



Inge Hanschke

Agile in der Unternehmenspraxis

Fallstricke erkennen und vermeiden,
Potenziale heben

EBOOK INSIDE

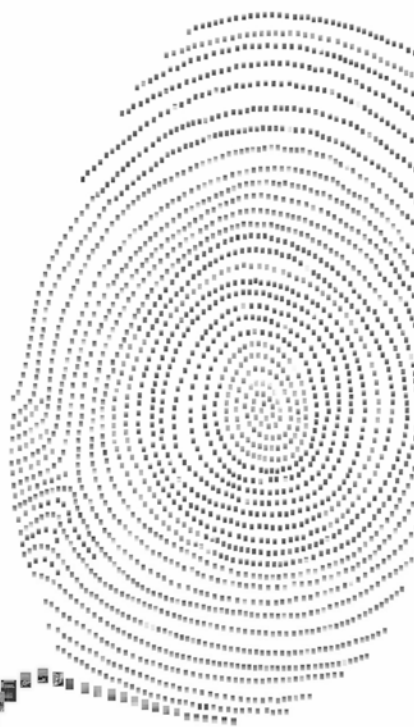
 **Springer** Vieweg

Agile in der Unternehmenspraxis

Lizenz zum Wissen.




Sichern Sie sich umfassendes Technikwissen mit Sofortzugriff auf tausende Fachbücher und Fachzeitschriften aus den Bereichen: Automobiltechnik, Maschinenbau, Energie + Umwelt, E-Technik, Informatik + IT und Bauwesen.

Exklusiv für Leser von Springer-Fachbüchern: Testen Sie Springer für Professionals 30 Tage unverbindlich. Nutzen Sie dazu im Bestellverlauf Ihren persönlichen Aktionscode **C0005406** auf www.springerprofessional.de/buchaktion/



**Jetzt
30 Tage
testen!**

Springer für Professionals.
Digitale Fachbibliothek. Themen-Scout. Knowledge-Manager.

-  Zugriff auf tausende von Fachbüchern und Fachzeitschriften
-  Selektion, Komprimierung und Verknüpfung relevanter Themen durch Fachredaktionen
-  Tools zur persönlichen Wissensorganisation und Vernetzung

www.entschieden-intelligenter.de

Springer für Professionals

 Springer

Inge Hanschke

Agile in der Unternehmenspraxis

Fallstricke erkennen und vermeiden,
Potenziale heben



Springer Vieweg

Inge Hanschke
Lean42 GmbH
München, Deutschland

ISBN 978-3-658-19157-3 ISBN 978-3-658-19158-0 (eBook)
<https://doi.org/10.1007/978-3-658-19158-0>

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

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH 2017

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist Teil von Springer Nature

Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Inhaltsverzeichnis

- 1 Einleitung** 1
 - 1.1 Was ist „Agile“? 1
 - 1.2 Inhalt und Abgrenzung 3
 - Literatur 4
- 2 Agile im Überblick** 7
 - 2.1 Agile Grundwerte 7
 - 2.2 Agile Prinzipien 8
 - 2.3 Agile Methoden 9
 - 2.3.1 Scrum 9
 - 2.3.1.1 Was macht Scrum aus? 16
 - 2.3.1.2 Typische Fallstricke von Scrum 18
 - 2.3.2 Kanban 22
 - 2.3.2.1 Was macht Kanban aus? 25
 - 2.3.2.2 Typische Fallstricke von Kanban 27
 - 2.3.2.3 Unterschiede zwischen Scrum und Kanban 28
 - 2.4 Agile Techniken 30
 - 2.4.1 Management-Techniken 31
 - 2.4.1.1 Daily-Scrum 31
 - 2.4.1.2 Burn-down-Chart 32
 - 2.4.1.3 Definition-of-Done 33
 - 2.4.1.4 Sprint-Review 34
 - 2.4.1.5 Retrospektive 35
 - 2.4.1.6 Führungsphilosophie „Servant Leadership“ 36
 - 2.4.2 Planungstechniken 37
 - 2.4.2.1 Planning Poker 38
 - 2.4.2.2 Planungs-Kick-off-Meeting 39
 - 2.4.2.3 Sprint-Planung 39
 - 2.4.2.4 Minimal Viable Product (MVP) 40
 - 2.4.2.5 User-Stories und Akzeptanztests 42
 - 2.4.2.6 Backlog-Grooming 46

