



Jürgen Hoffmann · Stefan Rook

Agile Unternehmen

Veränderungsprozesse gestalten,
agile Prinzipien verankern,
Selbstorganisation
und neue Führungsstile etablieren



dpunkt.verlag



Dr. Jürgen Hoffmann hat seit 2003 Erfahrung mit agilen Methoden und Scrum. Nach der ersten großen Scrum-Einführung in Deutschland bei der WEB.DE AG hat er in den Rollen als Coach, Trainer, Product Owner, Scrum Master und Teammitglied in unterschiedlichen Branchen und Firmengrößen gearbeitet. Diese Erfahrung aus Branchen wie Automotive, Energie, Finanzen, IT & Internet mit Soft- und Hardwareentwicklung fließen in jeden Beratungsprozess ein. Heute arbeitet er als Geschäftsführer und Managementberater bei der Emendare GmbH & Co KG, die er 2013 mitgründete. Als Certified Scrum Trainer (CST) und Certified Enterprise Coach (CEC) ist er Teil einer starken Gemeinschaft von über 280 Scrum-Trainern und Coaches der weltweiten Scrum Alliance®, die in ständigem Austausch miteinander ihre Trainings und Beratungsideen kontinuierlich verbessern und um aktuelle Fragestellungen ergänzen.



Dipl.-Inform. Stefan Roock ist Gründungsmitglied der it-agile GmbH. Ihm ist es in seiner Beratungstätigkeit wichtig, dass sich wirklich etwas ändert – hin zu erfolgreichen Unternehmen mit zufriedenen Mitarbeitern, die sich immer neuen Herausforderungen stellen. Stefan hat seit 1999 die Verbreitung agiler Ansätze in Deutschland maßgeblich mit beeinflusst. Zunächst hat er als Entwickler, später als Scrum Master und Product Owner in Scrum-Teams gearbeitet. Heute gibt er seine Erfahrung als Berater und Trainer weiter und hilft Unternehmen dabei, agiler zu werden. Neben seiner Beratungstätigkeit für it-agile ist er regelmäßiger Sprecher zu agilen Themen auf Konferenzen, schreibt Zeitschriftenartikel und hat mehrere Bücher veröffentlicht.

Jürgen Hoffmann · Stefan Rook

Agile Unternehmen

**Veränderungsprozesse gestalten,
agile Prinzipien verankern, Selbstorganisation
und neue Führungsstile etablieren**



dpunkt.verlag

Jürgen Hoffmann
juergen.hoffmann@emendare.de

Stefan Roock
stefan.roock@it-agile.de

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Birgit Bäuerlein
Herstellung: Susanne Bröckelmann
Umschlaggestaltung: Helmut Kraus, www.exclam.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 978-3-86490-399-1
PDF 978-3-96088-437-8
ePub 978-3-96088-438-5
mobi 978-3-96088-439-2

1. Auflage 2018
Copyright © 2018 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 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

*Jürgen:
Für Andrea, Linus und Lenka*



*Stefan:
Für Silke, Mika, Levi und Mailin*

Inhaltsübersicht

1	Einleitung	1
2	Begeisterte Kunden	13
3	Wertschöpfung als Teamaufgabe	49
4	Unterstützende Organisation	63
5	Organisationsentwicklung	123
	Anhang	159
A	User Research	161
B	Große Produkte mit dem LeSS-Framework entwickeln	183
	Literaturverzeichnis	191
	Index	197

Inhaltsverzeichnis

1	Einleitung	1
1.1	Echte Agilität	2
1.2	Agile Fluency	5
1.3	Fokus dieses Buches: echte Agilität oder auch »Optimize Value«	7
1.3.1	Eigenschaften von »Optimize Value«-Unternehmen	8
1.3.2	»Focus on Value« und »Deliver Value«: Literaturempfehlungen	9
1.4	An wen richtet sich das Buch?	10
1.5	Überblick über das Buch	10
1.6	Danksagung	11
2	Begeisterte Kunden	13
2.1	Definieren, was Wert bedeutet und schafft	13
2.1.1	Wert aus Kundensicht	13
2.1.2	Bedürfnisse identifizieren	16
2.2	Drei Horizonte für Wachstum und Innovation	17
2.2.1	Herausforderungen bei der Umsetzung des 3-Horizonte-Modells	19
2.2.2	Das 3-Horizonte-Modell und agile Entwicklung	20
2.2.3	Wert bedeutet in jedem Horizont etwas anderes	21
2.3	Wert in Horizont 1	22
2.3.1	Umsatz als Indikator für Wertschöpfung	22
2.3.2	Net Promoter System (NPS)	22
2.3.3	NPS: Bitte beachten	24
2.3.4	Produktreview/Sprint-Review	25
2.4	Wert in Horizont 2	27
2.4.1	Produktvision	27
2.4.2	Produktreview/Sprint-Review	30
2.4.3	Design Sprints	30

