



klaus WYBRANIETZ

# SAP

## The Agile Way

Praxisbewährte Tipps  
für die erfolgreiche agile Arbeit  
mit weltweit verteilten SAP-Teams

HANSER

HANSER

Klaus Wybranietz

## **SAP, The Agile Way**

Praxisbewährte Tipps für die  
erfolgreiche agile Arbeit mit  
weltweit verteilten SAP-Teams

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb 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.

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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2021 Carl Hanser Verlag, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Brigitte Bauer-Schiewek

Copy editing: Dolores Omann, Ternitz

Korrektur: Petra Kienle, Fürstenfeldbruck

Layout: Manuela Treindl, Fürth

Umschlagdesign: Marc Müller-Bremer, München, [www.rebranding.de](http://www.rebranding.de)

Umschlagrealisation: Max Kostopoulos

Titelmotiv: © [shutterstock.com/Wth](https://www.shutterstock.com/Wth)

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Print-ISBN: 978-3-446-46567-1

E-Book-ISBN: 978-3-446-46586-2

E-Pub-ISBN: 978-3-446-46706-4

# Inhalt

## Titelei

## Impressum

## Inhalt

## Vorwort

## Der Autor

# 1 SAP und Scrum – das geht doch nicht?

## 1.1 Sieben Vorurteile gegen Scrum

## 1.2 Was SAP so speziell macht

# 2 Die Grundlagen von Scrum

## **2.1 Was Scrum wirksam macht**

## **2.2 Rollen, Meetings und Artefakte**

### **2.2.1 Die Rollen in Scrum**

### **2.2.2 Die Scrum-Artefakte**

### **2.2.3 Die Scrum-Ereignisse oder Scrum-Events**

## **2.3 Planen und Schätzen in Scrum**

### **2.3.1 Die Planungsspannen in Scrum**

### **2.3.2 Schätzen in Scrum**

#### **2.3.2.1 Was sind Story Points?**

#### **2.3.2.2 Schätzen mit Story Points**

### **2.3.3 Agiles Schätzen mit Hilfe von Magic Estimation**

## **3 Mit verteilten SAP-Teams remote arbeiten**

### **3.1 Grundlagen der Kommunikation oder 55-38-7**

### **3.2 Voraussetzungen für erfolgreiche Online-Events**

### **3.3 Tools für erfolgreiche agile Remote-Arbeit**

3.3.1 Arbeiten mit Konferenz-Tools

3.3.2 Arbeiten mit Collaboration-Tools

### **3.4 Effizient remote arbeiten**

3.4.1 Deep-Work-Zeiten

3.4.2 Working Agreement

### **3.5 Agile Online-Reifegrade verteilter Teams**

## **4 Mit agilen SAP-Teams starten**

4.1 Wir brauchen eine Vision

4.2 Wir brauchen ein Team

4.3 Wir brauchen Regeln

4.3.1 Regeln managen mit dem Delegation Board

4.3.2 Die Definition of Ready (DoR)

4.3.3 Die Definition of Done (DoD)

4.3.4 Die Sache mit der Dokumentation

**4.4 Arbeiten mit dem Minimum Viable Product**

## **4.5 Komplexe Anforderungen zerlegen mit User Story Mapping**

## **4.6 Das Product Backlog Refinement als Motor der Produktivität**

4.6.1 Warum das Product Backlog Refinement so wichtig ist

4.6.2 Vorbereitung des Product Backlog Refinement

4.6.3 Effektive Gestaltung des Product Backlog Refinement mit „Focus – Spread – Focus“

4.6.4 Nachbereitung des Product Backlog Refinement

## **4.7 Magic Estimation für SAP-Teams**

## **4.8 Den Wissenstransfer im Team unterstützen**

4.8.1 Training Storys

4.8.2 Team-Skill-Matrix

4.8.3 Skillaufbau mit Hilfe von Pufferzeiten

## **4.9 Fachliche Qualität sichern**

4.9.1 C Collective Code Ownership light

4.9.2 SAP Test-Automation



# **5 Skalierung mit dem Agile Working Model 4 SAP**

## **5.1 Grundsätzliche Überlegungen zur Skalierung im SAP-Umfeld**

## **5.2 Praxisbeispiel: Skalierung für eine S/4HANA-Implementierung**

**5.2.1 3-Ebenen-Modell für die Abbildung agiler Prozesse**

**5.2.2 Zusätzliche Rollen für die Skalierung**

**5.2.3 Neue Anforderungsformate: Sagas und Epics**

**5.2.4 Die Zyklen im Agile Working Model 4 SAP**

**5.2.5 Agile Events im Domain Cycle**

**5.2.6 Agile Events im Product Cycle**

## **5.3 Unterstützung durch ein Executive Action Team**

# **6 Praktiken für fortgeschrittene agile SAP-Teams**

## **6.1 Was ist Kanban?**

**6.1.1 Prinzipien und Praktiken von Kanban**

[6.1.2 Serviceklassen in Kanban](#)

## [6.2 Kanban in der Praxis](#)

[6.2.1 Umgang mit WIP-Limits](#)

[6.2.2 Kanban in Scrum-Teams](#)

## [6.3 Value Cycle Loop \(VCL\)](#)

[6.3.1 Die Werte von User Storys](#)

[6.3.2 Der Ablauf der Bewertung mit dem Value Cycle Loop](#)

