	92
Scrum-Teams nehmen Rücksicht auf Änderungen während des Sprints	93
Das Agile Manifest hilft Ihnen, Scrum wirklich zu verstehen	96
Es sieht gut aus für das Team	102
Prüfungsfragen	104



Mit einem neuen Product Owner sollte das Team die wichtigsten Features für den nächsten Sprint herausfinden können.

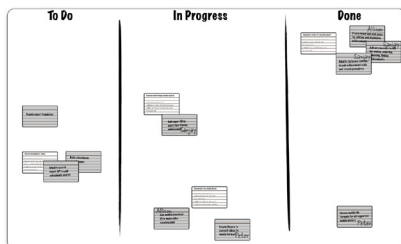
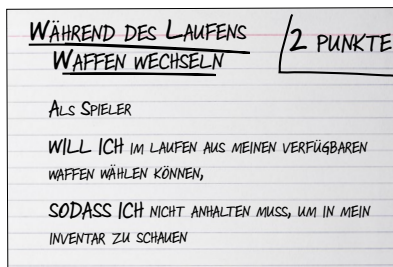


Agiles Planen und Schätzen

4

Allgemein anerkannte Scrum-Praktiken

Agile Teams nutzen einfache Planungswerkzeuge, um ihre Projekte im Griff zu halten. Scrum-Teams planen ihre Projekte gemeinsam, sodass sich jeder im Team jedem Sprint-Ziel verpflichtet fühlen kann. Um die **kollektive Verpflichtung** des Teams zu erreichen, müssen Planen, Schätzen und Nachverfolgen für das gesamte Team einfach und problemlos machbar sein. Von **User Stories** und **Planning Poker** bis hin zu **Velocity** und **Burndown Charts** wissen Scrum-Teams immer, was sie getan haben und was noch zu tun ist. Machen Sie sich bereit, die Werkzeuge kennenzulernen, die Scrum-Teams informiert halten und ihnen immer zeigen, was sie gebaut haben!



Währenddessen auf der Ranch ...	118
Also ... was kommt als Nächstes?	121
Dürfen wir vorstellen: GASPs!	122
Bitte keine weiteren 300-Seiten-Spezifikationen ...	124
User Stories lassen Teams verstehen, was Anwender brauchen	125
Mit Story Points fokussiert sich das Team auf die relative Größe jeder Story	126
Das gesamte Team schätzt zusammen	132
Keine detaillierten Projektpläne mehr	134
Taskboards halten das Team informiert	136
Frage-Sprechstunde: Der rote Hering	140
Burndown Charts sorgen dafür, dass das Team sieht, wie viel Arbeit noch zu tun ist	143
Die Velocity zeigt Ihnen, wie viel Ihr Team in einem Sprint schaffen kann	144
Burnups halten Ihren Fortschritt und Ihren Scope voneinander getrennt	147
Woher wissen wir, was gebaut werden soll?	148
Story Maps helfen beim Priorisieren Ihres Backlog	149
Personas helfen Ihnen, Ihre Anwender kennenzulernen	150
Die Nachrichten könnten besser sein ...	152
Retrospektiven helfen Ihrem Team, seine Arbeitsweise zu verbessern	154
Werkzeuge, die Ihnen dabei helfen, mehr aus Ihren Retrospektiven herauszuholen	156
Prüfungsfragen	164

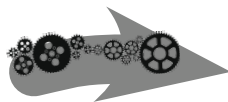
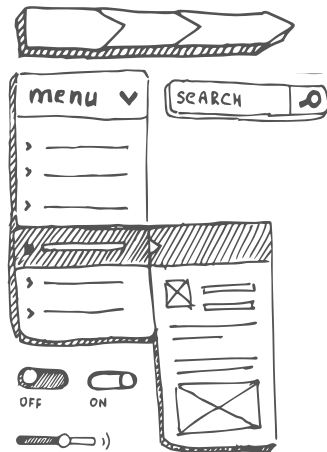
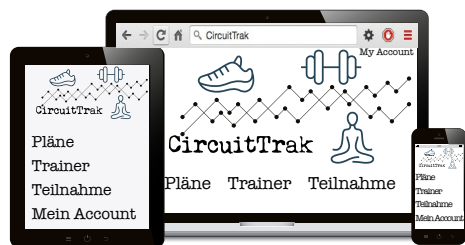
XP (Extreme Programming)

5

Veränderungen willkommen heißen

Softwareteams sind erfolgreich, wenn sie guten Code bauen.

Auch sehr gute Softwareteams mit ausgesprochen talentierten Entwicklern haben Probleme mit ihrem Code. Wenn sich kleine Änderungen am Code zu einer Reihe **aufeinanderfolgender Hacks** entwickeln oder jeder Code-Commit zu stundenlangen Merging-Orgien führt, wird Arbeit, die *eigentlich Spaß macht*, **nervig, mühsam und frustrierend**. Und hier kommt **XP** ins Spiel. XP ist eine agile Vorgehensweise, die sich darauf konzentriert, Teams aufzustellen, die zusammenhalten, gut **kommunizieren** und eine **entspannte, energiegeladene Umgebung** schaffen. Wenn Teams Code bauen, der nicht komplex, sondern **einfach** ist, können sie **Änderungen willkommen heißen**, statt sie zu fürchten.

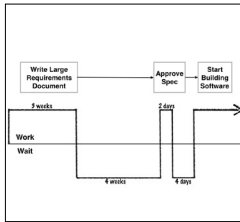


Dürfen wir vorstellen: das Team von CircuitTrak	178
Lange Nächte und Wochenenden führen zu Codeproblemen	180
XP sorgt für ein Mindset, das dem Team und dem Code hilft	181
Iterative Entwicklung hilft Teams, mit Änderungen zurechtzukommen	182
Mut und Respekt halten die Angst vom Projekt fern	184
Teams bauen besseren Code, wenn sie zusammenarbeiten	190
Teams arbeiten am besten, wenn sie zusammensitzen	192
XP-Teams ist Kommunikation wichtig	194
Teams arbeiten am besten mit entspanntem, erholtem Kopf	196
Frage-Sprechstunde: Die »Which is NOT«-Frage	200
XP-Teams heißen Änderungen willkommen	204
Häufiges Feedback sorgt für nur kleine Änderungen	205
Schlechte Erfahrungen führen zu einer rationalen Angst vor Änderungen	206
XP-Praktiken geben Ihnen Feedback zum Code	208
XP-Teams nutzen schnell durchlaufende automatisierte Builds	210
Continuous Integration verhindert unerfreuliche Überraschungen	211
Der wöchentliche Zyklus beginnt mit dem Schreiben von Tests	212
Agile Teams erhalten Feedback aus Design und Tests	214
Pair Programming	216
Komplexer Code lässt sich nur schwer warten	223
Ist Teams Einfachheit wichtig, bauen sie besseren Code	224
Einfachheit ist ein grundlegendes agiles Prinzip	225
Jedes Team baut technische Schulden auf	226
XP-Teams zahlen technische Schulden in jedem wöchentlichen Zyklus zurück	227
Inkrementelles Design beginnt (und endet) mit einfachem Code	228
Prüfungsfragen	234

Lean/Kanban

6 Waste ausmerzen und den Flow managen

Agile Teams wissen, dass sie ihre Arbeitsweise immer verbessern können. Teammitglieder mit einem **Lean Mindset** sind gut darin, herauszufinden, wo sie Zeit aufwenden, ohne dabei **Wert zu liefern**. Dann werden sie den **Waste** los, der sie aufhält. Viele Teams mit einer Lean-Arbeitsweise nutzen **Kanban**, um **Grenzen für ihre Work in Progress** zu setzen, und sie nutzen **Pull-Systeme**, um sicherzustellen, dass die Leute nicht durch Arbeit abgelenkt werden, die nicht viel zu den Zielen beiträgt. Machen Sie sich bereit und lernen Sie, wie das Betrachten Ihres Softwareentwicklungsprozesses als **ganzes System** Ihnen dabei hilft, bessere Software zu bauen!



Mike fühlt sich von all der Arbeit erdrückt, die sein Team noch zu tun hat.

Analytics-Algorithmus anpassen

Stat Mapper Report ändern

Concurrency-Bugs beheben

Änderungen an DB-Code

Verbessertes UI für Stat Mapper

Verbesserungen am Stats Service

UI-Updates

Audience-Profiler anpassen

Änderungen am Dateiformat



Probleme mit dem Audience Analyzer 2.5	246
Lean ist ein Mindset (und keine Vorgehensweise)	248
Die Prinzipien von Lean helfen dabei, die Dinge anders zu sehen	249
Mehr Prinzipien von Lean	250
Manche Denkwerkzeuge kennen Sie noch nicht	254
Weitere Lean-Denkwerkzeuge	256
Bürogespräche	258
Das Unterteilen von Waste kann dabei helfen, ihn besser zu sehen	260
Wertstromanalysen helfen, Waste zu erkennen	264
Zu viele Dinge auf einmal erledigen wollen	267
Anatomie einer Option	270
Systems Thinking hilft Lean-Teams dabei, das Ganze zu sehen	272
Manche »Verbesserungen« funktionieren nicht	273
Lean-Teams nutzen Pull-Systeme, um sicherzustellen, dass sie immer an den wertvollsten Aufgaben arbeiten	274
Frage-Sprechstunde: Die am wenigsten schlechte Option	278
Kanban nutzt ein Pull-System, um Ihren Prozess zu verbessern	280
Visualisieren Sie den Workflow durch Kanban-Boards	281
Wie Sie mit Kanban Ihren Prozess verbessern	282
Warum die WIP begrenzen?	283
Das Team erstellt einen Workflow	284
Bürogespräche	287
Das Team liefert schneller	291
Cumulative Flow Diagrams helfen Ihnen, Ihren Flow zu managen	292
Kanban-Teams reden über ihre Richtlinien	293
Feedback-Schleifen zeigen Ihnen, ob es läuft	294
Jetzt arbeitet das ganze Team zusammen, um eine bessere Arbeitsweise zu entwickeln!	295
Prüfungsfragen	300

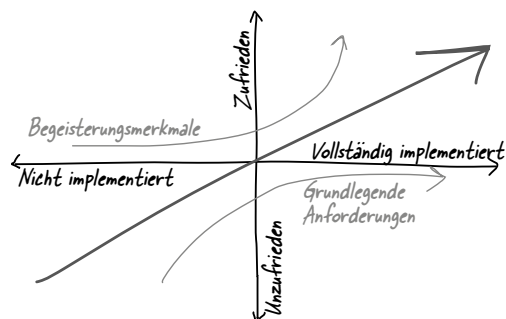
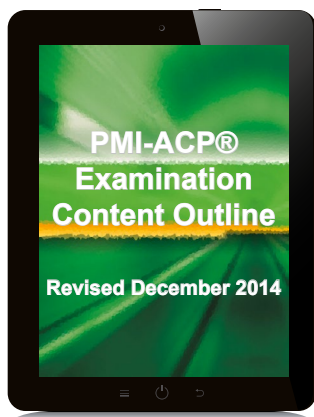
Vorbereitung auf das PMI-ACP® Exam

7

Testen Sie Ihr Wissen

Wow, in den letzten sechs Kapiteln haben Sie ganz schön viel gelernt!