2.4.3	Softwareauslieferungsprozess	47
2.4.3.1	Continuous Delivery	47
2.4.3.2	DevOps	49
	Literatur	52
3	Produkt-Owner	53
3.1	Rolle Produkt-Owner	56
3.1.1	Organisatorische Zuordnung des Produkt-Owners	57
3.1.2	Kompetenzprofil eines Produkt-Owners	59
3.1.2.1	Business-Analyse Fähigkeiten (siehe [1])	61
3.2	Systematisches Management der Geschäftsanforderungen	63
3.2.1	Granularitäten von Anforderungen (siehe [1])	66
3.2.2	Scaled Agile Framework (SAFe®)	70
3.2.3	Techniken für das systematische Management der Anforderungen	72
3.2.3.1	Techniken zum Zerlegen von Anforderungen	73
3.2.3.2	Backlog-Management	75
	Literatur	78
4	Agile Planung	79
4.1	Planung im Überblick	79
4.2	Planung im agilen Umfeld	82
4.3	Agiler Festpreis	94
	Literatur	99
5	Agile Kultur	101
5.1	Wie ist Ihre agile Kultur?	102
5.1.1	Agile in hierarchischen Organisationen	104
5.1.2	Soll-Vision Holacracy?	107
5.2	Verankern von agilen Methoden in der Organisation	110
5.3	Lean als Erfolgsvoraussetzung	116
	Literatur	121
	Glossar	123
	Literatur	127
	Stichwortverzeichnis	129

Wenn es nicht nötig ist, etwas zu ändern, dann ist es nötig, es nicht zu ändern.

Alter Spruch (Quelle: aphorismen.de)

Zusammenfassung

Es ist keine Frage mehr, ob Agile oder nicht. Nicht immer werden aber die gewünschten Erfolge erzielt. Dies liegt häufig nicht am Verständnis über die agilen Methoden wie Scrum (Röpstorff S, Wiechmann R (2015) Scrum in der Praxis: Erfahrungen, Problemfelder und Erfolgsfaktoren, 2. Aufl. dpunkt.verlag GmbH, Heidelberg), sondern an der Strategie, dem Zusammenspiel mit den Planungs-, Entscheidungs- oder Durchführungsprozessen sowie insbesondere an der Unternehmensorganisation und -kultur. Daher liegt in diesem Buch der Schwerpunkt auf einer Sammlung von Tipps & Tricks, um agile Initiativen doch zum Erfolg zu führen, und nicht auf der Beschreibung der einzelnen Methoden. Zu Letzterem gibt es auch bereits zahlreiche Literatur. In diesem Kapitel finden Sie neben Literaturhinweisen einer Einführung zu Agile.

1.1 Was ist „Agile“?

Agiles Vorgehen ist in der Softwareentwicklung inzwischen state-of-the-art. Insbesondere das agile Vorgehensmodell Scrum [1] und dessen Evangelisten haben Agile zum Durchbruch verholfen. Durch überschaubare Liefereinheiten und schnelle Feedbackzyklen wird im Vergleich zu den schwerfälligen, wasserfallartigen Vorgehensmodellen ein Weg aufgezeigt, mit unscharfen Zielen und sich ändernden Anforderungen und Randbedingungen