## [6.4 Flow-Metriken](#)

[6.4.1 Das kumulative Flussdiagramm \(Cumulative Flow Diagram\)](#)

[6.4.2 WIP-Analyse](#)

[6.4.3 Durchlaufzeit und Durchsatz](#)

## [Danksagungen](#)

## [Literatur](#)

# Vorwort

In den vergangenen Jahren habe ich von vielen Führungskräften auf der ganzen Welt gehört, wie toll ihre agilen Entwicklungsteams zum Beispiel im Java-Umfeld arbeiten. Ich habe gehört, wie glücklich und effektiv Softwareentwickler sind, die mit Scrum arbeiten, wie viel Power und positive Energie in diesen Teams entsteht. Und: Mir wurde auch erzählt, wie viele Bewerbungen von Top-Spezialisten diese Unternehmen bekommen, nachdem sie sich zum agilen Arbeiten bekannt oder sogar eine agile Transformation angestoßen haben.

Wenn ich mit Personen aus dem SAP-Umfeld spreche, höre ich genau das Gegenteil. Ich verstehe SAP-Projekte ganz einfach immer so: Der Kunde kauft eine Standard-Software, die nur angepasst werden muss und danach genutzt werden kann. Es wird also keine neue Software entwickelt, sondern gekaufte Software adaptiert. Das ist der zentrale Unterschied zu allen anderen individuellen Entwicklungsprojekten dieser Welt.

Trotzdem habe ich zu viele SAP-Projekte gesehen, die agil gescheitert sind, und ich habe zu oft gehört, dass SAP und Scrum einfach nicht kompatibel seien. Meistens lag und liegt die Ursache des Scheiterns darin, dass die Anforderungen nicht konsequent und kontinuierlich genug priorisiert werden. Ganz im

Sinne des Wasserfalls werden umfangreiche Anforderungen definiert, verbunden mit der Erwartung, dass sie auch zu 100 Prozent umgesetzt werden. Das hat nichts mit agilem Arbeiten zu tun und ist zum Scheitern verurteilt. Dies hat einige Projektverantwortliche dennoch nicht daran gehindert, ihren Wasserfall-Projekten eine agile Masche umzubinden, indem sie einfach die Events „agil“ benannt haben. Die gängigsten Ausreden, die hartnäckigsten „Ja, aber ...“, finden Sie gleich in [Kapitel 1](#). Der Grundtenor lautet: SAP ist angeblich viel zu komplex und viel zu integriert, als dass sich da agil was machen ließe.

Das mag in der Welt dieser SAP-Profis so sein. Doch auch an ihnen geht die Agilisierung der Arbeitswelt nicht spurlos vorbei, und wenn sie noch nicht mitten in einer agilen Transformation stecken, stehen die Chancen gut, dass sie in den kommenden Jahren davon betroffen sein werden. Spätestens dann werden sie – sehr schnell – lernen müssen, mit agilen Organisationseinheiten und vielleicht sogar mit ihren SAP-Teams agil zu arbeiten.

Ein solcher Anlassfall könnte auch S/4HANA sein. Mit „SAP Activate“ forciert SAP die schnelle, agile Einführung anhand vorgedachter Konfigurationen (Guided Configurations). Aus meiner Sicht unterstützt SAP Activate alle Ausgangspunkte für den agilen Projektstart, von der Neuimplementierung von S/4HANA über die Systemkonvertierung bis hin zur Landschaftstransformation. Beim empfohlenen Framework setzt SAP auf Scrum und Scrum@Scale – beides werde ich in diesem Buch besprechen.

Ich brauche aber nicht erst S/4HANA, um zu wissen, dass SAP und Agile gut zusammenpassen. Der Erfolg bei internationalen

Großkonzernen, wo wir nicht nur agil, sondern mit weltweit verteilten Teams agil arbeiten, gibt mir in dieser Hinsicht recht. Für mich war es deshalb an der Zeit, die „guten agilen Praktiken“, die ich in den letzten 14 Jahren (von 25 im SAP-Umfeld insgesamt) mit vielen Teams erarbeitet habe, komprimiert weiterzugeben. Mir geht es immer darum, die Dinge möglichst einfach zu machen. Daher werden Sie in diesem Buch keine philosophischen Vertiefungen in das Thema Agilität finden, sondern praktisch Erprobtes und ständig Verbessertes. Für sämtliche Aspekte von Scrum, Kanban und Agile im Allgemeinen gibt es inzwischen eine breite Palette an anderen Publikationen, die ich Ihnen ans Herz legen möchte, damit Sie das agile Arbeiten von Grund auf verstehen. Ich halte mich in meiner agilen SAP-Praxis sehr eng an den Scrum Guide von Jeff Sutherland und Ken Schwaber, alle Fragen der Skalierung haben den Guide zu Scrum@Scale als Basis.

Ein paar Dinge müssen für das SAP-Umfeld aber angepasst werden, um erfolgreich agil arbeiten zu können – diese Praktiken beschreibe ich in diesem Buch.