2.5	Wert in Horizont 3	32
2.5.1	Vorgehen in Horizont 3 zur Produkt-/Serviceentwicklung	33
2.5.2	Konkrete Techniken zum Einsatz in Horizont 3	35
2.6	Organisation für das 3-Horizonte-Modell	36
2.6.1	Freiraum in Horizont 1	36
2.6.2	Freiraum in Horizont 2	38
2.6.3	Freiraum in Horizont 3	39
2.6.4	Übergang von Horizont 3 nach Horizont 2	39
2.6.5	Übergang von Horizont 2 nach Horizont 1	41
2.6.6	Personalstrategien der Horizonte	42
2.6.7	Entwicklung in den drei Horizonten	44
2.6.8	Produkt-Roadmaps in den drei Horizonten	45
2.7	Das Kapitel in Stichpunkten	47
3	Wertschöpfung als Teamaufgabe	49
3.1	Eigenständige Teams	50
3.1.1	Manager-led Teams	51
3.1.2	Self-managing Teams	51
3.1.3	Self-designing Teams	52
3.1.4	Self-governing Teams	53
3.2	Funktionsübergreifende Teams	55
3.2.1	Zusammensetzung von Teams	56
3.2.2	Product-Owner-Rolle	58
3.2.3	Teambegleitung	59
3.2.4	Effizienz vs. Effektivität	59
3.3	Entscheidungen im Team	61
3.4	Das Kapitel in Stichpunkten	61
4	Unterstützende Organisation	63
4.1	Störungen durch das Unternehmen	64
4.2	Dezentrale Strukturen	65
4.2.1	Zellmodell in der Praxis der Softwareentwicklung	69
4.2.2	Mehr als ein Team pro Zelle	72
4.2.3	Alles Illusion?	73
4.3	Alignment bei dezentralen Strukturen	75
4.3.1	Management by Objectives (MbO)	76
4.3.2	Objectives and Key Results (OKR)	78
4.3.3	MbO-Beispiel – so bitte nicht	79
4.3.4	MbO-Beispiel – besser	81
4.3.5	Nutzen und Gefahren von Management by Objectives	82
4.3.6	Ziele ohne die MbO-Gefahren	86

4.4	Feedbackschleifen statt statischer Ziele	87
4.4.1	Feedbackschleife für den Umweltschutz	87
4.4.2	Feedbackschleifen bei Command & Control-Strukturen . . .	90
4.4.3	Feedbackschleifen in einem agilen Unternehmen	91
4.4.4	Das Unternehmen als Organismus	93
4.5	Übergreifende Entscheidungsfindung bei dezentralen Strukturen . . .	93
4.5.1	Konsent	93
4.5.2	Advice-Prozess	95
4.5.3	Das Unternehmen verstehen	98
4.5.4	Bewertung und Vergleich von Konsent und Advice-Prozess	98
4.6	Neue Rolle für Führungskräfte	99
4.6.1	Klassische Mitarbeiterführung	99
4.6.2	Probleme klassischer Führung in einer dynamischen Welt	101
4.6.3	Supporting Lines statt Reporting Lines	102
4.6.4	Verteilte Führung	103
4.6.5	Situative Führung	104
4.6.6	Ausbildung	105
4.7	Fallbeispiele zu moderner Mitarbeiterführung	107
4.7.1	ImmobilienScout24	107
4.7.2	siggate	111
4.7.3	it-agile	114
4.7.4	Zusammenfassung der Fallbeispiele für Mitarbeiterführung	117
4.8	Unternehmenskultur	118
4.8.1	Unternehmenskultur und agiles Arbeiten	119
4.9	Das Kapitel in Stichworten	120
5	Organisationsentwicklung	123
5.1	Organisationsentwicklung als komplexe Aufgabe	123
5.1.1	Satir Change Model	124
5.2	Erfolgsfaktoren für agile Organisationsentwicklung	127
5.2.1	Erfahrungen mit dem Kotter Change Model	129
5.3	Steuerung iterativer Organisationsentwicklung	130
5.3.1	Das agile Transitionsteam	131
5.3.2	Transition Backlog und Product Owner	131
5.3.3	Produktvision und Produktinkremente des Transitionsteams	133
5.3.4	Transitionsteam: Besetzung und Rollen	135