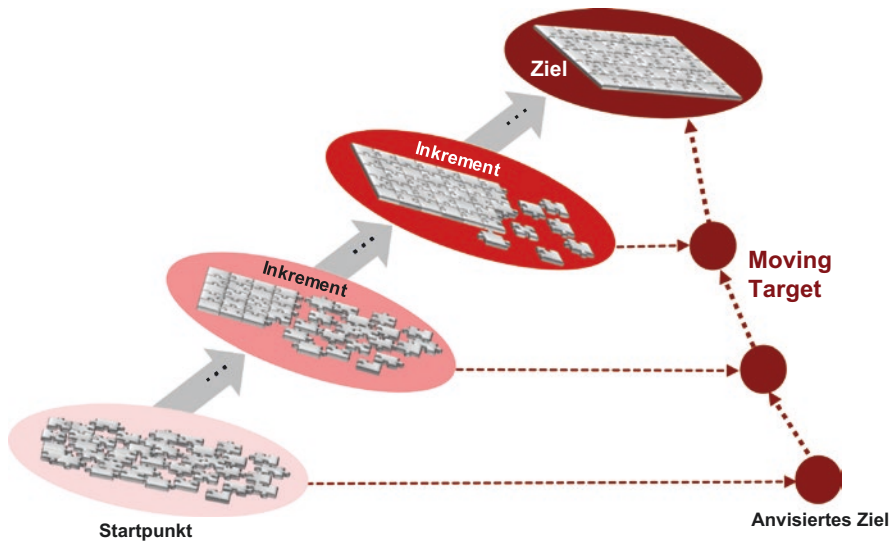


Abb. 1.1 Moving Target

umzugehen. Die Standish Group sieht in ihrem Chaos Report (<https://www.infoq.com/articles/standish-chaos-2015>) den Einsatz agiler Methoden als kritischen Erfolgsfaktor für Softwareprojekte.

- **Agilität beginnt im Kopf** Agilität geht einher mit „Agile Thinking“. Es ist eine Grundhaltung, die verinnerlicht werden muss. Das Verständnis für agile Werte reift durch Erfahrung und Selbstreflektion.

Agile beginnt im Kopf. Enge Kommunikation mit allen Beteiligten, Transparenz über Visualisierung und Ergebnisorientierung sowie Trennung von alten Zöpfen und Formalien, wie z. B. unproduktive Meeting-Marathons oder regelmäßige Berichterstattung ohne klaren Nutzen.

Agile geht davon aus, dass Veränderungen nicht die Ausnahme, sondern die Regel darstellen. Die sich über die Zeit verändernden Ziele und Anforderungen werden durch das „**Moving Target**“ (siehe Abb. 1.1) gut visualisiert. Am Anfang eines Projektes sind die Ziele, Anforderungen und Randbedingungen oft vage oder nicht bekannt. Der Kunde weiß häufig nicht genau, wie das Ergebnis im Detail aussehen soll. Die wirklichen Ziele und Anforderungen müssen durch Business-Analyse erst ermittelt und hinterfragt werden. Konkret „anfassbare“ Ergebnisse, wie z. B. GUI-Prototypen oder Software-Releases, schaffen eine Grundlage für die Diskussion und Festlegung der Anforderungen. Dies sind einzelne Puzzlesteine vom Gesamtbild. Iteration für Iteration entstehen Inkremente, die das Puzzle schrittweise lösen.

Ziele, Anforderungen und Randbedingungen ändern sich über die Zeit. Neue Ziele, Anforderungen oder Randbedingungen kommen hinzu, Anforderungen werden verstanden und Lösungen werden greifbar.

Tab. 1.1 Agile Methoden und Literatur-Beispiele

Adaptive Software Development (u. a. Jim Highsmith)	[2]
Chrystal Methoden (Alistair Cockburn)	[3]
Dynamic System Development Method (Arie van Bennekum)	[4]
eXtreme Programming (u. a. Kent Beck)	[5]
Feature Driven Development (Jon Kern)	[6]
Pragmatic Programming (Andrew Hunt, Dave Thomas)	[7]
Clean Code und Refactoring	[8, 9]
Scrum (Mike Beedle, Ken Schwaber, Jeff Sufferland)	[1, 10–13]
Kanban (David Anderson)	[14]

Es gibt noch eine ganze Reihe weiterer aber relativ unbekannter Ansätze wie DAD (Disciplined Agile Delivery) oder AUP (Agile Unified Process). Diese sind wenig verbreitet.

Hier setzen agile Methoden an. Literatur hierfür gibt es zahlreich. So sind agile Vorgehensmodelle wie Scrum oder Kanban sehr gut beschrieben. Beispiele für Literatur in den verschiedenen Ansätzen zeigt Tab. 1.1.

