



Jetzt lerne ich Scrum

Muster



Ein Lehrmittel für den Einstieg in das Vorgehen mit Scrum und für die Vorbereitung auf die Scrum-Zertifizierung.

[www.dassachbuch.com](http://www.dassachbuch.com)

Verlag DASsachbuch

ISBN: 978-3-033-06775-2

31.03.2018

---

# Inhaltsverzeichnis

1.	Agiles Projektmanagement.....	6
1.1	Einsatz agiler Methoden.....	6
1.2	Agiles Projektmanagement im Überblick.....	6
1.3	Das Wasserfall-Modell war nicht mehr geeignet.....	7
1.4	Der Unterschied zwischen „klassischen“ und „agilen“ Projekten.....	8
1.4.1	Agiles Projektmanagement für alle Projekte?.....	9
1.5	Das Agile Manifest.....	10
1.6	Agile Prinzipien.....	11
1.7	Agile Methoden.....	12
1.8	Repetitionsfragen zum Kapitel 1.....	13
2.	Das Scrum Framework.....	Fehler! Textmarke nicht definiert.
2.1	So entstand Scrum.....	Fehler! Textmarke nicht definiert.
2.2	Scrum als Management Framework.....	Fehler! Textmarke nicht definiert.
2.3	Die drei Säulen von Scrum.....	Fehler! Textmarke nicht definiert.
2.4	Scrum im Schnellüberblick.....	Fehler! Textmarke nicht definiert.
2.5	Scrum agile Prozesse in der Unternehmens-Governance.....	Fehler! Textmarke nicht definiert.
2.6	Repetitionsfragen zum Kapitel 2.....	Fehler! Textmarke nicht definiert.
3.	Die Rollen in Scrum.....	Fehler! Textmarke nicht definiert.
3.1	Rollenmodell.....	Fehler! Textmarke nicht definiert.
3.2	Product Owner.....	Fehler! Textmarke nicht definiert.
3.3	Das Scrum Team.....	Fehler! Textmarke nicht definiert.
3.4	Der Scrum Master.....	Fehler! Textmarke nicht definiert.
3.4.1	Die Aufgaben des Scrum Masters.....	Fehler! Textmarke nicht definiert.
3.4.2	Eigenschaften und weitere Funktionen des Scrum Masters.....	Fehler! Textmarke nicht definiert.
3.5	Weitere Rollen.....	Fehler! Textmarke nicht definiert.
3.6	Das Scrum Team und sein Umfeld.....	Fehler! Textmarke nicht definiert.
3.7	Repetitionsfragen zum Kapitel 3.....	Fehler! Textmarke nicht definiert.
4.	Scrum-Werte.....	Fehler! Textmarke nicht definiert.
4.1	Scrum-Werte.....	Fehler! Textmarke nicht definiert.
4.1.1	Verpflichtung (Commitment).....	Fehler! Textmarke nicht definiert.
4.1.2	Mut (Courage).....	Fehler! Textmarke nicht definiert.
4.1.3	Fokus (Focus).....	Fehler! Textmarke nicht definiert.
4.1.4	Offenheit (Openness).....	Fehler! Textmarke nicht definiert.
4.1.5	Respekt.....	Fehler! Textmarke nicht definiert.
5.	Die Scrum-Ereignisse – Übersicht.....	Fehler! Textmarke nicht definiert.
5.1	Scrum Ereignisse / Events.....	Fehler! Textmarke nicht definiert.
5.2	Der Sprint.....	Fehler! Textmarke nicht definiert.
5.3	Sprint Planning.....	Fehler! Textmarke nicht definiert.
5.4	Das Daily Scrum.....	Fehler! Textmarke nicht definiert.
5.5	Das Sprint Review.....	Fehler! Textmarke nicht definiert.
5.6	Sprint-Retrospektive.....	Fehler! Textmarke nicht definiert.
5.7	Product Backlog Refinement.....	Fehler! Textmarke nicht definiert.
5.8	Repetitionsfragen zum Kapitel 5.....	Fehler! Textmarke nicht definiert.
6.	Die Scrum Artefakte – Übersicht.....	Fehler! Textmarke nicht definiert.
6.1	Scrum Artefakte.....	Fehler! Textmarke nicht definiert.
6.2	Das Product Backlog.....	Fehler! Textmarke nicht definiert.
6.3	Das Sprint Backlog.....	Fehler! Textmarke nicht definiert.
6.4	Das Product Increment.....	Fehler! Textmarke nicht definiert.
6.5	Die "Definition of Done" (DoD).....	Fehler! Textmarke nicht definiert.
6.6	Der Scrum Workflow.....	Fehler! Textmarke nicht definiert.