5.3.5	Sprints im Transitionsteam	136
5.3.6	Einbindung ins Unternehmen	137
5.3.7	Weitere Probleme im Transitionsteam	137
5.4	Organisationsentwicklung über Experimente	139
5.4.1	Der PDCA-Zyklus	139
5.4.2	PDCA in der Praxis	141
5.4.3	Organisationsentwicklung als Abfolge von Experimenten	142
5.4.4	Safe-to-Fail-Experimente	143
5.4.5	Experimente erleichtern die Veränderung	143
5.4.6	Organisation der Organisationsentwicklung	144
5.5	Kultur der kontinuierlichen Verbesserung	146
5.5.1	Transparenz in alle Richtungen	147
5.6	Orientierung mit einem Nordstern (True North)	148
5.6.1	Nordstern bei Toyota	149
5.6.2	Nordsterne für die Wissensarbeit	150
5.6.3	Eigenschaften eines guten Nordsterns	151
5.6.4	Arbeiten mit dem Nordstern	152
5.6.5	Nordstern und der PDCA-Zyklus	153
5.6.6	Die A3-Technik	154
5.6.7	Der Weg zum eigenen Nordstern	155
5.7	Das Kapitel in Stichworten	157

Anhang

159

A	User Research	161
A.1	Design Thinking konkret	161
A.1.1	Team	163
A.1.2	Raum	164
A.1.3	Prozess	165
A.2	Design Sprints	166
A.3	Lean Startup	174
A.3.1	Die Historie und das Umfeld	175
A.3.2	Kundenbedürfnisse verstehen und Lösung validieren	175
A.3.3	Den Markt validieren	176
A.3.4	Minimum Viable Product (MVP)	177
A.3.5	Pivots	179
A.3.6	Skalierung	180
A.3.7	Fallbeispiel bei it-agile	180
A.3.8	Fazit zu Lean Startup	181
A.4	Das Kapitel in Stichworten	182

B	Große Produkte mit dem LeSS-Framework entwickeln	183
B.1	Veränderung folgt Notwendigkeiten	183
B.2	Agile Skalierungsprinzipien nach LeSS	183
B.3	Durchstarten zur Skalierung	185
B.3.1	Schule alle Beteiligten	186
B.3.2	Definiere das »Produkt«	186
B.3.3	Definiere, wann es »fertig« ist	187
B.3.4	Baue angemessen strukturierte Teams auf	187
B.3.5	Nur der Product Owner versorgt die Teams mit Arbeit . . .	188
B.4	Ein Produkt – mehrere Teams	189
B.5	Das Kapitel in Stichworten	190
	Literaturverzeichnis	191
	Index	197

1 Einleitung

Märkte verändern sich immer schneller. Dadurch stehen viele Unternehmen heute vor der Herausforderung, schneller und flexibler auf diese Änderungen zu reagieren. Menschen mit Verantwortung in Unternehmen stellen die große Frage: »Wie werden wir agiler?«

Vor einigen Jahren hat Ken Schwaber bei einem Vortrag in Karlsruhe eine Idee dazu präsentiert. Das CIF (Continuous Improvement Framework) schlägt einen Satz von Unternehmensmetriken vor. In regelmäßigen Abständen prüfen Mitarbeiter aller Unternehmensebenen den Fortschritt anhand dieser Metriken und beschließen Verbesserungsmaßnahmen. Beispiele für diese Metriken waren »Die Anzahl der Kunden« oder »Die Zeitdauer zwischen zwei Versionen eines Produktes«. Bei diesem Vorschlag wird das Unternehmen als Blackbox angesehen. Ken Schwaber machte dabei keine Vorschläge zur inneren Gestalt des Unternehmens. Dieses Buch dagegen ist voll von solchen Schritten. Der Leser kann, wie mit einer Lupe, einzelne Organisationsbereiche und Situationen fokussieren und bekommt dazu Ideen und Handlungsanweisungen für den individuellen Weg zu mehr Agilität.