Sie sind in die Werte und Prinzipien des Agilen Manifests eingestiegen, haben gelernt, wie diese eine agile Denkweise steuern, haben untersucht, wie Teams Scrum nutzen, um Projekte zu managen, entdeckt, wie Sie mit XP auf die nächste Entwicklungsstufe gelangen, und haben gesehen, wie Teams mithilfe von Lean/Kanban besser werden. Jetzt ist es an der Zeit, den Blick einmal zurückzurichten und einige der kennengelernten Konzepte genauer anzugehen. Beim **PMI-ACP® Exam** geht es aber um mehr als darum, agile Werkzeuge, Techniken und Konzepte zu verstehen. Um im Test wirklich gut zu sein, müssen Sie herausfinden, wie Teams all das in realen Situationen umsetzen. Lassen Sie Ihr Gehirn daher einen *frischen Blick auf agile Konzepte* werfen – als Unterstützung bekommen Sie einen **vollständigen Satz an Übungen, Rätseln und Fragen** (sowie ein paar neue Dinge), die dafür gedacht sind, Sie auf das englischsprachige PMI-ACP® Exam vorzubereiten.



Die PMI-ACP®-Zertifizierung ist wertvoll ...	308
Das PMI-ACP® Exam basiert auf dem Content Outline	309
»You are an Agile Practitioner...«	310
Eine langfristige Beziehung für Ihr Gehirn	313
Domain 1: Agile Principles and Mindset	314
Domain 1: Agile Principles and Mindset – Lösung	316
Domain 2: Value-Driven Delivery	322
Agile Teams nutzen den Wert für den Kunden, um Anforderungen zu priorisieren	325
Das Berechnen des Werts hilft Ihnen, herauszufinden, welche Projekte umgesetzt werden sollten	326
Domain 2: Value-Driven Delivery – Lösung	330
Domain 3: Stakeholder Engagement	336
Domain 4: Team Performance	337
Domain 3: Stakeholder Engagement – Lösung	338
Domain 4: Team Performance – Lösung	339
Domain 5: Adaptive Planning	348
Passen Sie Ihren Führungsstil laufend an Ihr Team an	349
Ein paar letzte Werkzeuge und Techniken	351
Domain 6: Problem Detection and Resolution	360
Domain 7: Continuous Improvement	361
Domain 5: Adaptive Planning – Lösung	372
Domain 6: Problem Detection and Resolution – Lösung	373
Domain 7: Continuous Improvement – Lösung	374
Sind Sie bereit für das Abschlussexamen?	376

Berufliche Verantwortung

8

Gute Entscheidungen

Es reicht nicht aus, einfach nur technisch versiert zu sein. Sie müssen auch gute Entscheidungen treffen, um in Ihrem Job gut zu sein. Jeder mit dem PMI-ACP Credential verpflichtet sich ebenfalls dazu, dem **Project Management Institute Code of Ethics and Professional Conduct** zu folgen. Dieser Code hilft Ihnen bei **ethischen Entscheidungen**, die im Wissensfundus nicht behandelt werden. Sie werden dazu eventuell ein paar Fragen im PMI-ACP Exam erhalten. Das meiste von dem, was Sie wissen müssen, ist **ziemlich klar**, und mit ein bisschen Nachdenken werden Sie sich gut schlagen.

Das Richtige tun	378
Das Geld behalten?	380
Business Class fliegen?	381
Neue Software	382
Abkürzungen	383
Ein guter Preis oder ein sauberer Fluss?	384
Wir sind nicht alle Engel	385
Prüfungsfragen	386

Wahnsinn. Ich wollte schon
lange wieder shoppen gehen.
Und was ist mit Urlaub?
Acapulco, ich komme!

So ein Geschenk würde ich
niemals annehmen. Meine Arbeit
gut zu erledigen, ist doch
Belohnung genug.

Sorry, aber das kann
ich nicht annehmen.
Trotzdem danke, dass
du an mich denkst.



Der Selbsttest als Download

Ein PMI-ACP-Examen zum Üben

Ich wette, Sie hätten nie gedacht, dass Sie so weit kommen! Es war eine lange Reise, aber falls Sie die PMI-ACP®-Prüfung machen möchten, können Sie jetzt Ihr Wissen überprüfen und sich auf den Prüfungstag vorbereiten. Sie haben sich eine Menge neuer Informationen über Agilität in Ihr Gehirn hochgeladen, und jetzt ist es an der Zeit zu sehen, wie viel davon Sie abrufen können. Deshalb haben wir den ausführlichen, 120 Fragen umfassenden PMI-ACP®-Selbsttest zusammengestellt. Wir haben uns exakt an die PMI-ACP®-Prüfungsinhaltsübersicht gehalten, die die Experten von PMI verwenden, so dass sie genauso aussieht wie die, die Sie am Prüfungstag sehen werden. Jetzt ist es an der Zeit, die mentalen Muskeln zu trainieren. Also tief durchatmen, fertig machen und lasst uns anfangen.

Achtung: Dieses Kapitel finden Sie als **Bonuskapitel zum Download** auf oreilly.de auf der **Webseite zu diesem Buch** unter **Zusatzmaterial**.

Wie man dieses Buch benutzt

Einführung



In diesem Abschnitt beantworten wir die brennende Frage:
»Und warum STEHT so was in einem Buch über Agile?«

Für wen ist dieses Buch?

Wenn Sie alle folgenden Fragen mit »Ja« beantworten können ...

- ① Sind Sie **Entwickler, Projektmanager, Business-Analyst, Designer** oder ein anderes Mitglied eines Teams und möchten Ihre Projekte besser gestalten?
- ② Will **Ihr Team agil werden**, aber Sie sind sich nicht sicher, was das bedeutet oder wie gut Sie dazu passen?
- ③ Halten Sie eventuell Ausschau nach einem neuen Job und wollen verstehen, **warum Firmen nach Agile-Erfahrung fragen**?
- ④ Ziehen Sie **anregende Partyunterhaltungen** trockenen, öden, akademischen Vorlesungen vor?

... dann ist dieses Buch etwas für Sie.

Wer sollte eher die Finger von diesem Buch lassen?

Wenn Sie eine dieser Fragen mit »Ja« beantworten müssen ...

- ① Ist die Arbeit in einem Team oder überhaupt die Zusammenarbeit mit anderen Menschen **für Sie etwas komplett Neues**?
- ② Machen Sie lieber alles allein und sind der festen Überzeugung, mit anderen in einem Team zusammenzuarbeiten, sei **totale Zeitverschwendung**?
- ③ Sie haben **Angst, etwas Neues auszuprobieren**? Sie unterziehen sich lieber einer Wurzelbehandlung, als in einer Streifen-Karo-Kombination auf die Straße zu gehen? Sie sind überzeugt davon, dass ein Fachbuch nicht seriös sein kann, wenn agile Konzepte, Tools und Ideen vermenschlicht werden?

... dann ist dieses Buch nichts für Sie.



[Anmerkung aus dem Marketing: Dieses Buch ist etwas für jeden, der überhaupt noch Puls hat.]

Das PMI-ACP® (Agile Certified Practitioner) ist eines der Zertifikate, das weltweit von immer mehr Firmen gefordert wird.

Bereiten Sie sich auf das PMI-ACP®-Examen vor?

*Dann ist dieses Buch **definitiv** für Sie gemacht! Wir haben es geschrieben, um die Ideen, Konzepte und Praktiken von Agile in Ihren Kopf zu bekommen – und wir haben sichergestellt, dass wirklich alle Themen aus dem Examen behandelt werden. Zudem finden Sie hier unglaublich viel Material, um sich vorzubereiten, unter anderem ein vollständiges (simuliertes) Examen, das so nahe am Original wie möglich ist!*

Wenn Sie noch nie in einem Team gearbeitet haben, werden sich viele der Ideen von Agile fremd anfühlen. Um eins klarzustellen: Wir reden nicht unbedingt von einem Softwareteam – es reicht Erfahrung mit einem beliebigen Team!

Wir wissen, was Sie gerade denken.

»Wie kann *das* ein ernsthaftes Agile-Buch sein?«

»Was sollen all die Abbildungen?«

»Kann ich auf diese Weise wirklich *lernen*?«

Und wir wissen, was Ihr Gehirn gerade denkt.

Ihr Gehirn lechzt nach Neuem. Es ist ständig dabei, Ihre Umgebung abzusuchen, und es *wartet* auf etwas Ungewöhnliches. So ist es nun einmal gebaut, und es hilft Ihnen zu überleben.

Also, was macht Ihr Gehirn mit all den gewöhnlichen, normalen Routinesachen, denen Sie begegnen? Es tut alles in seiner Macht Stehende, damit es dadurch nicht bei seiner *eigentlichen* Arbeit gestört wird: Dinge zu erfassen, die *wirklich* wichtig sind. Es gibt sich nicht damit ab, die langweiligen Sachen zu speichern, sondern lässt sie gar nicht erst durch den »Dies-ist-offensichtlich-nicht-wichtig«-Filter.

Woher *weiß* Ihr Gehirn denn, was wichtig ist? Nehmen Sie an, Sie machen einen Tagesausflug und ein Tiger springt vor Ihnen aus dem Gebüsch – was passiert dabei in Ihrem Kopf und Ihrem Körper?

Neuronen feuern. Gefühle werden angekurbelt. *Chemische Substanzen durchfluten Sie.*

Und so weiß Ihr Gehirn:

Dies muss wichtig sein! Vergiss es nicht!

Aber nun stellen Sie sich vor, Sie sind zu Hause oder in einer Bibliothek. In einer sicheren, warmen, tigerfreien Zone. Sie lernen. Bereiten sich auf eine Prüfung vor. Oder Sie versuchen, irgendein schwieriges Thema zu lernen, von dem Ihr Chef glaubt, Sie bräuchten dafür eine Woche oder höchstens zehn Tage.

Da ist nur ein Problem: Ihr Gehirn möchte Ihnen einen großen Gefallen tun. Es versucht, dafür zu sorgen, dass diese *offensichtlich* unwichtigen Inhalte nicht knappe Ressourcen verstopfen. Ressourcen, die besser dafür verwendet würden, die wirklich *wichtigen* Dinge zu speichern. Wie Tiger. Wie die Gefahren des Feuers. Oder dass Sie nie wieder in kurzen Hosen Snowboard fahren sollten.

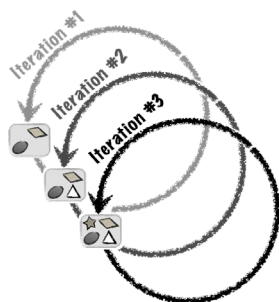
Und es gibt keine einfache Möglichkeit, Ihrem Gehirn zu sagen: »Hey, Gehirn, vielen Dank, aber egal, wie langweilig dieses Buch auch ist und wie klein der Ausschlag auf meiner emotionalen Richterskala gerade ist, ich *will* wirklich, dass du diesen Kram behältst.«



Wir stellen uns unseren Leser als einen aktiv Lernenden vor.