6.7	Repetitionsfragen zum Kapitel 6 .....	Fehler! Textmarke nicht definiert.
7.	Das Product Backlog (Zusatzinformationen) .....	Fehler! Textmarke nicht definiert.
7.1	Erklärung Product Backlog .....	Fehler! Textmarke nicht definiert.
7.2	Übersicht über das Product Backlog .....	Fehler! Textmarke nicht definiert.
7.3	Das Product Backlog detaillieren .....	Fehler! Textmarke nicht definiert.
7.4	Den Aufwand schätzen .....	Fehler! Textmarke nicht definiert.
7.5	Das Risiko der Anforderungen bestimmen .....	Fehler! Textmarke nicht definiert.
7.6	Das Product Backlog priorisieren .....	Fehler! Textmarke nicht definiert.
7.6.1	Priorisierung nach MoSCoW .....	Fehler! Textmarke nicht definiert.
7.6.2	Priorisieren ist eine wiederkehrende Optimierungsaufgabe .....	Fehler! Textmarke nicht definiert.
8.	Das Sprint Planning .....	Fehler! Textmarke nicht definiert.
8.1	Was ist ein Sprint .....	Fehler! Textmarke nicht definiert.
8.2	Der Sprint-Ablauf .....	Fehler! Textmarke nicht definiert.
8.2.1	An welchem Wochentag startet ein Sprint? .....	Fehler! Textmarke nicht definiert.
8.3	Die „Definition of Ready“ .....	Fehler! Textmarke nicht definiert.
8.4	Die „Definition of Done“ (DoD) .....	Fehler! Textmarke nicht definiert.
8.5	Vorbereitung auf das Sprint Planning .....	Fehler! Textmarke nicht definiert.
8.6	Sprint Planning 1 .....	Fehler! Textmarke nicht definiert.
8.7	Sprint Planning 2 .....	Fehler! Textmarke nicht definiert.
8.8	Repetitionsfragen zum Kapitel 8 .....	Fehler! Textmarke nicht definiert.
9.	Das Sprint-Durchführung .....	Fehler! Textmarke nicht definiert.
9.1	Defekte und Ereignisse im Sprintablauf .....	Fehler! Textmarke nicht definiert.
9.2	Das Sprint Backlog .....	Fehler! Textmarke nicht definiert.
9.3	Das Scrum Taskboard .....	Fehler! Textmarke nicht definiert.
9.4	Den Sprint abbrechen .....	Fehler! Textmarke nicht definiert.
9.5	Repetitionsfragen zum Kapitel 9 .....	Fehler! Textmarke nicht definiert.
10.	Das Daily Scrum .....	Fehler! Textmarke nicht definiert.
10.1	Was ist ein Daily Scrum .....	Fehler! Textmarke nicht definiert.
10.2	Das Daily Scrum vorbereiten und durchführen .....	Fehler! Textmarke nicht definiert.
10.3	Den Sprint-Umfang anpassen .....	Fehler! Textmarke nicht definiert.
10.4	Den Fortschritt überwachen .....	Fehler! Textmarke nicht definiert.
10.4.1	Das Taskboard .....	Fehler! Textmarke nicht definiert.
10.4.2	Das Sprint Burndown Chart .....	Fehler! Textmarke nicht definiert.
10.5	Repetitionsfragen zum Kapitel 10 .....	Fehler! Textmarke nicht definiert.
11.	Das Sprint Review .....	Fehler! Textmarke nicht definiert.
11.1	Das Ziel des Sprint Reviews .....	Fehler! Textmarke nicht definiert.
11.2	Vorbereitung und Ablauf des Sprint-Reviews .....	Fehler! Textmarke nicht definiert.
11.3	Das Resultat des Sprint Reviews .....	Fehler! Textmarke nicht definiert.
11.4	Repetitionsfragen zum Kapitel 11 .....	Fehler! Textmarke nicht definiert.
12.	Die Sprint-Retrospektive .....	Fehler! Textmarke nicht definiert.
12.1	Kontinuierliche Verbesserung und stetiges Lernen .....	Fehler! Textmarke nicht definiert.
12.2	Vorbereiten der Retrospektive .....	Fehler! Textmarke nicht definiert.
12.3	Die Retrospektive durchführen .....	Fehler! Textmarke nicht definiert.
12.3.1	Sicherheit schaffen .....	Fehler! Textmarke nicht definiert.
12.3.2	Was waren die wichtigsten Ereignisse .....	Fehler! Textmarke nicht definiert.
12.3.3	Erkenntnisse sammeln .....	Fehler! Textmarke nicht definiert.
12.3.4	Massnahmen definieren .....	Fehler! Textmarke nicht definiert.
12.3.5	Massnahmen priorisieren .....	Fehler! Textmarke nicht definiert.
12.4	Repetitionsfragen zum Kapitel 12 .....	Fehler! Textmarke nicht definiert.
13.	Glossar .....	Fehler! Textmarke nicht definiert.
13.1	Begriffe und Erklärungen .....	Fehler! Textmarke nicht definiert.
14.	Stichwortverzeichnis .....	Fehler! Textmarke nicht definiert.