Zu diesem Prozess fügen wir als Katalysator unsere Erfahrung aus diversen Unternehmen der verschiedensten Branchen hinzu. Wir, Stefan Roock und »Mentor« Jürgen Hoffmann, sind als ihre Trainer, Berater und Coaches dicht am Puls der Firmen. Unserer Beobachtung nach gibt es auch nicht das agile Unternehmen. Jedes hat andere Mitarbeiter, Herausforderungen und eine andere Geschichte. Der Versuch, von einem auf den anderen Tag einfach alle Ideen und Werkzeuge aus der Agile Community mit einem Schlag einzuführen, würde die Menschen, die Produkte und damit das ganze Unternehmen überfordern.

Eines der Prinzipien beim Einsatz von Agilität ist das ständige Experiment: »Try, inspect and adapt« – »Ausprobieren, Erfolgskontrolle und Anpassung«. Das ist der Weg, auf dem die hier vorgeschlagenen Ideen auch Ihr Unternehmen beleben und erfolgreicher machen können.

1.1 Echte Agilität

Unserer Meinung nach stellt ein einfacher Zyklus den Kern agiler Entwicklung dar (siehe Abb. 1–1): Kunden haben Probleme, die ein selbstorganisiertes autonomes Team löst. Dieser Zyklus muss möglichst schnell und in direkter Interaktion stattfinden.

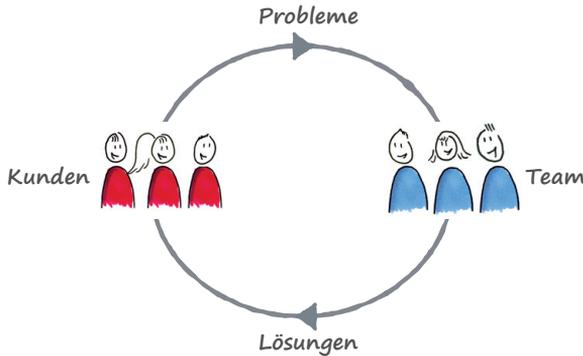


Abb. 1-1 Kernidee agiler Entwicklung

Ein agiles Entwicklungsframework wie Scrum soll die agile Kernidee unterstützen (siehe Abb. 1–2). Wichtig ist, dass dabei stets die agile Kernidee im Vordergrund bleibt und nicht vom Scrum-Framework in den Hintergrund verdrängt wird.

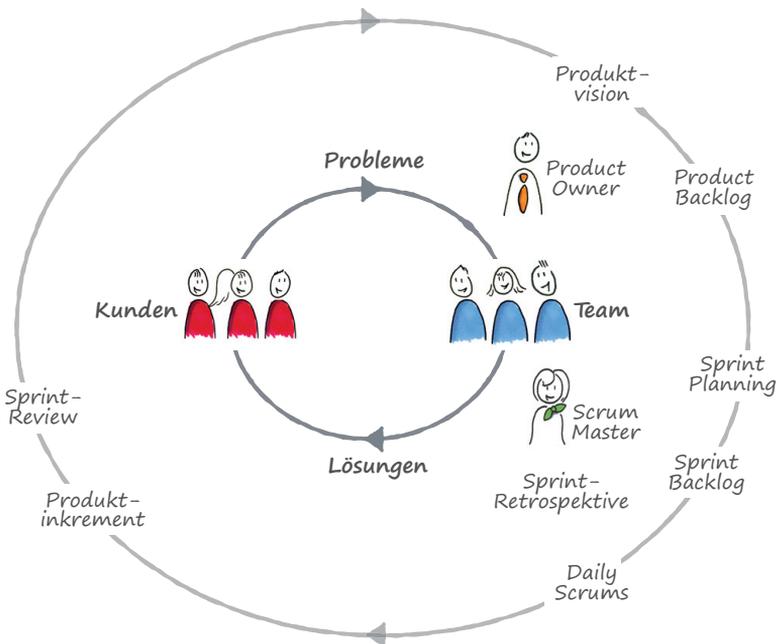


Abb. 1-2 Scrum zur Unterstützung der agilen Kernidee

Das klingt alles trivialer, als es ist. Faktisch haben die meisten Unternehmen sich Strukturen gegeben, die den beschriebenen Zyklus massiv stören (siehe Abb. 1–3). Die Beispiele für diese Störungen sind vielfältig:

- Das Team hat keinen direkten Kontakt zu den Kunden.
- Die Teammitglieder werden ständig zwischen Teams hin und her verschoben.
- Vorgesetzte geben den Teammitgliedern Aufgaben, die sie von der Lösung kundenrelevanter Probleme abhalten.
- Teammitglieder müssen Stage-Gate-Prozesse einhalten, die die Problemlösung für Kunden ganz erheblich verzögern.
- Etc.

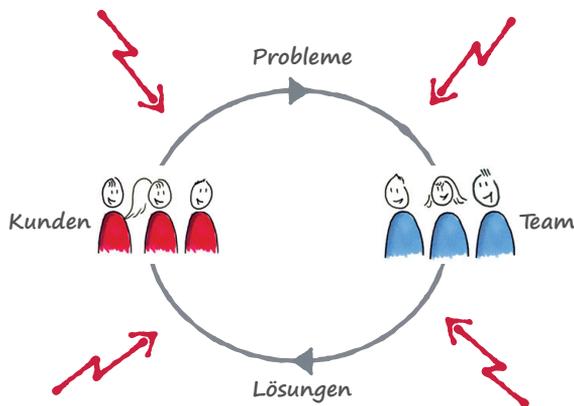


Abb. 1–3 Unternehmensstrukturen stören die agile Kernidee.

In der Konsequenz stecken die meisten »agilen« Implementierungen noch in den Kinderschuhen (auch die, die sich bereits seit Jahren daran versuchen). So findet sich in vielen Fällen die Struktur aus Abbildung 1–4: Das Team ist selbstorganisiert, hat aber keinen direkten Kundenkontakt. Den Kundenkontakt hält z. B. das Produktmanagement und überführt die Kundenprobleme in Anforderungen, die das Team dann umsetzt. Die entwickelte Software liefert das Team nicht direkt an Kunden aus, weil das Team keine vollständig lieferbare Software herstellen kann (es fehlen z. B. die Integrationstests).

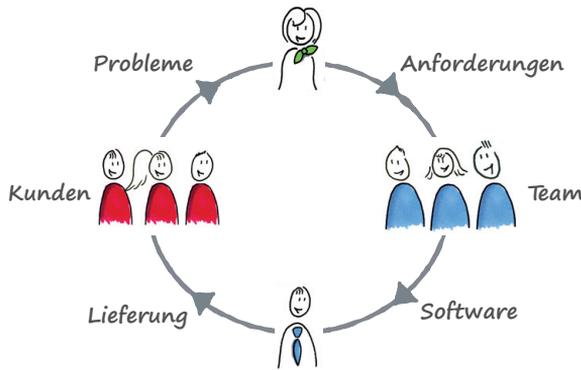


Abb. 1-4 Selbstorganisiertes Team

In dieser Struktur ist mit dem selbstorganisierten Team zwar bereits eine wichtige agile Idee implementiert. Die Wirksamkeit des Teams bleibt aber beschränkt.

Häufig entwickelt sich diese Struktur technisch weiter. Das Team liefert die Lösung direkt an den Kunden aus, nachdem es die Fähigkeit erworben hat, in kurzen Abständen wirklich lieferbare Software zu erstellen (siehe Abb. 1-5). Die extremste Ausprägung findet sich heute im Continuous Deployment – die Software wird nach jeder Änderung sofort (also mehrmals täglich) an Kunden ausgeliefert.

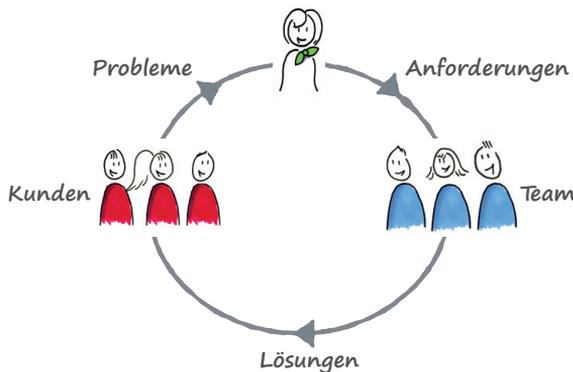


Abb. 1-5 Lieferndes Team

Wenn wir dann noch den oberen Teil des Zyklus von der Indirektion befreien und das Team direkt mit den Kunden über ihre Bedürfnisse und Probleme sprechen lassen, landen wir bei echter Agilität (siehe Abb. 1-6).