*Klaus Wybranietz*

München, September 2020

# Der Autor



**Klaus Wybranietz** war zwölf Jahre als SAP-Berater auf der ganzen Welt unterwegs, bevor er 2001 die antagon AG gründete. Nach der Fusion der antagon AG mit der EXXETA GmbH im Jahr 2006 wurde das SAP-Team rund um Klaus Wybranietz aufgrund der herausragenden Erfolge in agilen Projekten mit höchstem technischen Anspruch zu einem SAP Service Partner. Zu den Spezialgebieten des SAP ERP-Experten zählten Verkauf, Absatzplanungsprozesse, Verkaufssteuern und Sales Taxes (USA) sowie Rechnungslegung mit Sales Reporting.

Seit 2012 widmet sich Klaus Wybranietz ausschließlich der Verbindung von Agilität und SAP-Projekten. Als Agile Coach und Trainer begleitet er Organisationen und Führungskräfte in agilen Transformationen mithilfe von Scrum, Scrum@Scale, LeSS, Kanban, Design Thinking und den Management-3.0-Praktiken nach Jurgen Appelo. Als einer der wenigen auf SAP-Projekte spezialisierten Agile Coaches und Projektleiter arbeitet Klaus Wybranietz regelmäßig mit Teams, die über Europa, die USA und Südafrika verteilt sind.

# 1 SAP und Scrum – das geht doch nicht?

## 1.1 Sieben Vorurteile gegen Scrum

Wenn ich mich mit SAP-Projektleitern und anderen SAP-Profis über agiles Arbeiten unterhalte, geraten die Gespräche oft an folgenden Punkt: „Ja, aber ...“ Im SAP-Umfeld im Speziellen oder im ERP-Umfeld im Generellen könne man doch nicht vernünftig mit Scrum arbeiten. SAP sei viel zu kompliziert, viel zu komplex, viel zu integriert. Es gab und gibt Tausende Ausreden.

Allerdings gehen auch an diesen SAP-Profis und ihren Teams die agilen Transformationen nicht spurlos vorbei, denen sich Unternehmen derzeit unterziehen. Ganze IT-Einheiten, Abteilungen und Bereiche werden umgestellt und es bleibt den SAP-Spezialisten nichts anderes übrig, als mit anderen unter diesen neuen Umständen zusammenzuarbeiten. Was ich in den Gesprächen immer wieder heraushöre, sind die Beispiele aus der „alten“ Welt, die als Beweis dafür angeführt werden, dass Agilität und SAP nicht zusammenpassen: Die Rede ist von Anforderungen, Fach- oder IT-Konzepten, in 300-seitigen Word-



Dokumenten verpackt. Üblicherweise bekommt der Kunde erst nach eineinhalb Jahren etwas live zu sehen und zu testen und es dauert weitere sechs Monate, bis er die gesamte Software nutzen kann – mit all den Anforderungen, die erst später dazugekommen sind.

Der Ablauf von SAP-Projekten sah jahrzehntelang folgendermaßen aus:

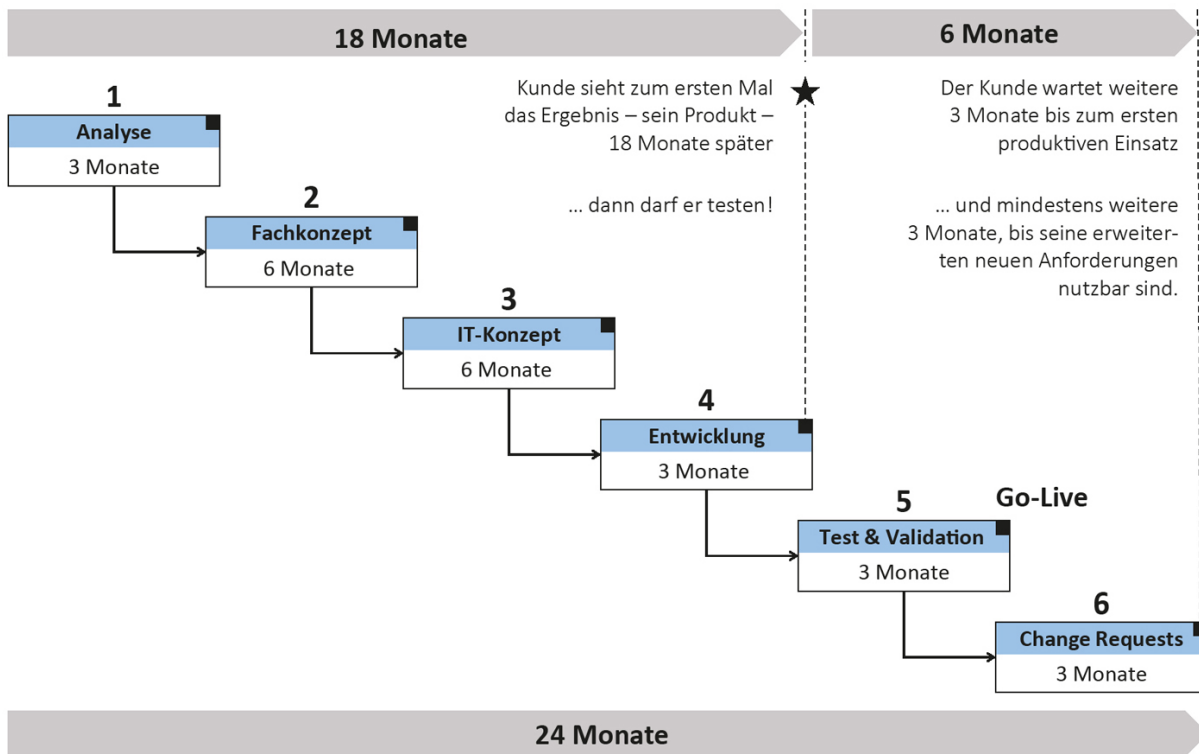
- *Analysephase – drei Monate:* Alle möglichen Add-on-Anbieter wurden analysiert und bewertet sowie erste Ideen zu Prozessen und dem Umfang der Anforderungen gesammelt.
- *Fachkonzept – drei Monate:* Dann ging es mit den Details los – von den Anforderern getrieben, meist in Word erstellt, in zahlreichen Meetings mühsam gemeinsam erarbeitet. Einer schrieb das Dokument, fünf Kollegen schauten zu und berieten sich. Erst danach wurde das Konzept der IT übergeben. Ich war selbst oft mit dabei. Wenn ich den Kunden gut genug kannte, habe ich auf den letzten Seiten des Fachkonzepts einen „Snickers-Absatz“ eingefügt: „Gratulation! Wer sich so tief eingelesen hat, verdient eine Belohnung. Bitte kontaktieren Sie mich, ich gebe ein Snickers aus!“ In 25 Jahren Arbeit in der IT hat sich niemand bei mir gemeldet.
- *IT-Konzept – sechs Monate:* Die SAP-Profis brauchten dann ein weiteres halbes Jahr, um ein IT-Konzept zu schreiben. Das war jedes Mal sehr technisch und beschrieb, wie die Anforderungen innerhalb des SAP-Systems umgesetzt werden konnten. Es ging um SAP-Funktionen, SAP-Methoden, das entsprechende Customizing der SAP-Standard-Software, User Exits, Includes, ABAP-Routinen, BADIs und BAPIs, um technische Erweiterungen und ganz oft