# Vorwort

## Zielgruppeninfo

Dies ist ein Buch für alle, die am agilen Projektmanagement interessiert sind und wissen wollen, wie die bekannteste agile Methode in der Softwareentwicklung, Scrum, funktioniert.

Dies ist kein Buch mit allen Details, vielen Beispielen, Geschichten und Anekdoten. Für das gibt es umfassendere und viel teure Bücher. Hier lernen Sie konzentriert die wichtigsten Aspekte des agilen Projektmanagement und Scrum.

Ob Sie bereits im Softwarebereich arbeiten, Student, angehender Arbeitgeber für Software oder bereits in einem agilen Projekt arbeiten – dieses Buch liefert Ihnen in kompakter Form das notwendige Wissen, agiles Projektmanagement und besonders Scrum besser zu verstehen und erfolgreich anzuwenden.

---

## Impressum

© 2018 DASsachbuchverlag, Horn, Schweiz

Nachdruck oder Reproduktion in irgendeiner Form (Druck, Fotokopie oder anderes Verfahren) sowie die Einspeicherung, Verarbeitung, Vervielfältigung und Verbreitung mit Hilfe elektronischer Systeme jeglicher Art sind für Einzelseiten erlaubt. Alle Übersetzungsrechte vorbehalten.

Alle Rechte, einschliesslich derjenigen des auszugsweisen Abdruckes sowie der fotomechanischen und elektronischen Wiedergabe, vorbehalten. Es wird darauf hingewiesen, dass die im Buch verwendeten Software- und Hardwarebezeichnungen sowie Markennamen und Produktbezeichnungen, wie z. B. The Scrum Guide™, der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patenrechtlichem Schutz unterliegen.

1. Auflage 2018

Autor, Herausgeber, Redaktion, Satz, Gestaltung (inkl. Umschlaggestaltung),  
Texte, Bilder: Roland und René Wanner  
Titelbild: Shutterstock

Gedruckt und hergestellt in Deutschland 2018

ISBN 978-3-033-06775-2

## Nützliche Links

---

Download the official Scrum Guide	<a href="http://www.scrumguides.org/download.html">http://www.scrumguides.org/download.html</a>
Scrum Wikipedia	<a href="https://de.wikipedia.org/wiki/Scrum">https://de.wikipedia.org/wiki/Scrum</a>
Scrum-Einführung	<a href="http://scrum-master.de/Scrum-Einfuehrung">http://scrum-master.de/Scrum-Einfuehrung</a>

## Inhalt und Aufbau dieses Lehrmittels

---

- The Scrum Guide™ von Ken Schwaber und Jeff Sutherland ist die offizielle Guideline für Scrum. Sie wird periodisch aktualisiert. Dieses Buch basiert auf der neusten Version des Scrum Guides vom November 2017.
- Der Lehrplan besteht aus 12 Hauptkapiteln. Ein Kapitel umfasst eine Lerneinheit (LE). Eine Lerneinheit besteht aus Lernzielen, welche in den Unterkapiteln beschrieben sind. Am Ende eines jeden Hauptkapitels befinden sich Musterfragen und die Lösungen (ausser Kapitel 7).
- Das Lehrmittel wird ergänzt durch ein Stichwortverzeichnis.

## Fragen zum Lehrmittel

---

**Mail:**  
[dassachbuch@bluewin.ch](mailto:dassachbuch@bluewin.ch)

MUSTER

## 1. Agiles Projektmanagement

<b>Lernziele<sup>1</sup></b>	<ul style="list-style-type: none"> <li>1.1 Wo werden agile Methoden eingesetzt</li> <li>1.2 Agiles Projektmanagement im Überblick</li> <li>1.3 Warum eignet sich das Wasserfallmodell nicht</li> <li>1.4 Unterschied zwischen agilen und klassischen Projekten</li> <li>1.5 Das Agile Manifest</li> <li>1.6 Was für Agile Prinzipien gibt es</li> <li>1.7 Was sind agile Methoden / Was ist Agilität</li> </ul>
------------------------------	---

### 1.1 Einsatz agiler Methoden

Agile Methoden werden in immer mehr Projekten eingesetzt – in der Software-Produktentwicklung schon seit vielen Jahren erfolgreich, bei anderen Projektarten stehen sie hier noch am Anfang. Der Trend zeigt aber eindeutig, immer mehr Unternehmen beschäftigen sich mit agilen Methoden oder sind bereits an deren Einführung. Auch in traditionellen Industrien wie zum Beispiel im Automobil- oder Flugzeugbau, führt man damit Projekte bereits erfolgreich durch. Beispiele dafür sind Toyota, BMW oder SAAB Technologies.