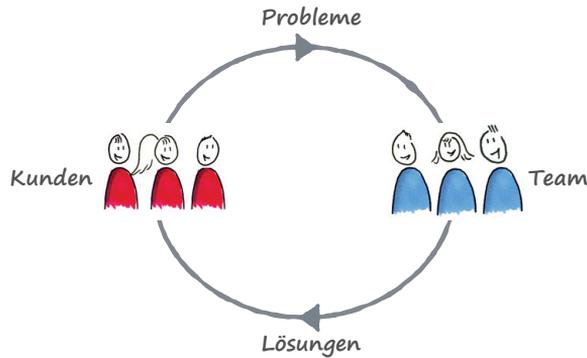


Abb. 1-6 Kundenwertoptimierendes Team

Vor diesem Hintergrund kann man agil arbeitende Teams in einem Satz folgendermaßen definieren: *Business-fokussierte Teams, die für Produkt und Prozess Verantwortung übernehmen.*

In diesem Buch beschreiben wir, was über selbstorganisierte, liefernde Teams hinaus für echte Agilität notwendig ist. Wir gehen also davon aus, dass Sie als Leser bereits selbstorganisierte, liefernde Teams erreicht haben oder sich anderswo die Informationen besorgen, die dafür notwendig sind.

Wichtig ist uns hier noch, dass die beschriebenen »Stufen« nicht sequenziell durchlaufen werden müssen. Sie können bei der Einführung agiler Entwicklung auch direkt auf kundenwertoptimierende Teams abzielen und die für selbstorganisierte und liefernde Teams notwendigen Veränderungen gleichzeitig vollziehen.

1.2 Agile Fluency

Die beschriebenen Unterscheidungen haben Diana Larsen und Jim Shore vor einigen Jahren im Agile Fluency Model™ formalisiert. Das Modell verwendet das Erlernen einer Fremdsprache als Metapher. Man kann eine Fremdsprache auf verschiedenen Stufen sprechen. Auf einer Basisstufe kann man vielleicht nach dem Weg fragen und Dinge des täglichen Gebrauchs einkaufen. Auf der nächsten Stufe kann man einfache Gespräche führen. Auf der dritten Stufe kann man anspruchsvolle Literatur verstehen und intellektuell anspruchsvolle Gespräche führen. Auf der vierten Stufe kann man alles Erdenkliche in der Fremdsprache ausdrücken. Fließend (fluent) ist man auf der jeweiligen Stufe, wenn man sie auch in Stresssituationen beibehält.

Dieser Ansatz wird mit dem Agile Fluency Model™ auf Agilität übertragen – es wird in der aktuellen Version allerdings nicht mehr von Stufen, sondern von Zonen gesprochen. Es werden die vier Zonen aus Abbildung 1-7 unterschieden.