um harte Modifikationen, also um Coding-Änderungen an der SAP-Standard-Software.

- *Entwicklung – drei Monate:* Nach einem Jahr konnte es dann endlich mit der Entwicklung losgehen. Drei Monate Entwicklungszeit deuten aber schon auf einen sehr kurzen, sehr modernen Ansatz hin. Gezielt wurde auf Erweiterungen verzichtet und viel mit SAP-Standard gearbeitet. In dieser Entwicklungsphase wurden auch SAP-Grundeinstellungen mit dem sogenannten Customizing vorgenommen.
- *Test und Validierung – drei Monate:* Nach einem ausführlichen Entwicklertest durfte der Kunde die Software endlich zum ersten Mal sehen und ausprobieren. Die SAP-Trainer schulten zuerst das „Testing Team“ – die Anwender – und überlegten sich dann mit dem Team die meist prozessbezogenen Testfälle, die drei Monate auf Herz und Nieren überprüft wurden. Aus drei Monaten konnten aber auch sechs Monate werden, wenn das Projekt nicht nahe genug an der SAP-Standard-Software war und die Anwendung modifiziert wurde.

Zwischen jeder dieser Phasen gab es Abnahme-Meetings, zu denen viele Mitarbeiter erschienen, oft auch hochkarätige Manager. Doch diese hatten keine Zeit, um sich die Dokumente mit mehreren hundert Seiten durchzulesen, und so wurden aus der Abnahme oft fünfstündige Marathonsitzungen. Alle neuen Anforderungen, die sogenannten Change Requests, wurden während dieser Phasen grundsätzlich abgelehnt und nach hinten verschoben, in die „Change-Request-Phase“. Die Devise lautete: „Stick to the plan!“ Da waren keine neuen Anforderungen erlaubt: in der Fachkonzeptphase nur selten, in der Entwicklungs- und Testphase überhaupt nicht.

Ich kann meinen Gesprächspartnern die erste ablehnende Reaktion beim Wort „agil“ also gar nicht verübeln. Wir haben jahrelang in Wasserfällen gearbeitet, das war unser bewährtes Vorgehen. So haben wir gedacht, gelebt, gearbeitet.



**Bild 1.1** Der bisherige Ansatz: Umsetzung von SAP-Projekten gemäß Wasserfall-Denken

Doch heute wissen wir aus diversen Untersuchungen, dass sich in der digitalen Welt jeden Monat mindestens drei Prozent der Anforderungen verändern und überholen. Wenn wir das auf die genannten 18 Monate umlegen, nach denen der Kunde das Ergebnis zum ersten Mal sieht, bedeutet das: 40 Prozent der Anforderungen sind zu diesem Zeitpunkt schon wieder veraltet oder werden nicht mehr gebraucht. Uns SAP-Projektleiter hat das aber nicht interessiert. Was wir hören wollten, war: „in time“ und

„in budget“. Darauf waren wir stolz. Die VUCA-Welt existierte für andere, aber nicht für uns. Dass wir viel zu lange brauchten, dass wir Software und Prozesse bauten, die in Teilen niemand mehr benötigte, und dass der Kunde jahrelang auf das Ergebnis warten musste – das war unser SAP-Alltag. Das war normal und die Kunden waren es nicht anders gewöhnt.