Seit kurzem wird versucht, agile Methoden auch in anderen Geschäftsprozessen einzusetzen als bei Projekten. Dies ist eine spannende Herausforderung, welche die gesamte Unternehmenskultur, Führungssysteme und die Zusammenarbeit stark verändern wird. Ich bin mir aber nicht so sicher, ob viele Grossunternehmen diesen Schritt in den nächsten Jahren schaffen werden. Kleinunternehmen sind diesbezüglich viel anpassungsfähiger.

Agile Methoden im Scrum zu lernen und zu verstehen ist relativ einfach. Die agilen Werte und Grundprinzipien zu verinnerlichen und zu leben ist hingegen einiges schwieriger – und hier haben und machen viele Unternehmen noch Mühe haben.

Der Einsatz von Methoden auf Unternehmensebene kommt schlussendlich aber nur zustande durch radikale Veränderungen innerhalb der Organisationen; hier sind wir noch weit davon entfernt, grosse Fortschritte zu machen. Auf Projektebene sind wir hingegen schon sehr weit fortgeschritten.

Agile Methoden haben einen radikalen Einfluss auf die Führungs- und Kompensationssysteme in Unternehmen. Manager verlieren Macht und Einfluss bei selbstorganisierenden Teams. Dies wird die Veränderung, speziell in der Unternehmenskultur, nicht einfach machen. Das Prinzip von selbstorganisierenden und funktionsübergreifenden Teams, Chefs als Coaches ohne Führungsfunktion und der Abbau von Managementebenen, war schon vor Jahrzehnten kurz ein „interessantes Thema“. Ich hoffe, in den nächsten Jahren haben wir damit mehr Erfolg. In der Softwareentwicklung mit Scrum werden agile Methoden schon seit einigen Jahren erfolgreich angewendet. Ein erster, wichtiger Schritt ist also schon gemacht!

### 1.2 Agiles Projektmanagement im Überblick

Imposante Projekte wurden schon vor tausenden von Jahren durchgeführt. Denken Sie z.B. an die Steinstruktur von Stonehenge, die 3500 Jahre v. Chr., und die ägyptischen Pyramiden, die 2500 v. Chr., erbaut wurden oder in der neueren Zeit an die mittelalterlichen Burgen, Festung, Schlösser und Kathedralen, die Dampfmaschine, Autos, die Atombombe oder die Wolkenkratzer. Dies waren teilweise riesige und komplexe Projekte für ihre Zeit. Software und Softwareprojekte gibt es jedoch erst seit ein paar Jahrzehnten. In den 1950er Jahren war Software noch Teil der Hardware und wurde als Programmcode bezeichnet. Ich kann mich auch noch gut an die Lochkarten erinnern, mit denen in

<sup>1</sup> generelle Lernziele – nicht nach einem bestimmten Lernplan

den 1970er Jahren Werkzeugmaschinen gesteuert wurden. Die Lochkarten waren die Programme, um mit Werkzeugmaschinen z.B. Teile zu fräsen.

Softwareentwicklungsmethoden gab es bis in die 1960er Jahre noch keine. Der Systems Development Life Cycle (SDLC) war der erste, der in dieser Zeit entstand, mit dem Ziel, grosse, funktionale Business Systeme zu entwickeln. Alle Projekte wurden bis in die 90er Jahre nach dem sequentiellen Wasserfallmodell abgewickelt, alle Anforderungen wurden zu Projektbeginn festgelegt, dann wurden Konzepte, Spezifikationen und Pläne erstellt und dann das Produkt gebaut und eingeführt.

Die Softwareentwicklung in den 1990er Jahren wurde durch die objektorientierte Programmierung und durch den Aufstieg des Internets und den [Dot.com-Boom](#) geprägt. Hier war Time-to-Market und Firmenwachstum entscheidend, d.h. die Entwicklungszyklen für Software wurden immer kürzer.

### 1.3 Das Wasserfall-Modell war nicht mehr geeignet

Mit dem starren Wasserfall-Prozess war man in der Software-Entwicklung deshalb immer weniger zufrieden, besonders weil die Projekte immer komplexer wurden. Produktlebenszyklen immer kürzer und das Umfeld und die Anforderungen dynamischer. Man benötigte immer schneller brauchbare Software, nicht mit allen Funktionen, sondern in Schritten. Leichtgewichtiger, flexibler und schneller sollte die Softwareentwicklung werden und weniger Administration war gewünscht.

Neue Methoden sollten den Softwareentwicklungsprozess flexibler und schlanker gestalten – als Gegenbewegung zu den schwerwichtigen und bürokratischen, traditionellen Methoden, wie eben z.B. dem Wasserfall-Modell. Diese Bedürfnisse stießen eine aktive Entwicklung von Methoden in der Softwareentwicklung an.