Die aufgeführten agilen Methoden und Techniken werden im Folgenden nicht ausführlich beschrieben. Hier sei auf die angegebene Literatur verwiesen.

Agilität ist aber nicht nur ein Thema von Softwareentwicklungsprojekten, sondern vom ganzen Unternehmen. Nicht nur Anforderungen ändern sich, sondern gerade im Zuge der zunehmenden Digitalisierung verändern sich rasant die Geschäftsmodelle mit veränderten Kundengruppen, verschiedenen Kommunikationskanälen, Ökosystem der Partner, technischen Möglichkeiten, wie z. B. Big Data oder IoT, und die Arbeitswelt an für sich.

Nur durch ein schnelles Einstellen auf Veränderungen und vorausschauendes Agieren bleibt ein Unternehmen auf Dauer wettbewerbsfähig. Gepaart mit der Kompetenz in der Digitalisierung ebenso wie im Demand Management und der Softwareentwicklung sowie anderen Anwendungsbereichen. Die Kompetenz umfasst das Wissen, die Vorstellungskraft und das Urteilsvermögen über weitere Entwicklungen und deren Umsetzung im jeweiligen Kontext.

► **Agilität** Agilität ist die Fähigkeit sich auf alle Arten von Veränderungen einzustellen und Ziele, Inhalte, Organisation und Prozesse zeitgerecht anzupassen.

1.2 Inhalt und Abgrenzung

Schwerpunkt des Buchs sind prägnante Tipps & Tricks, da bei der Einführung von agilen Methoden und Techniken nicht immer die gewünschten Erfolge erzielt werden. Hierfür gibt es die unterschiedlichsten Gründe, z. B. organisatorische Randbedingungen wie ein traditionelles, hierarchisch geprägtes Unternehmen oder aber eine halbherzige Einführung. Dieses Buch widmet sich gerade diesen Fallstricken und gibt Hilfestellungen, diese zu umgehen. Agile wird hierbei nicht nur für Softwareprojekte, sondern auch für andere

Disziplinen betrachtet. Agile ist ein Erfolgsfaktor für alle Situationen, wo es gilt, sich schnell auf Veränderungen einzustellen. Dies ist z.B. gerade in der Digitalisierung oder Industrie 4.0 entscheidend.

Die verbreiteten agilen Methoden und Techniken für Softwareprojekte, wie z. B. Scrum und Kanban sowie DevOps werden im folgenden Kapitel im Überblick vorgestellt und typische Fallstricke aufgezeigt.

In den darauffolgenden Kapiteln finden Sie Tipps & Tricks für agile Fragestellungen in und außerhalb von Softwareprojekten. Sie bekommen unmittelbar anwendbare Leitfäden und Empfehlungen an die Hand. Folgende Schwerpunkte werden dabei gesetzt:

- Einbezug des Kunden im **agilen Demand Management**, um die wirklichen und priorisierten Anforderungen in der richtigen Granularität zeitgerecht zu identifizieren und angemessen zu beschreiben. Wesentlich ist hierbei insbesondere die Rolle des Produkt-Owners. Siehe hierzu Abschn. 3.1.
- **Agile in einem eher traditionellen, hierarchisch geprägten Unternehmen** erfolgreich einführen und trotz Widerständen zum Leben bringen und am Leben erhalten. Siehe hierzu Abschn. 5.1.1.
- **Agile Planung** im Vorfeld von und innerhalb von Projekten, um entsprechend den Erfordernissen des Planungshorizontes in einer passenden Granularität zeitgerecht eine belastbare („good enough“) Aufwandsschätzung zu erstellen. Siehe hierzu Kap. 4.
- **Lean** als Erfolgsvoraussetzung, um den wirklichen Kundenwert zu identifizieren und schrittweise zu heben. Siehe hierzu Abschn. 5.3.

Nun schauen wir uns die agilen Grundwerte, Methoden und Techniken etwas näher an.