Auch wenn heute viele Projekte mit „Dailys“ und „Planning Events“ einen pseudo-agilen Anstrich bekommen, liegt ihnen im Denken noch immer das Wasserfall-Modell zugrunde. Scrum-Master-Zertifizierungen sagen nichts darüber aus, wie agil die persönliche Haltung tatsächlich ist. Vor solchen Projektleitern und Agile Coaches der eigentlich alten Schule möchte ich ausdrücklich warnen. Sie haben nichts dazugelernt und nichts verstanden von der neuen, agilen Welt. Sie haben sich eine Zertifizierung aus geschäftlichem oder beruflichem Kalkül geholt und trauern doch der alten Welt nach. Ich erlebe, wie Sprints und alle anderen agilen Events in Wasserfall-Projekte eingebunden werden und aus einem agilen Framework mit entsprechend geschnittenen User Storys ein „Fake Scrum“ oder „Fake Agile“ wird. Außer den Bezeichnungen ändert sich dabei nichts.

„In time“ und „in budget“ – das ist noch immer das vorherrschende Denken. Die wichtigste Frage, nämlich jene nach dem Nutzen der Software, spielt keine Rolle. Deshalb werden die schön beschriebenen Funktionen der Standard-Software einfach umgesetzt, ohne den Nutzen zu hinterfragen. Das Ergebnis: 60 Prozent der Software werden angepasst und eingeführt, aber nicht wirklich gebraucht. Das wäre alles nicht so schlimm, aber: Jedes Modul in SAP, das freigeschaltet, angepasst und eingeführt wird, muss gewartet werden – ob wir das wollen oder nicht. Diese Wartungs- und Betriebskosten müssen von irgendwem bezahlt werden. Deswegen blicken viele Unternehmen sorgenvoll auf das

Jahr 2027: Die Wartung der aktuellen SAP-ERP-Software endet dann – sie wird nicht mehr gepflegt und weiterentwickelt. „S/4HANA“ lautet das neue Zauberwort.

Spätestens dann werden die Sünden der vergangenen Jahre offenbar. Bei jedem der sogenannten Migrationspfade werden uns die Modifikationen und Z-Programme – die nichts anderes als Modifikationen sind – um die Ohren fliegen. Wer jetzt noch an der alten Welt festhält, wird entweder am gewählten S/4HANA-Migrationspfad scheitern oder den gleichen Funktionsumfang wie früher aktivieren – ohne zu merken, dass er ihn gar nicht braucht.

Warum das Wort „agil“ so viel Widerstand auslöst? Weil oft einfach der Mut fehlt, etwas wegzulassen, das gar nicht genutzt wird. Stellen wir die Aussagen, in denen sich dieser fehlende Mut widerspiegelt, einfach mal auf den Prüfstand.

## **SAP ist viel zu komplex**

Seit sich das agile Arbeiten in der IT-Szene durchgesetzt hat, höre ich das Argument, SAP sei viel zu komplex, um nach einem zweiwöchigen Sprint dem Kunden etwas wirklich Nutzbares zeigen zu können. „Lass uns das drei bis vier Monate untersuchen, ein IT-Konzept schreiben und dann mit dem Programmieren beginnen. Das dauert noch.“

Dieses Komplexitätsdenken ist mir schon vor zehn Jahren begegnet und es begegnet mir heute noch immer. Damals durften wir das Wort „Scrum“ gar nicht erwähnen und schon gar nicht durften wir offiziell mit Scrum arbeiten. Erst wenn ein Projekt zu scheitern drohte und um Hilfe gerufen wurde, dann

war Scrum erlaubt. Ich bezeichnete Scrum dann eben undercover als „den Weg der kleinen Schritte“.

## Wir entwickeln nicht!

Der Scrum Guide von Jeff Sutherland und Ken Schwaber sowie viele andere Publikationen, Trainings und Videos zum Thema Scrum haben eine entscheidende Schwäche: Sie sprechen immer von *Produktentwicklung*, *Softwareentwicklung* etc. Selbst die großen agilen Erklärer arbeiten immer noch mit den gleichen simplifizierten Beispielen: „Wir entwickeln einen Kundenstamm“, „Wir entwickeln einen Artikelstamm“, „Wir entwickeln einen Benutzerstamm“.

Jedem SAP-Projektleiter und SAP-Berater stellt es dabei die Nackenhaare auf. Aus ihrer – und auch aus meiner Sicht – passen diese Beispiele nicht. Ich finde sie sogar doof. Die SAP-Standard-Software kann das alles schon. Nur äußerst selten werden Funktionen komplett neu entwickelt. Also: „Scrum ist nichts für uns. Es passt nicht.“

Erst im September 2017 stellten Sutherland und Schwaber klar, dass mit „Entwicklung“ nicht nur das Programmieren gemeint ist, sondern alle Arten von komplexer Arbeit. Das liest sich im deutschen Scrum Guide so (Schwaber, Sutherland 2017, S. 4):

*„Scrum wurde ursprünglich dazu entwickelt, um Produkte zu managen und zu entwickeln.*

*Seit den frühen 1990er-Jahren wird Scrum weltweit ausgiebig genutzt zur:*

- *Erforschung und Identifizierung rentabler Märkte, Technologien und Produktfähigkeiten;*

- *Entwicklung von Produkten und Verbesserungen;*
- *Auslieferung von Produkten und Verbesserungen auch vielfach pro Tag;*
- *Entwicklung und Aufrechterhaltung von Cloud-Umgebungen (online, sicher, on-demand) und anderer Produktivumgebungen sowie*
- *Erhaltung und Erneuerung von Produkten.*