Also, was ist nötig, damit Sie etwas *lernen*? Erst einmal müssen Sie es *aufnehmen* und dann dafür sorgen, dass Sie es nicht wieder *vergessen*. Es geht nicht darum, Fakten in Ihren Kopf zu schieben. Nach den neuesten Forschungsergebnissen der Kognitionswissenschaft, der Neurobiologie und der Lernpsychologie gehört zum *Lernen* viel mehr als nur Text auf einer Seite. Wir wissen, was Ihr Gehirn anmacht.

Einige der Lernprinzipien dieser Buchreihe:



Bilder einsetzen. An Bilder kann man sich viel besser erinnern als an Worte allein und lernt so viel effektiver (bis zu 89 % Verbesserung bei Ab-rufbarkeits- und Lerntransferstudien). Außerdem werden die Dinge dadurch verständlicher.

Text in oder neben die Grafiken setzen, auf die sie sich beziehen, anstatt darunter oder auf eine andere Seite. Die Leser werden auf den Bild-inhalt bezogene Probleme dann mit *doppelt* so hoher Wahrscheinlichkeit lösen können.

Verwenden Sie einen gesprächs-orientierten Stil mit persönlicher Ansprache. Nach neueren Untersuchungen haben Studenten nach dem Lernen bei Tests bis zu 40 % besser abgeschnitten, wenn der Inhalt den Leser direkt in der ersten Person und im lockeren Stil angesprochen hat statt in einem formalen Ton. Halten Sie keinen Vortrag, sondern erzählen Sie Geschichten. Benutzen Sie eine zwanglose Sprache. Nehmen Sie sich selbst nicht zu ernst. Würden Sie einer anregenden Unterhaltung beim Abendessen mehr Aufmerksamkeit schenken oder einem Vortrag?

Ich habe Amy eine einfache 10-Sekunden-Frage gestellt, und jetzt warte ich hier seit zwei Stunden auf Antwort.



Bringen Sie den Lernenden dazu, intensiver nachzudenken. Mit anderen Worten: Solange Sie nicht aktiv Ihre Neuronen strapazieren, passiert in Ihrem Gehirn nicht viel. Ein Leser muss motiviert, begeistert und neugierig sein und dazu angeregt werden, Probleme zu lösen, Schlüsse zu ziehen und sich neues Wissen anzueignen. Und dafür brauchen Sie Herausforderungen, Übungen, zum Nachdenken anregende Fragen und Tätigkeiten, die beide Seiten des Gehirns und mehrere Sinne einbeziehen.



Ziehen Sie die Aufmerksamkeit des Lesers auf sich – und behalten Sie sie. Wir alle haben schon Erfahrungen dieser Art gemacht: »Ich will das wirklich lernen, aber ich kann einfach nicht über Seite 1 hinaus wach bleiben.« Ihr Gehirn passt auf, wenn Dinge ungewöhnlich, interessant, merkwürdig, auffällig, unerwartet sind. Ein neues, schwieriges, technisches Thema zu lernen, muss nicht langweilig sein. Wenn es das nicht ist, lernt Ihr Gehirn viel schneller.

Sprechen Sie Gefühle an. Wir wissen, dass Ihre Fähigkeit, sich an etwas zu erinnern, wesentlich von dessen emotionalem Gehalt abhängt. Sie erinnern sich an das, was Sie bewegt. Sie erinnern sich, wenn Sie etwas *fühlen*. Nein, wir erzählen keine herzzerreißenden Geschichten über einen Jungen und seinen Hund. Was wir erzählen, ruft Überraschungs-, Neugier-, Spaß- und Was-zum-...?-Emotionen hervor und dieses Hochgefühl, das Sie beim Lösen eines Puzzles empfinden oder wenn Sie etwas lernen, das alle anderen schwierig finden. Oder wenn Sie merken, dass Sie etwas können, das dieser »Ich-bin-ein-besserer-Techniker-als-du«-Typ aus der Technikabteilung *nicht kann*.



Metakognition: Nachdenken übers Denken

Wenn Sie wirklich lernen möchten, und zwar schneller und nachhaltiger, dann schenken Sie Ihrer Aufmerksamkeit Aufmerksamkeit. Denken Sie darüber nach, wie Sie denken. Lernen Sie, wie Sie lernen.

Die meisten von uns haben in ihrer Jugend keine Kurse in Metakognition oder Lerntheorie gehabt. Es wurde von uns *erwartet*, dass wir lernen, aber nur selten wurde uns auch *beigebracht*, wie man lernt.

Wir nehmen aber an, dass Sie wirklich etwas über Agile lernen möchten, wenn Sie dieses Buch in den Händen halten. Und wahrscheinlich möchten Sie nicht viel Zeit aufwenden. Weil Sie das aber bei einem echten Projekt anwenden müssen (und besonders dann, wenn Sie eine Prüfung dazu ablegen wollen), müssen Sie sich an das *erinnern*, was Sie lesen. Und deshalb müssen Sie es *verstehen*. Wenn Sie so viel wie möglich von diesem Buch profitieren wollen oder von *irgendeinem* anderen Buch oder einer anderen Lernerfahrung, übernehmen Sie Verantwortung für Ihr Gehirn. Ihr Gehirn im Zusammenhang mit *diesem* Lernstoff.

Der Trick besteht darin, Ihr Gehirn dazu zu bringen, neuen Lernstoff als etwas wirklich Wichtiges anzusehen. Als entscheidend für Ihr Wohlbefinden. So wichtig wie einen Tiger. Andernfalls stecken Sie in einem dauernden Kampf, in dem Ihr Gehirn sein Bestes gibt, um die neuen Inhalte davon abzuhalten, hängen zu bleiben.

Wie bringen Sie also Ihr Gehirn dazu, Agile für so wichtig zu halten wie einen hungrigen Tiger?

Da gibt es den langsamen, ermüdenden Weg oder den schnelleren, effektiveren Weg. Der langsame Weg geht über bloße Wiederholung. Natürlich ist Ihnen klar, dass Sie lernen und sich sogar an die langweiligsten Themen erinnern *können*, wenn Sie sich die gleiche Sache immer wieder einhämmern. Wenn Sie nur oft genug wiederholen, sagt Ihr Gehirn: »Er hat zwar nicht das *Gefühl*, dass das wichtig ist, aber er sieht sich dieselbe Sache *immer und immer wieder* an – dann muss sie wohl wichtig sein.«

Der schnellere Weg besteht darin, **alles zu tun, was die Gehirnaktivität erhöht**, vor allem verschiedene Arten von Gehirnaktivität. Eine wichtige Rolle dabei spielen die auf der vorhergehenden Seite erwähnten Dinge – alles Dinge, die nachweislich helfen, dass Ihr Gehirn *für* Sie arbeitet. So hat sich z. B. in Untersuchungen gezeigt: Wenn Wörter *in* den Abbildungen stehen, die sie beschreiben (und nicht irgendwo anders auf der Seite, z. B. in einer Bildunterschrift oder im Text), versucht Ihr Gehirn, herauszufinden, wie die Wörter und das Bild zusammenhängen, und dadurch feuern mehr Neuronen. Und je mehr Neuronen feuern, umso größer ist die Chance, dass Ihr Gehirn mitbekommt: Bei dieser Sache lohnt es sich, aufzupassen, und vielleicht auch, sich daran zu erinnern.

Ein lockerer Sprachstil hilft, denn Menschen tendieren zu höherer Aufmerksamkeit, wenn ihnen bewusst ist, dass sie ein Gespräch führen – man erwartet dann ja von ihnen, dass sie dem Gespräch folgen und sich beteiligen. Das Erstaunliche daran ist: Es ist Ihrem Gehirn ziemlich *egal*, dass die »Unterhaltung« zwischen Ihnen und einem Buch stattfindet! Wenn der Schreibstil dagegen formal und trocken ist, hat Ihr Gehirn den gleichen Eindruck wie bei einem Vortrag, bei dem in einem Raum passive Zuhörer sitzen. Nicht nötig, wach zu bleiben.

Aber Abbildungen und ein lockerer Sprachstil sind erst der Anfang.



Das haben WIR getan:

Wir haben **Bilder** verwendet, weil Ihr Gehirn auf visuelle Eindrücke eingestellt ist, nicht auf Text. Soweit es Ihr Gehirn betrifft, sagt ein Bild *wirklich* mehr als tausend Worte. Und dort, wo Text und Abbildungen zusammenwirken, haben wir den Text *in* die Bilder eingebettet, denn Ihr Gehirn arbeitet besser, wenn der Text *innerhalb* der Sache steht, auf die er sich bezieht, und nicht in einer Bildunterschrift oder irgendwo vergraben im Text.

Wir haben **Redundanz** eingesetzt, das heißt dasselbe auf *unterschiedliche* Art und mit verschiedenen Medientypen ausgedrückt, damit Sie es über *mehrere Sinne* aufnehmen können. Das erhöht die Chance, dass die Inhalte an mehr als nur einer Stelle in Ihrem Gehirn verankert werden.

Wir haben Konzepte und Bilder in **unerwarteter** Weise eingesetzt, weil Ihr Gehirn auf Neuigkeiten programmiert ist. Und wir haben Bilder und Ideen mit zumindest *etwas emotionalem Charakter* verwendet, weil Ihr Gehirn darauf eingestellt ist, auf die Biochemie von Gefühlen zu achten. An alles, was ein *Gefühl* in Ihnen auslöst, können Sie sich mit höherer Wahrscheinlichkeit erinnern, selbst wenn dieses Gefühl nicht mehr ist als ein bisschen **Belustigung**, **Überraschung** oder **Interesse**.

Wir haben einen **umgangssprachlichen Stil** mit direkter Anrede benutzt, denn Ihr Gehirn ist von Natur aus aufmerksamer, wenn es Sie in einer Unterhaltung wähnt, als wenn es davon ausgeht, dass Sie passiv einer Präsentation zuhören – sogar dann, wenn Sie *lesen*.

Wir haben mehr als 80 **Aktivitäten** für Sie vorgesehen, denn Ihr Gehirn lernt und behält von Natur aus besser, wenn Sie Dinge **tun**, als wenn Sie nur darüber *lesen*. Und wir haben die Übungen zwar anspruchsvoll, aber doch lösbar gemacht, denn so ist es den meisten Lesern am liebsten.

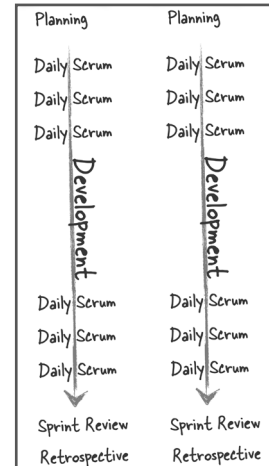
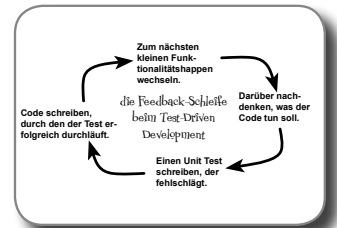
Wir haben **mehrere unterschiedliche Lernstile** eingesetzt, denn vielleicht bevorzugen Sie ein Schritt-für-Schritt-Vorgehen, während jemand anders erst einmal den groben Zusammenhang verstehen und ein Dritter einfach nur ein Beispiel sehen möchte. Aber ganz abgesehen von den jeweiligen Lernvorlieben profitiert *jeder* davon, wenn er die gleichen Inhalte in unterschiedlicher Form präsentiert bekommt.