- 1986 entwickelte Boehrs erste Ansätze eines iterativen Software-Entwicklungsprozesses mit dem risikoorientierten Spiralmodell.
- 1991 wurde Rapid Application Development (RAD) vorgestellt
- 1995 präsentierten Jeff Sutherland und Ken Schwaber zusammen ein Dokument, welches die Scrum Methode erstmals beschrieb
- 1998 wurde der Rational Unified Process (RUP) vorgestellt
- 1999 wurde Extreme Programming (XP) vorgestellt, welches auf grosses Interesse bei den Softwareentwicklern stiess.

Wie Sie sehen, sind erste Ansätze zu agiler Softwareentwicklung bereits Anfang der 1990er Jahre zu finden. Popularität erreichte die agile Softwareentwicklung erstmals 1999, als Kent Beck das erste Buch zu Extreme Programming (XP) veröffentlichte.

Das Interesse an Extreme Programming ebnete den Weg auch für andere agile Prozesse und Methoden. Die Bezeichnung „agil“ für diese Art der Softwareentwicklung wurde ausgewählt von 17 Vertretern von verschiedenen Softwareentwicklungsmethoden im Februar 2001, bei einem Treffen in Utah (USA). Dies als Ersatz für das bis dahin gebräuchliche leichtgewichtig (engl. lightweight). Bei diesem Treffen wurde auch das Agile Manifest (siehe Seite 10) formuliert. Daraus hat sich dann im Laufe der Jahre die Bezeichnung „agiles Projektmanagement“ entwickelt, denn nicht nur Softwareprojekte lassen sich agil planen und steuern, sondern auch andere Projektarten.

Das Ziel agiler Softwareentwicklung ist es, den Entwicklungsprozess flexibler und schlanker zu machen, als das bei den klassischen Vorgehensmodellen der Fall ist. Agile Softwareentwicklung zeichnet sich durch selbstorganisierende Teams sowie eine iterative und inkrementelle Vorgehensweise aus. Es wird versucht, mit geringem bürokratischem Aufwand und weniger Regeln auszukommen und sich so schnell an Veränderungen anzupassen, ohne dabei das Risiko für Fehler zu erhöhen.

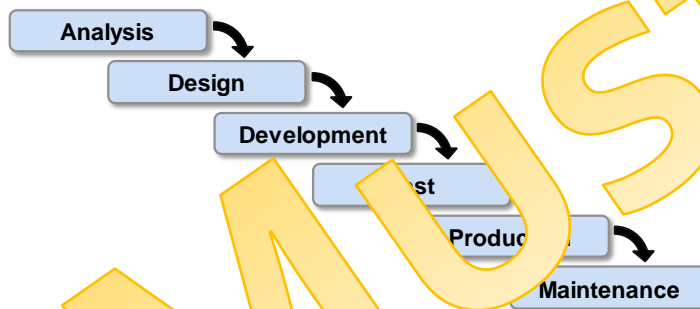
## 1.4 Der Unterschied zwischen „klassischen“ und „agilen“ Projekten

Bei „klassischen“ Projekten werden vom internen oder externen Kunden (Auftraggeber) zu Projektbeginn klare Ziele und Anforderungen definiert, die sich während der Projektdurchführung möglichst nicht mehr ändern. Am Ende des Projektes erhält der Kunde, was er am Anfang definiert hat.

Das Projekt wird strikt in nacheinander folgenden Phasen durchgeführt. Eine vorhergehende Phase muss beendet sein, bevor mit der nächsten gestartet werden kann. Das Projektergebnis entsteht im Ablauf der Phasen, bis es dann am Ende der letzten Phase vollständig fertiggestellt ist. Dieser Ablauf wird Wasserfall-Modell genannt.

Je weiter das Projekt fortschreitet, desto weniger Einfluss kann der Kunde auf das Ergebnis nehmen. Eine grosse Einschränkung beim Wasserfallmodell ist, dass jede Änderung oder neue Anforderungen, die der Kunde in einer späteren Projektphase noch umgesetzt haben will, ein Mehrfaches kostet, als wenn er diese am Anfang definiert hätte.

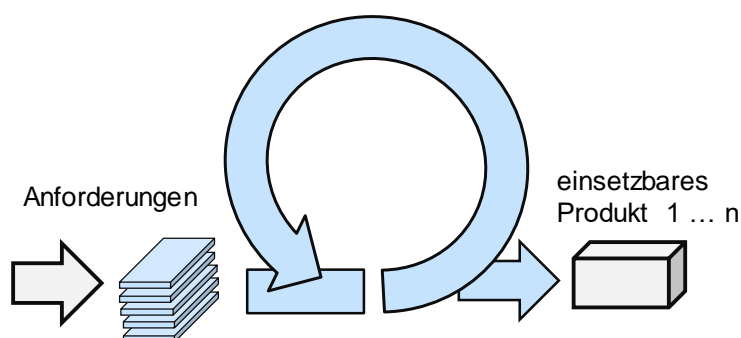
Sie können sich vermutlich gut vorstellen, wie man ein Haus baut. Wenn in diesem Projekt die Bauarbeiter bereits die Wände mauern und Sie wünschen sich plötzlich noch ein Zimmer mehr, dann wird dies sehr teuer oder es ist fast unmöglich.