Literatur

1. Röpstorff S, Wiechmann R (2015) Scrum in der Praxis: Erfahrungen, Problemfelder und Erfolgsfaktoren, 2. Aufl. dpunkt verlag GmbH, Heidelberg
2. Highsmith JA (1999) Adaptive software development: a collaborative approach to managing complex systems, 1. Aufl. Dorset House Publishing Co Inc., New York
3. Cockburn A (2004) Crystal clear: a human-powered methodology for small teams: a human-powered methodology for small teams (agile software development), 1. Aufl. Addison-Wesley Professional, Harlow
4. Stapleton J, Constable P (1997) DSDM: Dynamic systems development method: the method in practice by Jennifer Stapleton, 1. Aufl. Addison-Wesley Professional, Harlow
5. Beck K, Andres C (2004) Extreme programming explained: embrace change: embracing change, 2. Aufl. Addison-Wesley Professional, Reading
6. Palmer SR (2002) A practical guide to feature-driven development, 1. Aufl. Prentice Hall, Upper Saddle River
7. Hunt A, Thomas D (1999) The pragmatic programmer: from journeyman to master, 1. Aufl. Addison-Wesley Professional, Reading

8. Martin RC (2008) Clean code: a handbook of agile software craftsmanship, 1. Aufl. Prentice Hall, Upper Saddle River
9. Fowler M, Beck K, Brant J, Opdyke W, Roberts D, Gamma E (1999) Refactoring: improving the design of existing code, 1. Aufl. Addison-Wesley, Reading
10. Sutherland J (2014) Scrum: a revolutionary approach to building teams, beating deadlines and boosting productivity, 1. Aufl. Cornerstone Digital, London
11. Larman C, Vodde B (2016) Large-Scale scrum: more with LeSS, 1. Aufl. Addison-Wesley Professional, Boston
12. Leffingwell D (2016) Scaled agile framework reference guide: scaled agile framework® for lean software and systems engineering, 1. Aufl. Addison-Wesley, Boston
13. Glogger B, Margetich J (2014) Das Scrum-Prinzip – Agile Organisationen aufbauen und gestalten. Schäffer-Poeschel Verlag, Stuttgart
14. Skarin M (2015) Real-world kanban: do less, accomplish more with lean thinking, 1. Aufl. O'Reilly UK Ltd, Frisco

Zusammenfassung

Ausgehend vom **Agilen Manifest** haben sich agile Methoden in Softwareprojekten durchgesetzt. Im Folgenden finden Sie einen Überblick über agile Grundwerte und Prinzipien sowie die wesentlichen Methoden und Techniken, wie Scrum, Kanban und DevOps mit ihren Fallstricken.

2.1 Agile Grundwerte

Die agilen Werte Selbstverpflichtung, Mut, Fokus, Offenheit und Respekt zusammen mit dem agilen Manifest von 2001 bilden die Eckpfeiler für das agile Vorgehen. Zu den Unterzeichnern zählen neben den Scrum-Größen wie Mike Beedle, Ken Schwaber und Jeff Sutherland andere wichtige Wegbereiter wie Kent Beck, Ron Jeffries, Ward Cunningham und James Grenning (XP¹), Alistair Cockburn (Crystal), Arie van Bennekum (DSDM²), Martin Fowler (Clean Code und Refactoring), Jon Kern (FDD³), Brian Marick (Agile Testing), Dave Thomas und Andrew Hunt (Pragmatic Programmer), Jim Highsmith (Adaptive Software Development), Robert C. Martin und Dave Thomas (Ruby on Rails) sowie Steve Mellor.

¹ XP – eXtreme Programming.

² DSDM – Dynamic Systems Development Method.

³ FDD – Feature Driven Development.

► **Manifest für Agile Softwareentwicklung** Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

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

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein. Siehe www.agilemanifesto.org.

Das Agile Manifest gibt eine Orientierung vor, die durch agile Prinzipien sowie agile Methoden und Techniken operationalisiert werden. Diese schauen wir uns im Folgenden etwas näher an.

2.2 Agile Prinzipien

Agile Prinzipien setzen Leitplanken für die tägliche Praxis. Im agilen Manifest werden zwölf Prinzipien aufgelistet (siehe www.agilemanifesto.org/principles.html), die ein Gefühl für „Agile Thinking“ geben.

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essentiell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.