Wir liefern Inhalte für **beide Seiten Ihres Gehirns**, denn je mehr Sie von Ihrem Gehirn einsetzen, umso wahrscheinlicher werden Sie lernen und behalten, und umso länger bleiben Sie konzentriert. Wenn Sie mit einer Seite des Gehirns arbeiten, bedeutet das häufig, dass sich die andere Seite des Gehirns ausruhen kann; so können Sie über einen längeren Zeitraum produktiver lernen.

Und wir haben **Geschichten** und Übungen aufgenommen, die **mehr als einen Blickwinkel repräsentieren**, denn Ihr Gehirn lernt von Natur aus intensiver, wenn es gezwungen ist, selbst zu analysieren und zu beurteilen.

Wir haben **Herausforderungen** eingefügt: in Form von Übungen und indem wir **Fragen** stellen, auf die es nicht immer eine eindeutige Antwort gibt, denn Ihr Gehirn ist darauf eingestellt, zu lernen und sich zu erinnern, wenn es an etwas *arbeiten* muss. Überlegen Sie: Ihren *Körper* bekommen Sie ja auch nicht in Form, wenn Sie nur die Leute auf dem Sportplatz *beobachten*. Aber wir haben unser Bestes getan, um dafür zu sorgen, dass Sie – wenn Sie schon hart arbeiten – an den *richtigen* Dingen arbeiten. Dass Sie **nicht einen einzigen Dendriten darauf verschwenden**, ein schwer verständliches Beispiel zu verarbeiten oder einen schwierigen, mit Fachbegriffen gespickten oder übermäßig gedrängten Text zu analysieren.

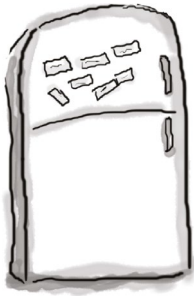
Wir haben **Menschen** eingesetzt. In Geschichten, Beispielen, Bildern usw. – denn *Sie sind* ein Mensch. Und Ihr Gehirn schenkt *Menschen* mehr Aufmerksamkeit als *Dingen*.



Punkt für Punkt

Kamingespräche





Und das können SIE tun, um sich Ihr Gehirn untertan zu machen

So, wir haben unseren Teil der Arbeit geleistet. Der Rest liegt bei Ihnen. Diese Tipps sind ein Anfang; hören Sie auf Ihr Gehirn und finden Sie heraus, was bei Ihnen funktioniert und was nicht. Probieren Sie neue Wege aus.

Schneiden Sie dies aus und heften Sie es an Ihren Kühlschrank.

① Immer langsam. Je mehr Sie verstehen, umso weniger müssen Sie auswendig lernen.

Lesen Sie nicht nur. Halten Sie inne und denken Sie nach. Wenn das Buch Sie etwas fragt, springen Sie nicht einfach zur Antwort. Stellen Sie sich vor, dass Sie das wirklich jemand *fragt*. Je gründlicher Sie Ihr Gehirn zum Nachdenken zwingen, umso größer ist die Chance, dass Sie lernen und behalten.

② Bearbeiten Sie die Übungen. Machen Sie selbst Notizen.

Wir haben sie entworfen, aber wenn wir sie auch für Sie lösen würden, wäre das, als würde jemand anderes Ihr Training für Sie absolvieren. Und *sehen* Sie sich die Übungen *nicht einfach nur an*. **Benutzen Sie einen Bleistift.** Es deutet vieles darauf hin, dass körperliche Aktivität beim Lernen den Lernerfolg erhöhen kann.

③ Lesen Sie die Abschnitte »Es gibt keine Dummen Fragen«.

Und zwar alle. Das sind keine Zusatzanmerkungen – **sie gehören zum Kerninhalt!** Überspringen Sie sie nicht.

④ Lesen Sie dies als Letztes vor dem Schlafengehen. Oder lesen Sie danach zumindest nichts Anspruchsvolles mehr.

Ein Teil des Lernprozesses (vor allem die Übertragung in das Langzeitgedächtnis) findet erst statt, *nachdem* Sie das Buch zur Seite gelegt haben. Ihr Gehirn braucht Zeit für sich, um weitere Verarbeitung zu leisten. Wenn Sie in dieser Zeit etwas Neues aufnehmen, geht ein Teil dessen, was Sie gerade gelernt haben, verloren.

⑤ Trinken Sie Wasser. Viel.

Ihr Gehirn arbeitet am besten in einem schönen Flüssigkeitsbad. Austrocknung (zu der es schon kommen kann, bevor Sie überhaupt Durst verspüren) beeinträchtigt die kognitive Funktion.

⑥ Reden Sie darüber. Laut.

Sprechen aktiviert einen anderen Teil des Gehirns. Wenn Sie etwas verstehen oder Ihre Chancen verbessern wollen, sich später daran zu erinnern, sagen Sie es laut. Noch besser: Versuchen Sie, es jemandem laut zu erklären. Sie lernen dann schneller und haben vielleicht Ideen, auf die Sie beim bloßen Lesen nie gekommen wären.

⑦ Hören Sie auf Ihr Gehirn.

Achten Sie darauf, Ihr Gehirn nicht zu überladen. Wenn Sie merken, dass Sie etwas nur noch überfliegen oder dass Sie das gerade erst Gelesene vergessen haben, ist es Zeit für eine Pause. Ab einem bestimmten Punkt lernen Sie nicht mehr schneller, indem Sie mehr hineinzustopfen versuchen; das kann sogar den Lernprozess stören.

⑧ Aber bitte mit Gefühl!

Ihr Gehirn muss wissen, dass es *um etwas Wichtiges geht*. Lassen Sie sich in die Geschichten hineinziehen. Erfinden Sie eigene Bildunterschriften für die Fotos. Über einen schlechten Scherz zu stöhnen, ist *immer noch* besser, als gar nichts zu fühlen.

⑨ Erschaffen Sie etwas!

Wenden Sie dies auf Ihre tägliche Arbeit an; setzen Sie das, was Sie gerade lernen, ein, um Entscheidungen in Ihren Projekten zu fällen. Tun Sie irgendetwas, um Erfahrungen zu sammeln, die über die Übungen und Aktivitäten in diesem Buch hinausgehen. Sie brauchen nur einen Bleistift und ein Problem, das es zu lösen gilt ... ein Problem, das vielleicht vom Einsatz der Tools und Techniken profitiert, von denen Sie in diesem Buch erfahren.

Lies mich

Dies ist ein Lehrbuch, keine Referenz. Wir haben mit Absicht alles weggelassen, was Ihnen dabei in die Quere kommen könnte, das zu lernen, was auch immer wir an einem bestimmten Punkt gerade behandeln. Und wenn Sie das Buch das erste Mal durchhaben, werden Sie es bestimmt auf dem Schreibtisch liegen lassen, um nützliche Ideen, Tools und Techniken nochmals nachlesen zu können. Aber Sie müssen auch nach dem ersten Lesen wieder am Anfang beginnen, denn das Buch baut stets auf dem auf, was Sie schon gesehen und gelernt haben.

Die Aktivitäten sind NICHT optional.

Die Übungen und Aktivitäten sind keine Zusätze. Sie sind ein wesentlicher Bestandteil des Buchs. Einige sollen Ihr Gedächtnis unterstützen, andere Ihr Verständnis wecken, und wieder andere werden Ihnen helfen, das Gelernte anzuwenden. **Überspringen Sie die Übungen nicht.** Selbst die Kreuzworträtsel sind wichtig – Sie helfen dabei, Konzepte in Ihr Gehirn zu bekommen. Vor allem aber sind sie eine gute Möglichkeit, Ihrem Hirn eine Chance zu geben, über die Wörter und Begriffe, die Sie gelernt haben, in einem anderen Kontext nachzudenken.

Die Redundanz ist beabsichtigt und wichtig.

Eins der Dinge, die ein Von Kopf bis Fuß-Buch so anders machen, ist, dass wir möchten, dass Sie die Sache *wirklich* verstehen. Und wir möchten, dass Sie, wenn Sie das Buch durchgearbeitet haben, behalten, was Sie gelernt haben. Die meisten Referenzbücher zielen nicht auf Behalten und Erinnern ab, aber in diesem Buch geht es ums *Lernen*, deswegen werden Ihnen manche Konzepte auch mehrfach begegnen.

Zu den Kopfnuss-Übungen gibt es keine Lösungen.

Für manche dieser Übungen gibt es keine richtige Lösung, und bei anderen gehört es zum Lernprozess der Kopfnuss-Aktivitäten, dass Sie selbst überlegen, ob und wann Ihre Lösungen richtig sind. Bei einigen Kopfnuss-Übungen finden Sie Hinweise, die Sie in die richtige Richtung lenken.

Versuchen Sie, die Prüfungsfragen zu lösen – auch wenn Sie das Examen gar nicht ablegen wollen!

Manche unserer Leser bereiten sich auf das dreistündige, aus 120 Fragen bestehende PMI-ACP® Certification Exam vor. Zum Glück ist **das Erlernen von Agile** der effektivste Weg, sich dafür zu wappnen. Also auch wenn Sie nicht an der PMI-ACP®-Zertifizierung interessiert sind, ist dieses Buch für Sie von Wert. Dennoch sollten Sie versuchen, die Prüfungsfragen am Ende der Kapitel zu lösen, denn das Beantworten dieser Frage ist ein **wirklich effektiver Weg**, agile Konzepte in Ihr Gehirn zu bekommen.

Das Team für das technische Review

Lisa Kellner



Dave Prior



Philip Cheung



Keith Conant



Kelly D. Marce



Technische Reviewer:

Dave Prior managt seit über 20 Jahren Projekte und hat sich seit 2009 vollständig auf Agile konzentriert. Er ist Certified Scrum Trainer und arbeitet für LeadingAgile. Sein Motivator ist Otis Redding, und wenn er nur ein Nahrungsmittel zu sich nehmen könnte, wäre das Kaffee.

Keith Conant ist seit 20 Jahren als Entwickler, Projektmanager und Group Manager im Softwaregeschäft tätig. Aktuell leitet er ein Team, das eine POS-Payment-Anwendung verbessert, die von Universitäten auf der ganzen Welt genutzt wird. Außerhalb seiner Arbeit kann man Keith beim Komponieren von Musik finden, er spielt Schlagzeug, Gitarre und Keyboards in einer Band oder fordert seinen Körper beim Kajakfahren, Laufen, Wandern und Radfahren heraus.

Philip Cheung entwickelt Software seit 15 Jahren und nutzt seit 2013 ausschließlich Agile, um Projekte zu managen und Ergebnisse zu liefern. Er arbeitet in der Finanzbranche und ist an der Entwicklung diverser Anwendungen für Großunternehmen beteiligt. Philip entspannt sich gern in der Landschaft Englands und hofft, eines Tages in einem schnuckeligen Cottage auf dem Land seinen Lebensabend verbringen zu können.