Agile Methoden, z.B. Scrum, werden in Entwicklungsprojekten (besonders von Software) verwendet, um sich der hohen Dynamik der Ziele, Anforderungen, Umfeld und Erwartungen anzupassen. Man setzt dabei u.a.

- auf enge Zusammenarbeit zwischen Kunden, dem Product Owner und dem sich weitgehend selbst organisierenden Team
- auf kurze Entwicklungszyklen, nach denen Änderungen und neue Anforderungen in die Planung zusätzlich aufgenommen werden können (Iterationen mit definierter Länge; typisch sind 14 bis maximal 30 Tage).

Entwicklungszyklus 1 ... n





### 1.4.1 Agiles Projektmanagement für alle Projekte?

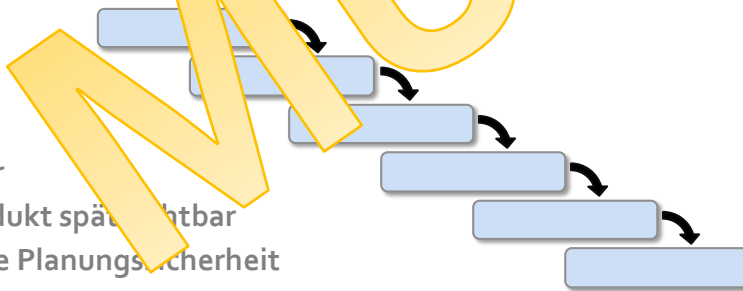
Agiles Projektmanagement setzt sich immer mehr durch, auch ausserhalb der traditionellen Softwareentwicklung. Viele sagen dem Wasserfallmodell schon den Tod voraus. Soweit wird es aber nicht kommen. Versuchen Sie ein Haus, Flugzeugträger, Werkzeugmaschine oder eine Fabrikationsanlage mit agilem Projektmanagement zu entwickeln und zu fertigen. Unmöglich! Dort müssen praktisch alle Anforderungen zu Projektbeginn bekannt sein, und der Spielraum, diese während dem Projektablauf zu ändern oder neue hinzuzufügen, ist sehr gering.

Das heisst aber nicht, dass agile Methoden nicht für bestimmte Lieferobjekte von „Wasserfallprojekten“ eingesetzt werden können. Überall, wo zum Beispiel Software entwickelt werden muss, können wahrscheinlich agile Entwicklungsmethoden eingesetzt werden, zum Beispiel für die Software einer Werkzeugmaschine oder eines Car Entertainment Systems.

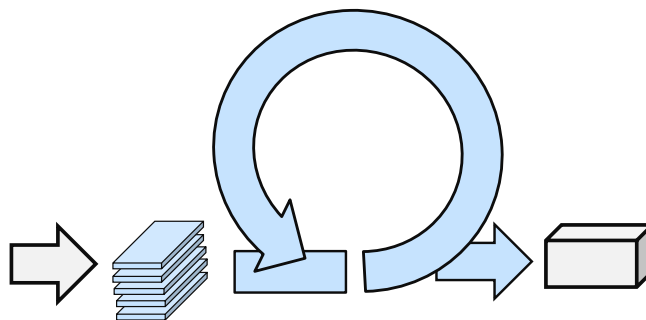
Bestimmte Prinzipien, Praktiken und Werte des agilen Projektmanagement werden aber auch das traditionelle Projektmanagement in Zukunft stark beeinflussen. Ich denke schon an die Zusammenarbeit im Team und an die agilen Werte und Prinzipien. Aber generell, wenn physische Produkte entwickelt werden, ist die Flexibilität im Entwicklungsprozess eindeutig geringer als in der Softwareentwicklung.

## Wasserfall-Modell

- starr
- Produkt spät sichtbar
- hohe Planungssicherheit
- konstante Anforderungen notwendig



## Agiles Vorgehen



- wenig Struktur, iterativ
- früh einsetzbare Resultate
- hohe Flexibilität, marktnah
- ändernde Anforderungen willkommen

## 1.5 Das Agile Manifest

Im Frühjahr 2001 verabschiedeten 17 projekterfahrene Software-Entwickler in Utah (USA) das sogenannte „Manifesto for Agile Software Development“, heute vor allem unter der Kurzbezeichnung „Agile Manifesto“ bekannt. Diese Erstunterzeichner, darunter auch die beiden Scrum-Gründer Ken Schwaber und Jeff Sutherland, formulierten mit dem Agile Manifesto erstmals die zentralen Werte agiler Softwareentwicklung – ein Meilenstein und zugleich das Fundament des agilen Projektmanagements.