*Scrum wird verwendet, um Software, Hardware, Embedded Software, Netzwerke von interagierenden Funktionen und autonome Fahrzeuge zu entwickeln. Scrum wird aber auch in Schulen, Regierungs- und Marketingprojekten genutzt, zur Verwaltung von Organisationen und der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen und als Gesellschaften verwenden. (...)*

*Wenn die Wörter ‚entwickeln‘ und ‚Entwicklung‘ im Scrum Guide verwendet werden, beziehen sie sich auf komplexe Arbeit, wie z. B. die oben beschriebenen Typen.“*

Erst mit dieser Klarstellung wurde das „Ja, aber ...“ von den Erfindern von Scrum endlich schriftlich entkräftet. Im Scrum Guide 2020 findet sich dieser explizite Hinweis nicht mehr. Aber der Guide wurde sprachlich so überarbeitet, dass die Prinzipien von Scrum allgemeingültig auf jegliche Form von komplexer Arbeit anwendbar sind. Im Vorwort steht zu lesen (Scrum Guide 2020, S. 1): *„We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots.“*

## **SAP-Projekte sind zu groß für Scrum**

SAP-Projekte sind mächtig, mit vielen Teams und unterschiedlichen Schwerpunkten, mit einem hohen Grad an notwendigen Integrationen – zum Beispiel zwischen Logistik, Finanzen und/oder Controlling – und meist auch mit komplexen Anforderungen an das Reporting. Und bei all den komplexen Aufgabenstellungen soll vielleicht noch mit mehreren agilen Teams gearbeitet werden?

Der Scrum Guide beschreibt lediglich, wie man mit einem einzigen Team arbeitet. Um auf Modelle für die Skalierung zu stoßen, muss man sich schon etwas intensiver mit dem Thema beschäftigen. Aber es gibt sie, die Skalierungsmodelle, und auch in diesem Buch werde ich beschreiben, wie aus Sagas und Epics doch noch User Storys werden und wie man mit 10, 20 oder 30 Teams agil arbeiten kann.

Gerade in großen und komplexen Projekten kommen die Vorteile von agilen Arbeitsweisen zum Tragen. Eine Anforderung an ein Produkt wird in kleine Einheiten zerlegt. Diese Einzelteile werden priorisiert und reihum fertiggestellt. Das ist einfach schneller und zuverlässiger als das Wasserfallvorgehen.

## **In Scrum gibt es keine Planung**

Dieses Vorurteil, das ich mit vielen Beispielen widerlegen werde, begegnet mir heute noch immer. Schon für ein einziges SAP-Team, das agil arbeitet, gibt es drei Planungshorizonte auf Tages-, Wochen- und Monatsebene. Wenn mehrere Teams miteinander arbeiten, kommen noch weitere Planungshorizonte hinzu. Planung ist der Motor des agilen Arbeitens und wir planen



nicht nur einmal am Beginn eines Projekts, sondern in regelmäßigen Abständen. Die flexible, inkrementelle Planung ist das Prinzip des Sprints, damit wir flexibel auf neue Anforderungen reagieren können.

## **Die SAP-Helden sind nicht teamfähig**

Die SAP-Profis, die Modulspezialisten, die Helden unter den ITlern – sie lassen sich angeblich nicht in ein Scrum-Team integrieren. Warum bezeichnen wir sie überhaupt als Helden?

In den beschriebenen Wasserfallprojekten gab es immer wieder Situationen, in denen Teams auf einem anderen Kontinent oder in einem anderen Projekt nicht mehr weiterwussten. Jene SAP-Profis, die gleich mehrmals zu solchen Rettungsaktionen erschienen sind, haben dann allmählich den Heldenstatus erreicht. Die Projektleiter wussten ganz genau: „Wenn wir ein Problem mit der Produktionsplanung haben, dann lassen wir Tom einfliegen. Tom weiß, wie es geht.“

Ich war selbst einer dieser Helden und ich gebe zu: Ich habe es geliebt, zu Rettungseinsätzen auf der ganzen Welt zu fliegen. Natürlich Business Class und immer nur direkt. Denn nur wer im Flugzeug von München nach Tokio schlafen kann, ist nach der Landung fit für die Arbeit. Eine Zeitlang war das ein wirklich schönes Leben und das will man auch nicht einfach so aufgeben.

Doch Helden gehen eigene Wege, sie sind nicht sehr teamfähig und teilen ihr Wissen nicht gerne. Sie ziehen in einem Sprint die Aufgaben an sich und werden mit diesen Aufgaben dann oft nicht fertig. Helden erkennen Sie auch daran, dass sie für mehrere Teams arbeiten, ihre Kapazität aufteilen und vom Chef jede Menge Spezialaufgaben bekommen. Oft genügt ein böser Blick

des Helden und das Team schweigt. Damit vermässeln die Helden dem Team den Sprint, die Storys, das Commitment und die Velocity.

Der Heldenstatus ist zweifellos ein Problem, doch es lässt sich in den Griff bekommen. Wie das funktionieren kann, werden Sie in [Kapitel 4](#) erfahren.