Kelly D. Marce, PMP®, PMI-ACP®, besitzt über neun Jahre Erfahrung im Projektmanagement. Er ist Agile-Trainer, zertifizierter Projektmanager und PMP®-Mentor bei einem führenden Finanzdienstleister in Kanada. In seiner Freizeit agiert er als Organisator von Liveevents in seiner Community und versucht, mit seinem vier Jahre alten Sohn Jacob Schritt zu halten.

Und wie immer freuen wir uns, dass sich **Lisa Kellner** wieder zu unserem Tech-Review-Team gesellt hat. Lisa war wie jedes Mal wunderbar. Vielen Dank an alle!

Danksagungen

Unsere Lektorin:

Wir möchten unserer Lektorin **Nan Barber** dafür danken, dass sie dieses Buch betreut.



←
Nan Barber

Das Team bei O'Reilly:

Es gibt so viele Menschen bei O'Reilly, denen wir danken wollen – hoffentlich haben wir niemanden vergessen!

Auf jeden Fall bedanken wir uns beim Production-Team. Und dann sind da noch ein paar Leute, denen wir danken möchten:

Wie immer lieben wir **Mary Treseler** und können es gar nicht erwarten, wieder mit ihr zusammenzuarbeiten! Ein weiterer großer Dank geht an unsere anderen Freunde und Lektoren **Mike Hendrickson, Tim O'Reilly, Andy Oram, Laurel Ruma, Lindsay Ventimiglia, Melanie Yarbrough, Ron Bilodeau, Lucie Haskins** und **Jasmine Kwityn**. Und wenn Sie dieses Buch gerade lesen, können Sie dafür dem großartigsten Publicity-Team der Branche Dank aussprechen: **Marsee Henon, Kathryn Barret** und all den anderen in Sebastopol.

Über den Übersetzer dieses Buchs

Thomas Demmig ist Diplom-Physiker und lebt mit seiner Familie in Mannheim. Er arbeitet bei der SAP SE als Entwickler im UI-Umfeld und hat in den letzten Jahren einige Bücher aus den Bereichen Management und Entwicklung übersetzt.

Thomas Demmig



© Jan Abel (janabel.com)

Lob für Agile von Kopf bis Fuß

Wir haben den Experten für Organisationstransformationen und internationalen Teilzeit-Rockstar Mike Monsoon gebeten, eine frühe Version dieses Buchs zu begutachten und uns etwas für die Lob-Seite zu schicken. Stattdessen hat er ein Lied geschrieben!

Praise Quote

I've read so many agile books before
But they all come off the same,
A bunch of pretentious preachy solemn righteous stuff,
I just find them all lame.

Break it down (you) break it down for me.
(If I could T-shirt size it
It'd be an extra-large)

Meta meta meta cognition
I finally found the right information.
I really appreciate the book that you wrote.
But I can't make a praise quote.

– Mike Monsoon, internationaler Rockstar

Hier können Sie das Lied hören: <https://bit.ly/head-first-agile-song>

1 Was ist Agile?

Prinzipien und Praktiken

Wir **glauben** einfach mal, dass im Projekt alles perfekt ablaufen wird, und das halten wir hier in diesem Plan fest.



Brillante Planung, Sir!

Es ist so eine spannende Zeit, agil zu sein! Zum ersten Mal hat unsere Branche einen echten, nachhaltigen Weg gefunden, Probleme zu lösen, mit denen Generationen von Softwareentwicklungsteams gekämpft hatten. Agile Teams verwenden einfache, gradlinige **Praktiken**, die sich schon in realen Projekten bewährt haben. Aber Moment mal ... wenn Agile so toll ist, warum nutzt es dann nicht schon jeder? Nun, in der Welt da draußen kann es passieren, dass eine Praktik, die für ein Team gut funktioniert, bei einem anderen Team zu echten Problemen führt – der Unterschied liegt im **Mindset** des Teams. Machen Sie sich also bereit, Ihre Denkweise über Projekte zu verändern!

Das neue Feature hört sich toll an ...

Das ist Kate. Sie ist Projektmanagerin bei einem erfolgreichen Start-up im Silicon Valley. Ihre Firma baut Software, die von Video- und Musikstreaming-Diensten sowie Internetradio-stationen eingesetzt wird, um das Publikum in Echtzeit analysieren und Programmvorschlägen nachkommen zu können, die die Betrachter oder Zuhörer glücklich machen. Und jetzt hat Kates Team die Gelegenheit, etwas auszuliefern, das der Firma wirklich helfen wird.

Ich werde
mit dem Team besprechen,
dass wir diese neuen, ausgefeilten Publi-
kumsanalysefeatures im nächsten Release
unterbringen müssen.

Kate ist Projekt-
managerin für ein
Softwareteam.



Danke sehr,
Kate. Wenn das Team das schnell
erledigen kann, wird unser größter Kunde 50
zusätzliche Lizenzen kaufen, und wir werden dieses
Jahr einen fetten Bonus erhalten!

Ben ist der Product
Owner. Seine Aufgabe
besteht darin, mit Kunden
zu reden, herauszufinden,
was sie brauchen, und neue
Features vorzuschlagen,
die sie nutzen können.



... aber es läuft nicht immer wie erwartet

Kates Unterhaltung mit dem Projektteam läuft nicht annähernd so gut wie erhofft. Was wird sie Ben erzählen?

Mike ist der leitende
Programmierer und
Architekt.



Es klingt,
als hätten wir wirklich eine
Chance, unsere Kunden glücklich zu
machen.

Kate: Wenn wir also dieses neue, ausgefeilte Feature zur Publikumsanalyse in das nächste Release bekommen, werden wir alle einen dicken Bonus erhalten.

Mike: Nun ja, das würde ziemlich großartig klingen.

Kate: Fantastisch! Wir können also auf euch zählen?

Mike: Moment mal, nicht so schnell. Ich sagte, das *würde* ziemlich großartig *klingen*. Aber ich sehe keine Chance, dass es klappt.

Kate: Was, wieso?! Das kannst du nicht machen, Mike.

Mike: Schau mal, wenn wir davon vor vier Monaten gehört hätten, als wir den Service zur Analyse der Publikumsdaten entwarfen, wäre das kein Problem gewesen. Aber jetzt müssten wir sehr viel Code wegwerfen und ihn ersetzen durch ... Na ja, ich will hier nicht in die technischen Details gehen.

Kate: Gut. Ich will sie auch gar nicht hören.

Mike: Also sind wir dann hier fertig? Denn mein Team hat ganz schön viel zu tun.

Agile eilt zur Hilfe!

Kate hat etwas über Agile gelesen und glaubt, dass es ihr helfen könnte, diese Features in das nächste Release zu bekommen. Bei Softwareteams ist Agile ziemlich beliebt geworden, und diejenigen, die jetzt »agil« sind, reden gerne über die tollen Ergebnisse, die sie damit erzielt haben. Die gebaute Software ist besser, was *sowohl* für die Teams *als auch* für die Anwender einen großen Unterschied bedeutet. Dazu kommt: Wenn agile Teams effektiv sind, ist die Arbeit gleich viel angenehmer! Es ist entspannter, und die Arbeitsumgebung macht mehr Spaß.

Warum ist Agile also so beliebt? Dafür gibt es viele Gründe:

- ★ Wenn Teams agil werden, merken sie, dass es viel einfacher ist, Deadlines einzuhalten.
- ★ Auch werden Bugs in ihrer Software viel seltener.
- ★ Der Code lässt sich leichter warten – Ergänzungen, Erweiterungen oder Änderungen an der Codebasis sorgen nicht länger für Kopfschmerzen.
- ★ Die Anwender sind viel glücklicher, was allen das Leben leichter macht.
- ★ Und das Beste: Wenn agile Teams effektiv sind, lebt es sich für die Teammitglieder besser, denn sie können zu vernünftigen Zeiten nach Hause gehen und müssen am Wochenende nur noch selten ran (für viele Entwickler sicherlich eine Premiere!).

Ein Daily Standup ist ein guter Ausgangspunkt

Eine der verbreitetsten agilen Praktiken, die von Teams übernommen wird, ist das **Daily Standup**, ein tägliches Meeting, in dem die Teammitglieder davon erzählen, woran sie arbeiten und was ihnen im Weg steht. Weil sämtliche Teilnehmer die ganze Zeit stehen, dauern diese Meetings nie lang. Viele Teams wurden erfolgreicher, indem sie ihre Projekte um Daily Standups ergänzten, und es ist häufig der erste Schritt auf dem Weg zu Agile.

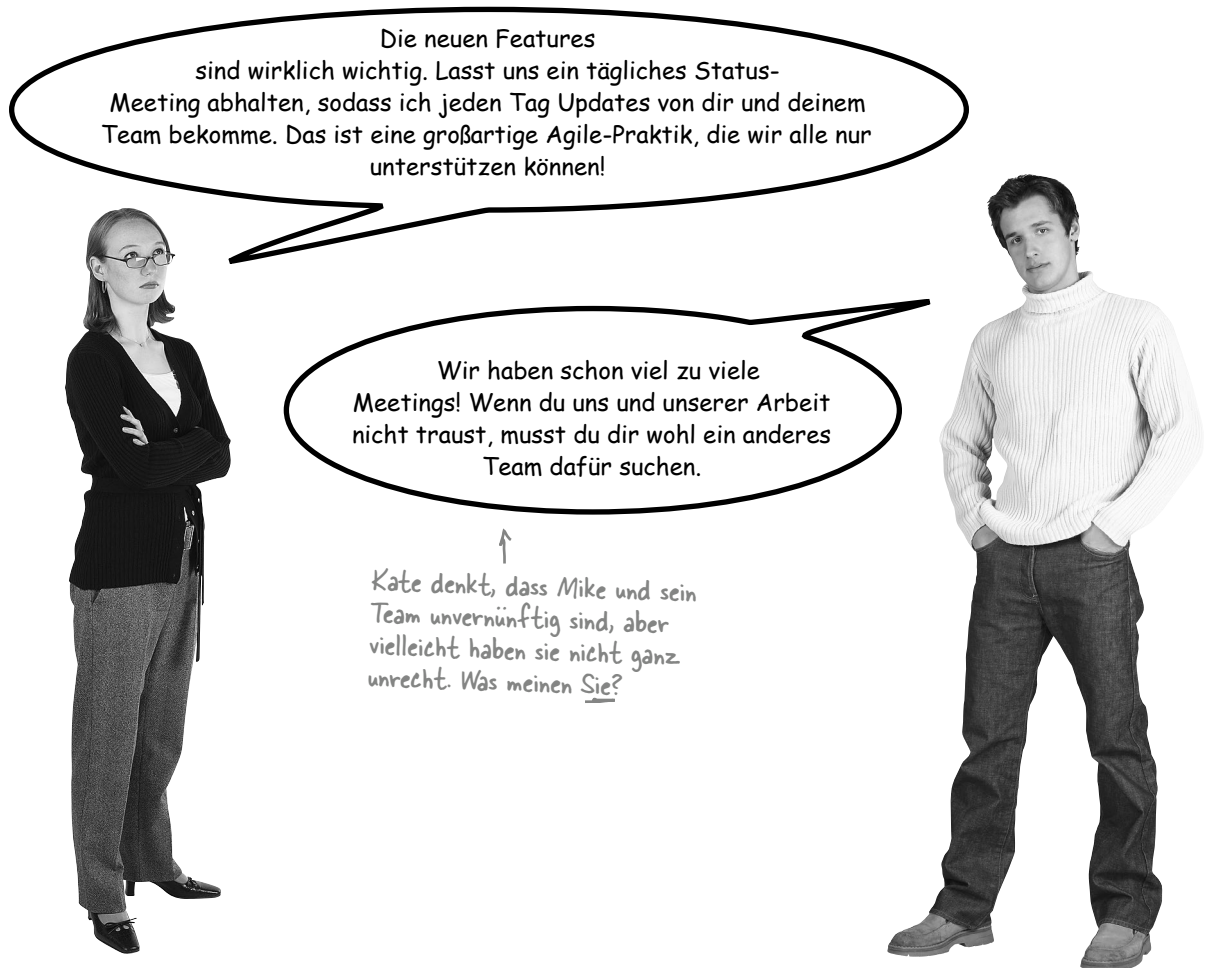
In einem Daily Standup stehen alle im Team während des gesamten Meetings. So bleibt es kurz, nett und fokussiert.