Die Werte des Agilen Manifests ([www.agilemanifesto.com](http://www.agilemanifesto.com)) sind paarweise beschrieben, wobei die Werte auf der linken Seite jeweils höher eingeschätzt werden als die Werte auf der rechten Seite. Dies heisst aber nicht, dass diese bedeutungslos sind.

### Das Agile Manifest lautet folgendermassen:

„Wir erschliessen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir die folgenden Werte geschätzt und gelernt:

- Individuen und Interaktionen stehen über Prozessen und Werkzeugen
- Funktionierende Software steht über einer umfassenden Dokumentation
- Zusammenarbeit mit dem Kunden geht über der Vertragsverhandlung
- Reagieren auf Veränderung geht über dem 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.“

Ich würde die agilen Werte des Agilen Manifests eher als „Grundsätze der agilen Softwareentwicklung bezeichnen“, damit diese nicht mit den Scrum-Werten verwechselt werden.

### Die Bedeutung des Agilen Manifests

In der Bedeutung für das agile Projektmanagement hat das Agile Manifesto bis heute nichts eingebüsst. Im Laufe der Jahre sind sogar noch Tausende weitere Befürworter und Unterzeichner hinzugekommen. Den Verfassern ist damals gelungen, die Kerngedanken moderner Software-Entwicklung trotz teils höchst unterschiedlicher Auffassungen und Herangehensweisen ein für alle Mal auf einen gemeinsamen Nenner zu bringen.

Der grosse Fortschritt bestand und besteht darin, dass mit dem agilen Manifest endlich ein Wertesystem in Stein gemeisselt wurde, das eine konkrete Vorgehensweise umreisst. Gleichzeitig konnte damit auch der bis dahin gängige, vergleichsweise unscharfe Begriff der „leichtgewichtigen Software-Entwicklung“ abgelöst werden. Insofern lässt sich das Agile Manifesto am ehesten als die „Geisteshaltung der Agilität“ verstehen, und diese Geisteshaltung lebt bis heute in agilen Methoden wie Scrum oder Kanban bzw. deren Regeln und Rollen fort.

Das Agile Manifest hatte auch zur Folge, dass Mitarbeiter fortan nicht einfach nur als Ressourcen betrachtet wurden, sondern als Akteure im Mittelpunkt stehen.

Agilität im Projektmanagement fordert und fördert die individuellen Fähigkeiten der Mitarbeiter, indem man ihnen mehr Verantwortung überträgt und kreative Gestaltungsmöglichkeiten einräumt. Dadurch wird der Weg geebnet für effektivere und erfolgreichere Projektverläufe.

## 1.6 Agile Prinzipien

Das Agile Manifest kann missverstanden oder falsch interpretiert werden, deshalb wurden die Aussagen von den Verfassern des Manifests durch zwölf Prinzipien detaillierter erklärt.

Es ist zum Beispiel nicht die Meinung, dass es keine Dokumentationen mehr geben sollte. Gemeint waren die mehreren hundert Seiten langen Dokumente, die niemand liest noch pflegt.

Im Zentrum der Prinzipien steht der Mensch, sei es das Projektteam oder der Kunde. Die agilen Prinzipien sind eine wesentliche Orientierungshilfe für erfolgreiche agile Projekte, die sie bis heute immer noch uneingeschränkt gültig.

**Die zwölf agilen Prinzipien** hinter dem Agilen Manifest:

1. **Den Kunden zufriedenstellen:** Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. **Änderungen willkommen heissen:** Reaktionen auf Veränderung, selbst spät in der Entwicklung, sind willkommen. Agile Prozesse nutzen Veränderung zum Wettbewerbsvorteil des Kunden.
3. **Häufige Auslieferung:** Liefern funktionierende Software regelmässig, innerhalb weniger Wochen oder Monate, und bevorzugen die kürzeren Zeitspanne.
4. **Bereitschaft zur Zusammenarbeit:** Fachexperten und Entwickler müssen während des Projekts eng zusammenarbeiten.
5. **Unterstützung und Vertrauen schenken:** 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. **Persönliche Kommunikation:** Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungs-Teams zu übermitteln, ist im Gespräch, von Angesicht zu Angesicht.
7. **Funktionierende Software:** Funktionierende Software ist der wichtigste Massstab für den Fortschritt.
8. **Nachhaltige Geschwindigkeit:** Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Nutzer sollten ein gleichmässiges Tempo auf unbegrenzte Zeit halten können.
9. **Streben nach technischer Exzellenz:** Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. **Einfach ist besser:** Einfachheit – die Kunst, die Menge der Arbeit, die nicht getan wird, zu maximieren – ist entscheidend.
11. **Selbstorganisiert agieren:** Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. **Überprüfen und anpassen:** In regelmässigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

### Der Mensch, die Zusammenarbeit und das Wissen stehen im Zentrum

Die agilen Werte und Prinzipien stellen den Menschen in den Vordergrund, sowie deren Zusammenarbeit und neue Führungsmodelle, aber auch das Wissen und schlanke Prozesse. Hier steht der Mitarbeiter im Zentrum, genauso wie der Kunde, für den das Produkt entwickelt wird.