## **Scrum funktioniert nicht mit externen Partnern**

In SAP-Projekten werden oft externe Profis mit Spezial-Know-how eingekauft, dadurch entstehen neue Teams. Einkäufer und Einkaufsleiter haben mit dem Argument „Scrum funktioniert nicht mit externen Partnern“ vollkommen recht. Denn es werden oft die alten Musterverträge eingesetzt und das alte Spiel „Einkäufer vs. Anbieter“ gespielt – schon in der Vergangenheit hat das nicht für die beste Stimmung gesorgt und die Denkweise dahinter ist alles andere als agil. Wer glaubt, Werkverträge mit dem günstigsten Anbieter schließen zu müssen, sollte tatsächlich jegliche Form von agiler Zusammenarbeit einfach vergessen. Nicht der billigste Partner steigert die Velocity, sondern derjenige, der das agile Arbeiten verstanden hat, es mit seinem Team lebt und Spaß daran hat.

Nun wird das Ganze vor allem in Deutschland noch ein Stück schwieriger. Hinter dem schönen Wort „Arbeitnehmerüberlassungsgesetz“ (AÜG) versteckt sich eine Regelung aus den 1970er-Jahren, die erst 2017 angepasst wurde. Das heißt aber nicht, dass dieses Gesetz deswegen modern wäre und agile Arbeitssituationen berücksichtigen würde. Es schützt externe Mitarbeiter auf eine so komplexe Art, dass Klagen auf Einstellung beim Auftraggeber – auf Basis des

Gleichstellungsgrundsatzes – immer wieder Erfolg haben. Deshalb habe ich ein gewisses Verständnis für die Rechtsabteilungen von Unternehmen: Sie versuchen, das Unternehmen zu schützen, indem sie die Aussage „Scrum funktioniert nicht mit externen Partnern“ immer wieder untermauern. Es soll nie zu einer wirklich agilen Zusammenarbeit kommen. Wenn dann noch die Einkäufer den Preis ins Unerträgliche drücken, ist die selbsterfüllende Prophezeiung gelungen. In der Praxis geht es dann nur noch um die Frage, wie schnell sich der Vertrag mit dem nicht performenden billigsten Anbieter wieder auflösen lässt.

Es geht aber tatsächlich anders bzw. agil. Zusammenarbeit beruht auf einer Partnerschaft zwischen Auftraggeber und externem Anbieter, nicht auf Unterbuttern. In der ersten Phase einer agilen Zusammenarbeit wird gegenseitiges Vertrauen aufgebaut, um das Risiko auf beiden Seiten zu minimieren. Mit neuen Vertragsformen, wie zum Beispiel dem „Agilen Festpreis“, gibt es Wege, um sich die Risiken zu teilen – zum Beispiel, indem die ersten Sprints mit zwei oder drei Partnern gestartet werden und erst danach eine endgültige Entscheidung fällt.

## **Scrum funktioniert nicht mit weltweit verteilten SAP-Teams**

Spätestens seit der Corona-Krise wissen wir: Arbeit *muss* ganz einfach mit verteilten Teams funktionieren – Unternehmen müssen eben Wege dafür finden und die Voraussetzungen schaffen. Grundsätzlich funktioniert Agilität mit weltweit verteilten Teams sehr gut, dafür gibt es inzwischen genügend Belege. Aufgrund der unterschiedlichen Zeitzonen, Sprachen und

Kulturen ist es aber eine Situation, die klar eine Herausforderung darstellt und in jeder Organisation individuell gelöst werden muss. Gerade in SAP-Projekten muss dafür auch die vertragliche Basis stimmen.

Die Remote-Arbeit mit weltweit verteilten Teams ist aber in SAP-Projekten sehr gut möglich. Die Erfahrungen, die ich damit seit vielen Jahren mache, sowie das Wissen zu passenden Praktiken und Tools gebe ich in diesem Buch weiter. Eine wichtige Voraussetzung für die verteilte Zusammenarbeit ist, dass sie auf den Werten von Scrum basiert: Offenheit, Mut, Respekt, Commitment, Fokus.

Ich kann alle genannten Argumente, warum Scrum und SAP-Projekte nicht zusammenpassen, sehr gut nachvollziehen. Da ich es seit Jahren erfolgreich anders mache, kann ich Ihnen sagen: Und es passt doch zusammen. Wer den Schritt wagt und sich mit Offenheit darauf einlässt, kann Großes erwarten. Wenn ich mir sämtliche agil abgewickelten SAP-Projekte in meiner Karriere ansehe, kann ich guten Gewissens sagen:

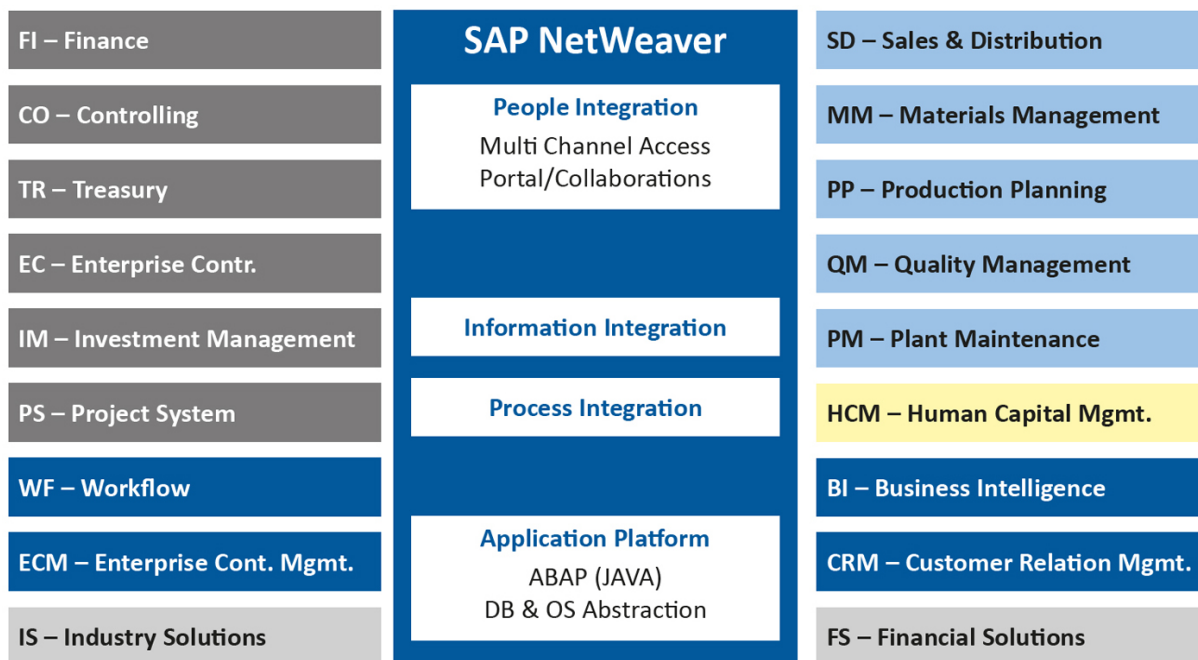
**Wir haben in den meisten Projekten das Doppelte in der vorgegebenen Zeit erreicht.**

Das bedeutet, dass diese Projekte um den Faktor 2 bis 3 schneller abgeschlossen waren. Das spart richtig Geld und dabei werden die wichtigen Ressourcen, zum Beispiel die teuren Helden, richtig eingesetzt.

Vielleicht denken Sie jetzt: „Der spinnt doch.“ Nun, ich kann Ihnen nur meine Erfolgsgeheimnisse und besten Tipps weitergeben. Irgendwo auf der Welt wird es immer Kollegen und agile Teams geben, die es sogar noch besser hinkriegen. Daher beschreibe ich in diesem Buch nicht die besten, sondern ganz

einfach meine „guten Praktiken“: für den Aufbau eines neuen Teams, aber auch für die Arbeit mit bereits länger bestehenden Teams sowie für die Skalierung. Probieren Sie die Vorschläge aus – vielleicht erst mal als Experiment für zwei, drei oder vier Sprints. Danach suchen Sie in der Retrospektive nach Verbesserungen oder sogar besseren Ansätzen. Seien Sie sich einfach im Klaren, dass die Hälfte der agilen Experimente schief geht. Haben Sie den Mut, gescheiterte Experimente zu beenden und neue zu beginnen.

## 1.2 Was SAP so speziell macht



**Bild 1.2** Überblick über die SAP-Module

Die Systeme von SAP waren zweifellos ein Meilenstein im Bereich der ERP-Software, den Anwendungen für alle unternehmerischen

Aufgaben. Die einzelnen SAP-Module sind sehr eng miteinander verknüpft: Die Daten und Informationen müssen daher nicht mehrmals erfasst und auch nicht mehrfach in den Datenbanken abgespeichert werden.

Ein Beispiel: Wenn Waren an einen Kunden geliefert werden, umfasst das unter anderem die Schritte Lieferungserstellung, Kommissionierung und Warenausgangsbuchung. Fast gleichzeitig werden im SAP-System alle Informationen in den Lägern, auf den Bestandskonten, in den Finanzkonten und in den Reporting-Systemen aktualisiert. Die allgemeinen Daten zu diesem Vorgang – zum Beispiel Informationen über den Kunden, wie Kundennummer, Adresse und Steuerdaten – werden nur einmal im System gespeichert und stehen dem Vertrieb, dem Verkauf, dem Lager und der Buchhaltung übergreifend zur Verfügung. Wir sprechen von einem sogenannten „nahtlosen Datenfluss“. Dadurch sind zwischen den einzelnen Modulen keine Softwareschnittstellen mehr notwendig, diese würden nur einen zusätzlichen Wartungsaufwand benötigen und damit höhere Kosten verursachen. Das Beispiel in [Bild 1.3](#) zeigt den Datenfluss (Document Flow) von einer Kundenanfrage zu einem Angebot an den Kunden, der dann zum Auftrag wird. Es wird eine Lieferung erstellt, der Warenausgang gebucht und die Rechnung erzeugt.

Geschäftspartner: 000001000568		179 Arnold St, San Francisco CA 94104, United States	
Beleg	Datum	Status	
➤➤➤ ➔ Terminauftrag 0000000014525	24.10.2020	erledigt	
👉 ➤➤➤ Auslieferung 0080000011002	28.10.2020	in Arbeit	
👉 ➤➤➤ Rechnung 0090000004586	28.10.2020	erledigt	
👉 ➤➤➤ Buchhaltungsbeleg 0090000004586	28.10.2020	nicht ausgeziffert	

**Bild 1.3** Datenfluss in einem SAP-System