Aber achtet der da wirklich darauf, was seine Kollegen sagen?

Kate versucht, ein Daily Standup abzuhalten

Zu Kates Überraschung lässt sich nicht jeder in Mikes Team von ihrer Begeisterung für diese neue Praktik anstecken. Tatsächlich ist einer seiner Entwickler sogar richtiggehend verärgert darüber, dass sie noch ein weiteres Meeting vorschlägt, und er ist beleidigt, dass er jeden Tag an einem Meeting teilnehmen soll, bei dem er neugierige Fragen zu seiner täglichen Arbeit zu beantworten hat.



KOPF- NUSS

Was passiert hier? Verhält sich Mike unvernünftig? Fordert Kate zu viel? Warum führt diese einfache, weitverbreitete Praxis zu einem Konflikt?

Verschiedene Teammitglieder verhalten sich unterschiedlich

Kate bekam bei der Übernahme von Agile direkt Probleme – und sie ist da nicht die Einzige.

In Wahrheit gibt es viele Teams, die mit Agile nicht so erfolgreich waren, wie sie es gehofft hatten. Wussten Sie, dass mehr als die Hälfte der Firmen, die Software bauen, schon mit Agile experimentiert haben? Trotz der Erfolgsgeschichten – und davon gibt es viele! – probieren diverse Teams Agile aus, gelangen aber zu Ergebnissen, mit denen sie nicht wirklich glücklich sind. Tatsächlich fühlen sie sich sogar ein bisschen übers Ohr gehauen! Es scheint, dass Agile verspricht, viele große Veränderungen zu bewirken, aber dass ein Ausprobieren in eigenen Projekten nie so richtig funktioniert.

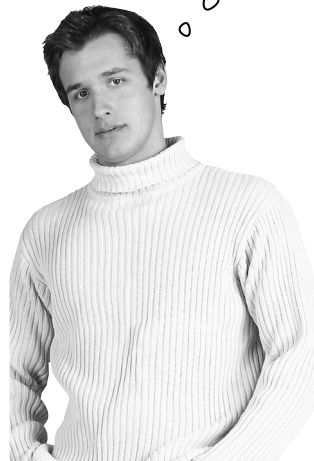
Und das ist auch Kate passiert. Sie hat ganz allein einen Plan erstellt und will nun Status-Updates von ihrem Team bekommen. Also hat sie damit begonnen, ein widerwilliges Team zu einem Daily Standup zu drängen. Sie schafft es, die Mitarbeiter im Raum zu versammeln. Aber wird das wirklich einen Unterschied machen? Sie sorgt sich, dass die Leute zu sehr von ihrem Plan abweichen, also konzentriert sie sich darauf, von jeder einzelnen Person ein Status-Update zu erhalten. Mike und seine Entwickler wollen hingegen, dass das Meeting so schnell wie möglich endet, damit sie zurück an die »richtige« Arbeit gehen können.

In Kates nicht so effektivem Daily Standup hört niemand den anderen zu, und wenn der Moment für das eigene Update gekommen ist, sagt jeder so wenig wie möglich. Sie bekommt zwar trotzdem nützliche Informationen, aber auf Kosten von Langeweile und Widerstand – und keiner im Meeting kooperiert.

Ich erfahre nur dann etwas über Probleme, wenn es schon **zu spät** ist, um noch etwas daran ändern zu können.

Wenn du uns weiterhin in Meetings zwingst, haben wir **gar keine Zeit** mehr, Code zu schreiben.

Zwei Menschen können die gleiche Praktik sehr unterschiedlich bewerten. Wenn sie nicht beide das Gefühl haben, sie nützt ihnen etwas, kann das die Praktik sehr ineffektiv machen.





So sind Softwareprojekte
nun mal, oder? Im richtigen Leben läuft es halt
nicht immer so wie in den Lehrbüchern, und daran
kann man auch nichts ändern. Hab ich recht?

Nein! Das richtige *Mindset* macht Praktiken viel effektiver.

Damit das klar ist: So wie Kate ihre Standups leitet, läuft es in vielen Firmen ab. Und auch wenn das Ganze nicht optimal ist, wird selbst ein solchermaßen durchgeführtes Daily Standup *trotzdem zu Ergebnissen führen*. Kate wird Probleme in ihrem Plan erkennen, und Mikes Team wird langfristig profitieren, weil die sie betreffenden Probleme nun eher früher als später angegangen werden können. Das Ganze nimmt nicht viel Zeit vom Tag weg und lohnt schon deshalb.

Aber es gibt einen großen Unterschied zwischen einem agilen Team, das nur den vorgeschlagenen Übungen folgt, und einem, das großartige Ergebnisse liefert. Entscheidend für diesen Unterschied ist das **Mindset**, das das Team in ein Projekt mit einbringt. Ob Sie es glauben oder nicht – die Einstellung, die jede Person gegenüber einer Praktik zeigt, kann diese viel effektiver machen!

Die Einstellung, die jedes Teammitglied einer Praktik wie dem Daily Standup gegenüber mitbringt, sorgt für einen großen Unterschied in der Effektivität. Aber selbst wenn alle innerlich abschalten, wird das Meeting weiterhin so effektiv sein, dass es lohnt, es durchzuführen, auch wenn es alle langweilt.

Ein besseres Mindset führt zu einer besser funktionierenden Praktik

Was wäre, wenn Kate und Mike ein anderes Mindset hätten? Und wenn jeder im Team mit einer ganz anderen Einstellung am Daily Standup teilnehmen würde?

Was würde zum Beispiel passieren, wenn Kate das Gefühl hätte, das gesamte Team **arbeite zusammen**, um das Projekt zu planen? Dann würde sie wirklich jedem einzelnen Entwickler zuhören. Wenn Kate ihre Einstellung bezüglich des Standup ändert, wird sie nicht mehr versuchen, herauszufinden, wie weit das Team von *ihrem* Plan abgewichen ist, damit *sie* die Leute korrigieren kann. Der Fokus des Meetings wird sich für sie ändern: Jetzt geht es darum, den Plan zu verstehen, an dem alle im Team gemeinsam gearbeitet haben, und Kates Aufgabe ist es dann, dem ganzen Team dabei zu helfen, effektiver arbeiten zu können.

Das ist bezüglich der Planung eine ganz andere Denkweise, eine, die Kate in keinem einzigen ihrer Kurse zum Projektmanagement gelernt hat. Ihr wurde immer erzählt, dass es ihre Aufgabe sei, einen Projektplan zu liefern und diesen dem Team mehr oder weniger aufzuoktroyieren. Sie hatte Tools, mit denen sie messen konnte, wie gut das Team ihrem Plan folgt, und strikte Prozesse, an die sie sich halten würde, wenn Änderungen daran anstünden.

Jetzt ist es ganz anders. Sie hat erkannt, dass das Daily Standup nur dann funktioniert, wenn **sie in die Zusammenarbeit mit dem Team investiert**, sodass alle im Team zusammen daran arbeiten können, das beste Vorgehen für das Projekt herauszufinden. Dann wird das Daily Standup zu einer Möglichkeit für das ganze Team, gemeinsam sicherzustellen, dass jeder solide Entscheidungen trifft und das Projekt in der Spur bleibt.

↑
Kate war bisher immer sehr gefrustet, wenn Änderungen an ihrem Projektplan notwendig waren, weil es für das Team im Allgemeinen zu spät war, noch effektiv die Richtung zu ändern.

↑
Mit dem Daily Standup arbeitet nun das ganze Team zusammen mit ihr daran, diese Änderungen zu ermitteln, sodass sie leichter umgesetzt werden können. Das ist viel effektiver!

Ich
kenne nicht alle
Antworten. Wir brauchen
dieses Meeting, um das Projekt
gemeinsam zu planen.





Also wirst du bei den Daily Standups **mir und meinem Team zuhören** und tatsächlich das Projekt entsprechend anpassen?

Und was, wenn Mike das Gefühl hätte, bei diesem Meeting ginge es nicht nur darum, Status-Updates zu liefern, sondern vielmehr darum, *zu verstehen, wie das Projekt läuft*, und jeden Tag zusammen Wege zu finden, auf denen alle besser vorankommen? Dann würde das Daily Standup für ihn auch wichtig werden.

Ein guter Entwickler hat so gut wie immer nicht nur eine Meinung zu seinem eigenen Code, sondern zum gesamten Projekt. Das Daily Standup ist seine Möglichkeit, sicherzustellen, dass das Projekt vernünftig und effizient läuft – und Mike weiß, dass dies langfristig seine Programmierarbeit für das Team lohnenswerter macht, weil der Rest des Projekts gut läuft. Und er weiß, dass **jeder zuhören wird**, wenn er während des Meetings auf ein Problem hinweist, und dass das Projekt dadurch besser laufen wird.

Am besten würde sich die Sache entwickeln, wenn Mike und sein Team feststellten, dass sie mithilfe der Daily Standups die Arbeit des nächsten Tages planen können – und dass jede einzelne Person im Team Teil des Planungsprozesses ist.

Das hat Sinn! Ein Projekt lässt sich viel besser planen, wenn jeder im Team beteiligt ist. Aber ich wette, dass das nur dann geht, wenn jeder im Meeting die ganze Zeit entsprechend aufmerksam ist.



KOPF- NUSS

Wie können Sie das Mindset in einem Team oder das einer einzelnen Person ändern? Denken Sie an bisherige eigene Projekte, in denen sich die Denkweise von jemandem geändert hat – vielleicht sogar von Ihnen selbst?

Was ist denn dann nun Agile?

Agile ist eine Sammlung von **Methoden und Vorgehensweisen**, die daraufhin optimiert sind, bei den spezifischen Problemen zu helfen, denen sich Softwareteams gegenübersehen, und die so einfach gehalten sind, dass sie sich recht unkompliziert umsetzen lassen.

Diese Methoden und Vorgehensweisen betreffen alle Bereiche der klassischen Softwareentwicklung, so auch Projektmanagement, Softwaredesign und -architektur sowie Prozessverbesserungen. Alle Methoden und Vorgehensweisen bestehen aus **Praktiken**, die vereinfacht und optimiert wurden, um sie so leicht wie möglich umsetzen zu können.