Das Wissen im Unternehmen und der Wissensmitarbeiter wird leider immer noch viel zu wenig beachtet, dabei ist dies heute der entscheidende Wettbewerbsvorteil gegenüber der Konkurrenz.

Wissensmitarbeiter brauchen wenig Führung und wollen mitentscheiden. Zeigen von Anerkennung und Wertschätzung ist wichtig, aber auch das Fördern von Kreativität und Mut. Hier braucht es auch eine offene Fehlerkultur.

Bei der Zusammenarbeit geht es um das Miteinander und nicht das Gegeneinander. Hier sind Werte wie Rücksicht, Vertrauen, Toleranz, Offenheit und Respekt gefragt.

Agilität ist eine Haltung, die immer mehr gefragt ist, nicht nur in der Softwareentwicklung. Wir werden immer mehr zu einer Wissensgesellschaft. Veränderungen, speziell im Businessumfeld, aber auch mit IT-Werkzeugen, mit denen wir arbeiten, verändern sich immer schneller. Dabei geht die persönliche Kommunikation oft unter – und genau auch die will die agile Bewegung wieder stärken.

Wenn Sie die agilen Prinzipien und Werte anschauen, dann können Sie, dass diese nicht nur für agile Softwareprojekte anwendbar sind, sondern auch für andere Projektarten.

## 1.7 Agile Methoden

Wie Sie bereits ein paar Mal in früheren Kapiteln gelesen haben, entstanden viele Grundelemente agiler Methoden schon in den 1970er Jahren in Japan (Toyota (Toyota Production System), Canon und bei NEC und wurden 1986 im Harvard Business Review Artikel "The New New Product Development Game" beschrieben.

Im Projektmanagement wurden neue Methoden entwickelt, die allerdings fast ausnahmslos auf die Softwareentwicklung zugeschnitten sind. Erste Ansätze entstanden in der agilen Softwareentwicklung bereits Anfang der 1990er Jahre. Popularität erreichte die agile Softwareentwicklung erstmals 1999, als Kent Beck das Buch zu Extreme Programming veröffentlichte. In der Industrie findet man agile Methoden hauptsächlich bei den Automobilproduzenten.

In der folgenden Aufstellung finden Sie einige wichtige agile Methoden in der Softwareentwicklung. Je weiter oben sie steht, desto verbreiteter und bekannter ist sie.

- Scrum
- Extreme Programming (XP)
- Unified Process (RUP, AUP, OUP)
- Feature Driven Development (FDD)
- Adaptive Software Development (ASD)

### Was ist Agilität?

Agilität ist die Eigenschaft von Schnelligkeit, Leichtgewichtigkeit und Anpassungsfähigkeit. Dies umfasst:

- Die Fähigkeit, Neues entstehen zu lassen und **sich anpassen können** in einem sich schnell verändernden Geschäftsumfeld.
- Die Fähigkeit, Ressourcen **schnell zu repriorisieren**, wenn sich Anforderungen, Technologie und Wissen verändern.
- Auf Marktveränderungen und entstehende Trends **schnell reagieren können durch intensiven Kundenkontakt**.
- Verwenden von **evolutionären, inkrementellen** und **iterativen** Methoden, um eine optimale Kundenlösung zu liefern.

## 1.8 Repetitionsfragen zum Kapitel 1

1) Agile Methoden .....

- A) ... sind technologie- und toolorientiert
- B) ... verhindern das Fehlschlagen von Projekten
- C) ... fördern die Zusammenarbeit der Beteiligten
- D) ... fordern das Einhalten eines strikten Projektplans

2) Wo liegen die Vorteile Agiler Methoden gegenüber klassischen Methoden (z.B. Wasserfall)?

- A) ...hohe Planungssicherheit
- B) ...stabile Entwicklung ist gewährleistet
- C) ...Anforderungen an das System sind weitgehend bekannt
- D) ...ändernde Anforderungen sind möglich
- E) ...Risikofaktor sind frühzeitig sichtbar

3) Was ist kein agiles Prinzip?

- A) ... Auslieferungen erst am Schluss des Projektes
- B) ... selbstorganisiert agieren
- C) ... laufend überprüfen und anpassen
- D) ...ändernde Anforderungen nur im Ausnahmefall
- E) ... Unterstützung leisten und Vertrauen schenken

Lösungen: 1) C 2) D + E 3) A + D