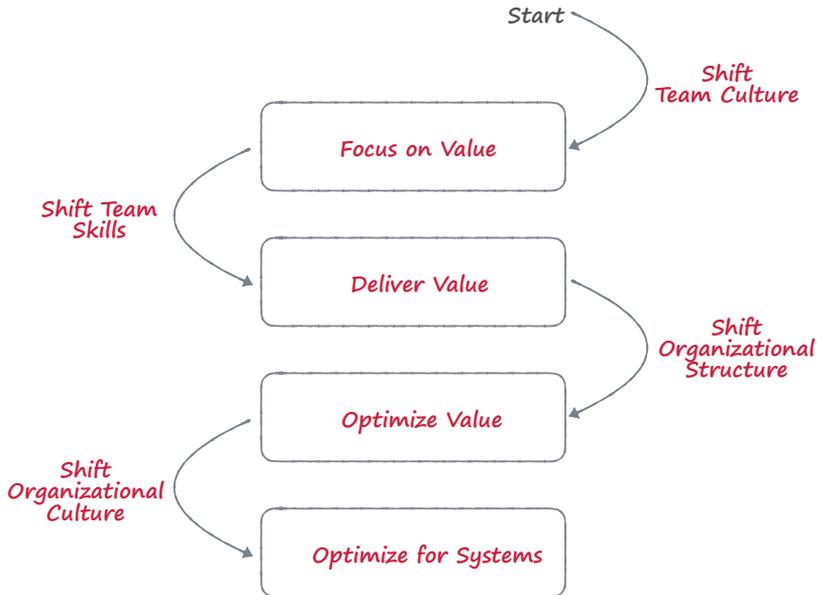


Abb. 1-7 Das Agile Fluency Model™

Das Modell geht davon aus, dass die Beteiligten in der Lage sind, Programmcode für Software zu schreiben (*Start*). Sie beherrschen also ihr grundsätzliches Handwerkszeug, um etwas herzustellen oder eine Dienstleistung zu erbringen. Für das Modell, wie wir es benutzen, ist die Art des Produktes nicht wichtig.

In der ersten Zone »Focus on Value« wird auf Wert fokussiert. Aus Geschäftssicht muss regelmäßig Fortschritt erkennbar sein und es muss die Möglichkeit geben, die Richtung zu ändern. Dazu ist in erster Linie ein Kulturwandel im Team notwendig – weg von einer technisch bestimmten wasserfallartigen Planung hin zu einer Planung aus Geschäftssicht. Regelmäßige Demonstrationen von Produktinkrementen schafft Transparenz über den Fortschritt. Regelmäßige Planungen erlauben das Umsteuern, wenn gewünscht. Ein einfaches iteratives Verfahren mit Planung und Review kann ausreichen, um »Focus on Value« zu erreichen. Diese Zone haben wir oben »selbstorganisiertes Team« genannt (siehe Abb. 1-4).

In der folgenden Zone wird auf die Lieferung von Wert fokussiert. Das Team muss in der Lage sein, so häufig an Kunden auszuliefern, wie es aus Geschäftssicht sinnvoll ist. Um diese Zone zu erreichen, müssen geeignete Entwicklungspraktiken wie Continuous Integration, automatisierte Unit Tests, testgetriebene Entwicklung, Pair Programming und Continuous Delivery installiert werden. Diese Zone haben wir oben »lieferndes Team« genannt (siehe Abb. 1-5).

In der dritten Zone wird auf die Optimierung von Wert durch das Team fokussiert. Hier geht es um die Frage, welche Produkte und Produkteigenschaften wirklich wertvoll für Kunden sind. Es müssen exzellente Produktentscheidungen gefällt werden. Hierzu ist sowohl eine Veränderung der Unternehmensstruktur

als auch Fachexpertise im Team notwendig. Das kann durch die Integration von Business-Analysten oder sogar direkt von Kunden ins Team erfolgen.

In der vierten Zone »Optimize for Systems« werden schließlich Gesamtsysteme optimiert. Dazu werden Gesamt-Wertschöpfungsketten optimiert und systematisch marktrelevante Innovationen erzeugt. Um diese Zone zu erreichen, ist ein Wandel der Unternehmenskultur notwendig. So muss es z.B. möglich sein, Dinge auszuprobieren, die vermeintlich nichts mit dem Geschäftszweck zu tun haben. Es muss auch erlaubt sein, Fehler zu machen. Und nicht zuletzt muss global optimiert werden.

Wichtig ist bei allen Zonen der Fluency-Aspekt. Ein Team ist in einer Zone *fluent*, wenn es auch in Stresssituationen innerhalb dieser Zone bleibt. Ein Team ist also dann in der Zone »Deliver Value« *fluent*, wenn es auch bei anspruchsvollen Terminen und großem Druck durch das Management die agilen Entwicklungspraktiken wie testgetriebene Entwicklung einsetzt.

Das Agile Fluency Model™ ist nicht als Reifegradmodell gedacht. Zum einen werden die Zonen nicht vollständig sequenziell erreicht. Es kann also durchaus sein, dass man die Praktiken für »Deliver Value« einführt, bevor man in »Focus on Value« *fluent* ist. Zum anderen muss die Reihenfolge auch nicht in allen Fällen wie beschrieben ablaufen. Die dargestellte Reihenfolge deckt sich allerdings mit dem, was viele Agile Coaches in ihrer täglichen Arbeit beachten.