Ich habe so viel Zeit dafür aufgewendet, **meinen** Plan zu entwerfen, aber das Team weicht immer wieder davon ab. Ich kann das Daily Standup nutzen, um sicherzustellen, dass es **alles** tut, was ich ihnen sage.

Mindset versus Vorgehensweise

Agile ist *außerdem* ein **Mindset**, und das ist für viele Leute, die im Rahmen ihrer Arbeit damit noch nicht in Berührung gekommen sind, eine ziemlich neue Idee. Es stellt sich heraus, dass die Einstellung jedes einzelnen Teammitglieds bezüglich der verwendeten Praktiken *einen sehr großen Unterschied* dahin gehend bedeuten kann, wie effektiv diese sind. Das agile Mindset konzentriert sich darauf, den Menschen dabei zu helfen, untereinander Informationen zu teilen, was ihnen erleichtert, wichtige Projektentscheidungen zu treffen (statt sich nur darauf zu verlassen, dass ein Boss oder Projektmanager diese Entscheidungen trifft). Es geht darum, Planung, Design und Prozessverbesserungen für das *gesamte* Team zu öffnen. Um allen dabei zu helfen, eine effektive Denkweise zu entwickeln, besitzt jede Agile-Vorgehensweise ihren eigenen Satz an **Werten**, die Teammitglieder als Leitlinien nutzen können.

Wenn wir **alle zusammenarbeiten**, um unser Projekt zu planen, können wir das Daily Standup nutzen, um unterwegs Kurskorrekturen vorzunehmen.



**KOPF-
NUSS**

Was passiert, wenn sich ein Teammitglied während des Daily Standup ausklinkt und seinen Kollegen nicht zuhört?



Spitzen Sie Ihren Bleistift

Hier sehen Sie ein paar der Probleme, die Kate, Ben und Mike in einem Daily Standup vorgebracht haben. Daneben haben wir die Namen einiger Praktiken geschrieben, die in agilen Teams gern eingesetzt werden. Machen Sie sich keine Sorgen, wenn Sie davon noch nichts gehört haben – Sie werden im Buch noch viel darüber erfahren, daher haben wir jede Praktik nur kurz beschrieben. Versuchen Sie, jedes der Probleme mit einer Praktik zu verbinden, die bei der Lösung helfen kann.



»Wir haben gerade Stunden damit verbracht, uns durch Spaghetti-Code zu wühlen, um diesen Fehler zu finden!«



»Okay, wir sind die User Stories durchgegangen. Lasst uns jetzt herausfinden, wie sie zusammenpassen, sodass wir die Arbeit für die nächsten Wochen planen können.«



»Wir scheinen bei jedem Release immer wieder die gleichen Probleme zu bekommen.«



»Ich habe gerade einer unserer Videostreaming-Anwenderinnen ein neues Feature demonstriert. Sie sagt, dass damit das Problem, für das es gedacht ist, überhaupt nicht gelöst würde.«



»Ich dachte, wir wären am Freitag mit dem Aktualisieren der Song-Datenbank fertig. Jetzt erzählt ihr mir, dass es drei Wochen länger dauert?«

Eine **Retrospektive** ist ein Meeting, in dem jeder darüber spricht, wie der letzte Teil des Projekts verlaufen ist und was daraus gelernt werden kann.

Eine **User Story** ist eine Möglichkeit, eine sehr spezifische Anforderung eines Anwenders auszudrücken – meist in ein paar Sätzen auf einem Post-it oder einer Karteikarte.

Ein **Taskboard** ist ein agiles Planungstool, bei dem User Stories an ein Board gepinnt und abhängig von ihrem Status kategorisiert werden.

Ein **Burndown Chart** ist ein Liniendiagramm, in dem die Menge der verbleibenden Arbeit im Projekt aufgetragen wird. Diese »brennt« bis auf null herunter, wenn die Arbeit erledigt ist.

Entwickler beheben Codeprobleme, indem sie fortlaufend ihren Code **refaktorisieren** oder die Codestruktur verbessern, ohne dessen Verhalten zu verändern.

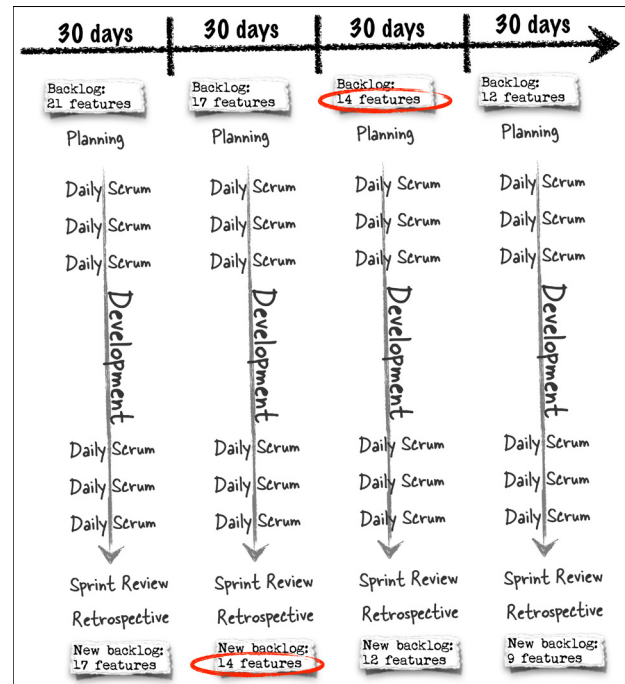
Es ist kein Problem, wenn Sie von diesen Praktiken bisher noch nichts gehört haben. Sie werden in den nächsten Kapiteln mehr darüber erfahren.

Scrum ist das am häufigsten genutzte agile Konzept

Es gibt viele Möglichkeiten, wie Teams agil sein können, und es gibt eine lange Liste mit Methoden und Vorgehensweisen, die agile Teams einsetzen. Aber in vielen Umfragen der letzten Jahre hat sich herausgestellt, dass das am häufigsten eingesetzte agile Konzept **Scrum** ist – ein Framework zur Softwareentwicklung, das sich auf Projektmanagement und Produktentwicklung fokussiert.

Wenn ein Team Scrum nutzt, folgt jedes Projekt dem gleichen grundlegenden Muster. Es gibt in einem Scrum-Projekt drei wichtige Rollen: Der **Product Owner** (wie Ben) arbeitet mit dem Team zusammen, um ein **Product Backlog** zu verwalten, der **Scrum Master** hilft dem Team, Hindernisse zu umschiffen, und darüber hinaus gibt es die **Mitglieder des Development-Teams** (alle anderen im Team). Das Projekt ist in **Sprints** unterteilt – gleich lange Zeiträume (meist zwei Wochen oder 30 Tage), die dem Scrum-Muster folgen. Zu Beginn eines Sprints führt das Team eine **Sprint-Planung** durch, um herauszufinden, welche Features aus dem Product Backlog es während des Sprints bauen wird. Dies wird als **Sprint Backlog** bezeichnet, und das Team arbeitet während des Sprints daran, alle diese Features zu bauen. Täglich führt das Team ein kurzes Meeting durch – das **Daily Scrum**. Am Ende des Sprints wird die funktionsfähige Software dem Product Owner und den Stakeholdern im **Sprint Review** vorgeführt, und das Team hält eine **Retrospektive** ab, um zu besprechen, was es gelernt hat.

Wir werden Scrum detailliert in den Kapiteln 3 und 4 behandeln und Ihnen dabei nicht nur zeigen, wie es Teams dabei hilft, bessere Software zu bauen und mehr erfolgreiche Projekte durchzuführen, sondern wir werden es auch nutzen, um wichtige Konzepte und Ideen vorzustellen, die bei allen agilen Teams zum Einsatz kommen.



XP und Lean/Kanban

Während Scrum die beliebteste agile Vorgehensweise ist, verfolgen viele Teams andere Ansätze. Die nächsthäufig genutzte Methodik ist **XP**, eine auf die Softwareentwicklung und Programmierung fokussierte Vorgehensweise, die gern kombiniert mit Scrum eingesetzt wird. Andere Teams gehen Agile mit **Lean** und **Kanban** an – einer Denkweise, mit der Sie verstehen, wie heutzutage Software gebaut wird, und einer Methode, die dabei hilft, Ihre Situation in Zukunft besser zu gestalten. Sie werden XP und Lean/Kanban in den Kapiteln 5 und 6 kennenlernen.



Entspannen
Sie sich

Von den ganzen neuen Begriffen etwas überwältigt?

Wir werden neue Begriffe **fett drucken**, wenn sie das erste Mal auftauchen. Das ist auf dieser Seite ziemlich häufig vorgekommen – und sollten ein paar dabei sein, die Ihnen nichts sagen, ist das in Ordnung! Neue Ideen in einem

Kontext zu sehen, ist hilfreich, um sie besser zu verstehen, wenn Sie später die Details kennenlernen. Das ist Teil der »Von Kopf bis Fuß«-Neurowissenschaft, die dieses Buch so hilfreich macht!



Spitzen Sie Ihren Bleistift

Lösung

Teams können das Wiederholen von Fehlern vermeiden, indem sie auf das Projekt zurückblicken und darüber reden, was gut lief und was verbessert werden könnte.



»Wir haben gerade Stunden damit verbracht, uns durch Spaghetti-Code zu wühlen, um diesen Fehler zu finden!«

Ein Taskboard ist ein großartiges Hilfsmittel, damit das gesamte Team den gleichen Überblick über das Projekt erhalten kann.



»Okay, wir sind die User Stories durchgegangen. Lasst uns jetzt herausfinden, wie sie zusammenpassen, sodass wir die Arbeit für die nächsten Wochen planen können.«



»Wir scheinen bei jedem Release immer wieder die gleichen Probleme zu bekommen.«



»Ich habe gerade einer unserer Videostreaming-Anwenderinnen ein neues Feature demonstriert. Sie sagt, dass damit das Problem, für das es gedacht ist, überhaupt nicht gelöst würde.«

Wenn jeder im Team die Anwender und ihre Bedürfnisse versteht, kann er bessere Software bauen, die die Benutzer lieben.



»Ich dachte, wir wären am Freitag mit dem Aktualisieren der Song-Datenbank fertig. Jetzt erzählt ihr mir, dass es drei Wochen länger dauert?«

Eine **Retrospektive** ist ein Meeting, in dem jeder darüber spricht, wie der letzte Teil des Projekts verlaufen ist und was daraus gelernt werden kann.

Eine **User Story** ist eine Möglichkeit, eine sehr spezifische Anforderung eines Anwenders auszudrücken – meist in ein paar Sätzen auf einem Post-it oder einer Karteikarte.

Ein **Taskboard** ist ein agiles Planungstool, bei dem User Stories an ein Board gepinnt und abhängig von ihrem Status kategorisiert werden.

Ein **Burndown Chart** ist ein Liniendiagramm, in dem die Menge der verbleibenden Arbeit im Projekt aufgetragen wird. Diese »brennt« bis auf null herunter, wenn die Arbeit erledigt ist.

Dies ist eine **XP-Praktik**. Manche Projektmanager sind überrascht, wenn sie erstmals eine agile Praktik kennenlernen, die sich auf den Code konzentriert, statt Planung und Durchführung des Projekts anzugehen.

Entwickler beheben Codeprobleme, indem sie fortlaufend ihren Code **refaktorisieren** oder die Codestruktur verbessern, ohne dessen Verhalten zu verändern.

Es gibt keine Dummen Fragen

F: Es hört sich so an, als ob sich Scrum, XP und Lean/Kanban sehr unterscheiden. Wie kann das dann alles Agile sein?

A: Scrum, XP und Lean/Kanban konzentrieren sich auf sehr unterschiedliche Bereiche. Bei Scrum geht es vor allem um Projektmanagement: Welche Arbeit wird erledigt, und wie ist sichergestellt, dass diese zu den Bedürfnissen der Anwender und Stakeholder passt? XP fokussiert sich auf Softwareentwicklung: das Bauen qualitativ hochwertigen Codes, der gut designt und leicht zu warten ist. Lean/Kanban ist eine Kombination aus dem Lean Mindset und der Kanban-Methode. Teams können dies nutzen, um sich auf das kontinuierliche Verbessern ihrer Vorgehensweise beim Entwickeln von Software zu konzentrieren.

Mit anderen Worten: Scrum, XP und Lean/Kanban beziehen sich auf drei verschiedene Bereiche der Softwareentwicklung: Projektmanagement, Design und Architektur sowie

Prozessverbesserung. Daher ist es durchaus sinnvoll, dass es drei verschiedene Praktiken gibt.

Im nächsten Kapitel werden Sie erfahren, was die Praktiken gemeinsam haben: **gleiche Werte und Prinzipien**, die Teams dabei helfen, ein agiles Mindset zu übernehmen.

F: Ist das nicht nur alter Wein in neuen Schläuchen? Scrum-Sprints sind doch einfach Milestones und Projektphasen, oder?

A: Wenn Sie das erste Mal einer agilen Vorgehensweise wie Scrum begegnen, ist es verständlich, nach Dingen Ausschau zu halten, die Ihnen vertraut sind – und das ist auch gut so! Sofern Sie bereits länger in Teams arbeiten, *sollte* Ihnen eine Menge von Agile bekannt vorkommen. Ihre Teams bauen etwas, und Sie und Ihre Kollegen machen mit Sicherheit schon vieles gut und richtig, was Sie gar nicht ändern wollen (noch nicht!).

Aber man begeht leicht den Fehler, zu glauben, dass das, was an Agile vertraut vorkommt, genau das ist, was Sie schon kennen. So sind zum Beispiel Scrum-Sprints *nicht das Gleiche wie Projektphasen*. Es gibt viele Unterschiede zwischen Phasen oder Milestones im klassischen Projektmanagement und Sprints in Scrum.

So werden zum Beispiel in einem typischen Projektplan alle Projektphasen zu Beginn des Projekts geplant; in Scrum wird immer nur der nächste Sprint im Detail vorbereitet. Dieser Unterschied kann für ein Team, das an klassisches Projektmanagement gewohnt ist, verwirrend sein.

Sie werden noch viel mehr über Scrum-Planung und die Unterschiede zu den von Ihnen bisher genutzten Techniken lernen. Bis dahin bleiben Sie einfach aufgeschlossen – und passen auf, dass Sie nicht sagen: »Das ist doch wie etwas, das ich schon kenne!«

Punkt für Punkt

- Viele Teams, die Agile übernehmen wollen, beginnen mit dem **Daily Standup**, einem Meeting mit dem gesamten Team, in dem jeder steht, um es kurz zu halten.
- Agile ist ein Satz von **Methoden und Vorgehensweisen**, aber es ist *gleichzeitig* ein **Mindset** oder eine Einstellung, die von allen im Team geteilt wird.
- Das Daily Standup ist viel effektiver, wenn alle im Team das richtige **Mindset** haben – jeder hört den anderen zu, und alle arbeiten zusammen, um sicherzustellen, dass das Projekt in der Spur bleibt.
- Jede Agile-Methodik bringt einen Satz von **Werten** mit, die dabei helfen, das Team zu einem möglichst effektiven Mindset zu bringen.
- Wenn Teammitglieder gemeinsamen **Prinzipien** folgen und die gleichen **Werte** haben, kann das die genutzte Methode sehr viel effektiver machen.
- **Scrum**, ein Framework, das sich auf Projektmanagement und Produktentwicklung konzentriert, ist der bekannteste Ansatz in Agile.
- In einem Scrum-Projekt teilt das Team die Arbeit in **Sprints** oder Zeitabschnitte gleicher Länge auf (meist 30 Tage), die dem Scrum-Muster folgen.
- Jeder Sprint beginnt mit einer **Sprint-Planungssession**, um zu entscheiden, was gebaut wird.
- Während des Sprints arbeitet das Team am Projekt, und jeden Tag hält es ein kurzes Meeting mit dem Namen **Daily Scrum** ab.
- Am Sprint-Ende veranstaltet das Team ein **Sprint Review** mit den Stakeholdern, um die von ihm gebaute lauffähige Software zu demonstrieren.
- Um den Sprint abzuschließen, hält das Team eine **Retrospektive** ab, in der es auf den Sprint zurückblickt und überlegt, wie es als Team besser werden kann.



Tun Sie die Idee, dass das Mindset wichtig ist, nicht einfach ab!

Viele Leute – insbesondere Hardcore-Entwickler – schalten sofort ab, wenn sie Wörter wie Mindset, Werte oder Prinzipien hören. Das gilt besonders für Coder, die sich gern alleine in einen Raum einschließen und mit niemandem reden. Wenn Sie so denken, versuchen Sie bitte trotzdem, diesen Ideen eine Chance zu geben. Denn schließlich wurde bereits sehr viel großartige Software auf diesem Weg gebaut, also muss da doch etwas dran sein, oder?



Spitzen Sie Ihren Bleistift

Welche dieser Szenarien sind Beispiele für die Anwendung von Praktiken und welche für die Anwendung von Prinzipien? Machen Sie sich keine Sorgen, wenn Sie manche dieser Praktiken noch nicht kennen – nutzen Sie einfach den Kontext, um die richtige Antwort herauszufinden. (Das ist eine gute Fähigkeit, an der es sich zu feilen lohnt, wenn Sie eine Zertifizierungsprüfung ablegen wollen!)

1. Kate weiß, dass ein **Face-to-Face-Gespräch** der effektivste Weg ist, wichtige Informationen über das Projekt an ihr Team zu kommunizieren.

☐ Prinzip

☐ Praktik

2. Mike und sein Team wissen, dass die Anwender vermutlich noch ihre Meinung ändern werden und dass diese Änderungen ihren Code ins Chaos stürzen können. Daher nutzen sie ein **inkrementelles Design**, um sicherzustellen, dass sich der gebaute Code später leicht ändern lässt.

☐ Prinzip

☐ Praktik

3. Ben nutzt eine **Persona**, um einen typischen Anwender zu modellieren, weil er weiß, dass das Team umso bessere Software bauen wird, je mehr es den Benutzer versteht.

☐ Prinzip

☐ Praktik

4. Mike stellt immer sicher, dass sein Team an etwas arbeitet, das er Kate und Ben demonstrieren kann, denn er weiß, dass **lauffähige Software** der beste Weg ist, den Fortschritt des Teams zu zeigen.

☐ Prinzip

☐ Praktik

5. Kate möchte die Art und Weise verbessern, wie das Team Software baut, daher bringt sie es dazu, **sich gemeinsam zu verbessern und durch Ausprobieren weiterzuentwickeln**, indem sich das Team Änderungen überlegt, die es an seinem Prozess vornehmen kann, und Daten nutzt, um herauszufinden, ob diese Änderungen die Dinge wirklich verbessert haben.

☐ Prinzip

☐ Praktik

6. Mike und sein Team **akzeptieren Veränderungen**, indem sie Code bauen, der sich später noch leicht ändern lässt.

☐ Prinzip

☐ Praktik

—————> Antworten auf Seite 20



Wow! Wir haben
noch nie so gut zusammengearbeitet. Das Daily
Standup hat alles verändert!

Kate: Dieses Projekt läuft so viel besser als die anderen Projekte. Und alles nur wegen eines kleinen Meetings pro Tag!

Mike: Na ja, das würde ich nicht sagen.

Kate: Komm schon, Mike! Sei nicht so pessimistisch.

Mike: Nein, ernsthaft. Schau mal, du glaubst doch nicht wirklich, dass du die Erste bist, die versucht, unsere Projektprobleme zu lösen, indem sie zusätzliche Meetings ansetzt, oder?

Kate: Nun, ich ... ähm ...

Mike: Wir haben ein paar wirklich gute Ergebnisse erzielt, daher will ich ehrlich zu dir sein. Als du mit diesen Daily Standups begonnen hattest, war fast das ganze Team unglücklich.

Kate: Echt?

Mike: Ja. Erinnerst du dich nicht, dass die meisten von uns in den ersten anderthalb Wochen eigentlich nur auf ihre Handys gestarrt haben?

Kate: Ja, schon. Ich glaube, das war nicht sehr hilfreich. Und wenn ich ehrlich bin, habe ich tatsächlich darüber nachgedacht, das Ganze wieder abzublasen.

Mike: Aber dann hat eine meiner Programmiererinnen dieses ernste Architekturproblem aufgebracht. Jeder hat zugehört, weil sie wirklich gut ist und alle ihre Meinung achten.

Kate: Stimmt. Wir mussten eine größere Änderung vornehmen, und ich habe zwei der Features aus dem Release geschmissen, um die Zeit dafür zu haben.

Mike: Ja! Das war wirklich wichtig. Wenn wir früher so ein Problem hatten, mussten wir bis spät in die Nacht arbeiten, um die Folgen auszubügeln. Zum Beispiel als wir herausfanden, dass der Analysealgorithmus für das Listener-Feedback Ärger machte.

Kate: Oh ja, das war übel. Normalerweise erfahre ich von solchen Problemen immer erst, nachdem wir all das zugesagt haben, was wir gar nicht liefern können. Dieses Mal haben wir das Problem frühzeitig erkannt, und ich konnte mit Ben daran arbeiten, die Erwartungen unserer Anwender in den Griff zu bekommen und euch die Zeit zu verschaffen, die ihr für eine Lösung brauchtet.

Mike: Wir werden jetzt solche Probleme auf jeden Fall immer gleich aufs Tapet bringen, sobald wir sie bemerken.

Kate: Was? So etwas passiert häufiger?!

Mike: Machst du Witze? Ich hatte noch nie ein Projekt, in dem es keine unangenehme Überraschung gab, nachdem wir erst einmal mit dem Programmieren begonnen hatten. So laufen Softwareprojekte in der Realität nun einmal ab, Kate.

Es sieht so aus, als ob Kate erkannt hat, dass Softwareprojekte in der Realität viel chaotischer und komplizierter sind als auf dem Papier. Bisher konnte sie einfach ihren Plan entwerfen und das Team dann zwingen, ihn umzusetzen ... und wenn etwas schiefging, war es dessen Problem, nicht theirs.

Andererseits lief dieses Projekt viel besser als ihre bisherigen. Sie musste viel härter dafür arbeiten, mit den Problemen umzugehen, aber sie erhielt auch bessere Ergebnisse!