1.3 Fokus dieses Buches: echte Agilität oder auch »Optimize Value«

Dieses Buch fokussiert auf echter Agilität alias »Optimize Value« nach dem Agile Fluency Model™ – aber nicht beschränkt auf einzelne Projekte, sondern als relevanter Aspekt der Unternehmensorganisation. Das Buch erläutert nicht, wie »Focus on Value« oder »Deliver Value« erreicht werden, sondern geht davon aus, dass die dafür notwendigen Praktiken bereits erfolgreich eingeführt wurden.

Teams, die »Optimize Value« erreicht haben, liefern größeren Wert an Kunden und treffen bessere Produktentscheidungen. Diese Verbesserung kann direkt mit geschäftsrelevanten Metriken (z.B. Kundenzufriedenheit und Umsatz) gemessen werden.

Kundenwertoptimierende Teams liefern im Verhältnis zur Investition den größtmöglichen Wert. Sie verstehen, was der Markt wünscht, was das Unternehmen benötigt und wie beide Bedürfnisse befriedigt werden können. In einer Start-up-artigen Umgebung weiß das Team, was es lernen muss und wie es das lernen kann.

Diese Teams nutzen z.B. Lean Startup und Design Thinking sowie weitere Techniken zur Product Discovery. Sie arbeiten mit adaptiver Planung, haben das Produktmanagement ins Team integriert und interagieren direkt und persönlich mit Endkunden.

Sie schaffen Transparenz, indem sie mit konkreten Geschäftsmetriken (z.B. Return on Investment, Kundenzufriedenheit, Umsatzrendite pro Mitarbeiter) ins Unternehmen berichten.

Gegenseitiges Vertrauen zwischen Team und Unternehmen führt zu schnellen und effektiven Aushandlungsprozessen. Das Team ist so breit bzgl. seiner Fähigkeiten aufgestellt, dass Übergaben eliminiert werden und Entscheidungen schnell gefällt werden können.

Kundenwertoptimierende Teams erkennt man daran, dass sie mit Geschäftsmetriken arbeiten, auf Kundenbegeisterung hin optimieren und profitabel sind. Weisen relevante Geschäftsmetriken (z.B. Kundenzufriedenheit) auf Probleme hin, reagieren kundenwertoptimierende Teams sofort darauf und ändern ggf. selbstständig die Richtung. Im Extremfall schlägt das Team vor, das Projekt oder Produkt zu beenden.

Um kundenwertoptimierende Teams zu erhalten, muss das Unternehmen in sie investieren und Business-Experten ins Team integrieren. Diese müssen reguläre Vollzeit-Teammitglieder sein.

Darüber hinaus können diese Teams nur dann erreicht werden, wenn organisatorische Hindernisse bei Werterzeugung und Wertlieferung beseitigt werden. Dazu muss das Management im ganzen Unternehmen kooperativ zusammenarbeiten. In den meisten Fällen müssen die Manager dabei gecoacht werden, wie sie diesen veränderten Herausforderungen gerecht werden können.

1.3.1 Eigenschaften von »Optimize Value«-Unternehmen

Damit dies alles möglich wird, müssen Unternehmen in der Regel die folgenden Eigenschaften herausbilden:

- Jeder Mitarbeiter sieht die Wertschöpfung des Unternehmens durch die Augen des Kunden.
- Jeder Mitarbeiter versteht, wie er zu dieser Wertschöpfung beiträgt.
- Jeder Mitarbeiter engagiert sich in der kontinuierlichen Verbesserung der Wertschöpfung.
- Jedes Teammitglied versteht sich als Bestandteil eines Teams mit gegenseitigen Abhängigkeiten.
- Jedes Teammitglied bringt sich so in das Team ein, wie es gerade notwendig ist – auch außerhalb der eigenen Spezialisierung und Komfortzone. Titel und Positionen treten in den Hintergrund.
- Karrierepfade im Unternehmen orientieren sich nicht mehr an fixen Positionen. Stattdessen tritt der Beitrag zum großen Ganzen in den Vordergrund. Für Gehaltserhöhungen spielt daher das Ansehen der Kollegen eine große Rolle. Sie wissen durch die Arbeit im Team am besten um den Beitrag ihres Kollegen.