



**PSSST ...
LERN DOCH, WAS
DU WILLST!**

Vera Gehlen-Baum & Manuel Illi

Agiles Lernen für zukunftsorientierte Unternehmen

Inhalt

1. **Einleitung**
2. **Agiles Lernen – was ist das?**
 - 2.1 Die Ursprünge der Agilität
 - 2.2 Das Agile Manifest
 - 2.3 Die Agile Softwareentwicklung nach Scrum
 - 2.4 Der Kern der Agilität
 - 2.5 Der agile Lernansatz
 - 2.6 Agiles Lernen im Unternehmen
 - 2.7 Das agile und das klassische Lernsetting im Vergleich
3. **Theoretische Hintergründe**
 - 3.1 Lernziele und Motivation
 - 3.2 Scripting
 - 3.3 Selbstgesteuertes Lernen und Metakognition
 - 3.4 Feedback
4. **Rollen im agilen Lernen**
 - 4.1 Der Lernende
 - 4.2 Der Lerncoach
 - 4.3 Der Instruktor
 - 4.4 Die Führungskraft
 - 4.5 Die Kunden und Partner

5. Meetings und Prozesse im agilen Lernen

5.1 Der Kick-off: Das Erstgespräch im agilen Lernen

5.2 Das Planning im agilen Lernen: »Wo geht die Reise hin?«

5.3 Der Lernsprint: »Los geht's!«

5.4 Review und Retro im agilen Lernen: »Wieder was gelernt!«

5.5 Das Zwischengespräch: »Na, wie gehts?«

5.6 Die Rahmenbedingungen, Vor-und Nachbereitung

6. Agiles Lernen in der Praxis

6.1 Einen Anfang wagen: das MVP

6.2 Erfahrungen sammeln: der Lerncoach

6.3 Sicherheit gewinnen: eine solide Ausbildung

6.4 Gesundes Wachstum: agiles Lernen und agile Transitionen

6.5 Return on Investment: von der Wertschätzung zur Wertschöpfung

6.6 Blick in die Zukunft: Lernen als Radar für Veränderung

Literaturverzeichnis

Autorenvorstellung

1. Einleitung

Agilität kann man nicht kaufen -Agilität muss man leben!

Wir freuen uns, dass Sie sich für agiles Lernen interessieren. Agile Prinzipien und Methoden gepaart mit der richtigen Einstellung können das Zusammenleben und die Zusammenarbeit von Menschen wirklich verbessern – diese Erfahrung haben wir in den letzten Jahren in vielen Projekten und Teams immer wieder gemacht.

Vielleicht teilen Sie diese Erfahrung, weil Sie schon in einem agilen Umfeld arbeiten – vielleicht ist Ihnen aber auch das Schlagwort ›Agilität‹ bisher nur vom Hörensagen bekannt und Sie möchten mehr über neue Lernansätze erfahren. Egal was Sie neugierig auf dieses Buch gemacht hat, wenn Sie sich dafür interessieren, wie man Lernprozesse spannender, effizienter und nachhaltiger gestalten kann, dann halten Sie genau das richtige Buch in Ihren Händen.

Unsere Leidenschaft ist gelebte Agilität. Aus dieser Leidenschaft und aus der jahrelangen Erfahrung in den Bereichen agile Coaching, agile Projekte, Lehre, Training und Seminarleitung entstand der Gedanke, endlich auch die Art und Weise, wie Menschen lernen und sich persönlich weiterentwickeln, agiler zu gestalten. Wir haben uns auf den Weg gemacht, um in ganz unterschiedlichen Kontexten agile Lernoasen zu schaffen.

Auf dem Weg zu einem wirklich agilen Lernansatz haben wir viel recherchiert, geforscht, entwickelt, erprobt und unsere

Ergebnisse schließlich in die Praxis gebracht. Mit diesem Buch wollen wir unsere Erkenntnisse und Erfahrungen teilen. Es soll eine theoretisch fundierte aber auch praxistaugliche Einführung in das agile Lernen sein.

Darum geht es in diesem Buch

Es erwarten Sie Hintergründe aus dem Bereich der Pädagogik und Lerntheorie sowie der Softwareentwicklung, aber vor allem möchten wir konkrete Tipps für die praktische Realisierung des agilen Lernens geben. Vieles davon adressiert den betrieblichen Kontext, lässt sich aber auch in anderen Bereichen umsetzen. Wir haben auch schon sehr positive Erfahrungen mit agilem Lernen in der Jugendarbeit, bei NGOs und in der Schule sammeln können.

Bisher gibt es relativ wenig Forschungsergebnisse zu konkret angewandtem agilem Lernen. Wir haben daher in [Kapitel 2](#) und [3](#) bereits gut belegte Ergebnisse der Organisationstheorie, Lehr-Lernforschung, Lernpsychologie und Pädagogik zusammengetragen und stellen jeweils ihre Bedeutung für das agile Lernen dar.

In [Kapitel 4](#) und [5](#) geben wir praktische Tipps für die Gestaltung von Rollen, Prozessen und Meetings im agilen Lernprozess, wobei wir die theoretischen Überlegungen mit unseren eigenen Untersuchungsergebnissen und Praxiserfahrungen verbinden. Diese Kapitel sind besonders interessant, wenn Sie selbst agil lernen möchten oder andere als Coach, als Führungskraft oder Personalentwickler in ihrem Lernen unterstützen möchten.

Die Tipps und praktischen Hinweise, die wir im Folgenden vorstellen, wirken auf ersten Blick vielleicht sehr detailliert oder sogar schematisch. Der Grund dafür ist recht einfach. Das agile Lernen kann zu Beginn herausfordernd sein, weil

es in zentralen Aspekten ganz anders funktioniert als die Lernumgebungen, in denen wir aufgewachsen sind und die uns jahrelang geprägt haben. Um hier ein Umdenken anzustoßen, ist es hilfreich, zunächst die neuen Rollen, Prozesse und Meetings relativ kleinteilig zu beschreiben, um dann mit der Zeit die Vorgaben zu reduzieren und freier zu interpretieren. Der Erfolg dieses Vorgehens ist übrigens recht gut in der sog. Skript-Forschung untersucht worden (vgl. [Kap. 3.2](#)).

Mit dem letzten Kapitel möchten wir Ihnen eine praktische Hilfestellung geben, wenn Sie überlegen, das agile Lernkonzept in Ihrer Organisation oder Ihrem Unternehmen zu etablieren. Wir zeigen einen gangbaren Weg von den ersten Schritten bis hin zu einer umfassenden, neuen und inspirierenden Lernkultur.

Je nachdem, was Sie besonders interessiert, können Sie die einzelnen Kapitel auch für sich lesen. Thematisch wird jeweils von den praktischen auf die theoretischen Kapitel (und umgekehrt) verwiesen.

Nach dem Buch ist vor dem Buch

Als wir uns daran gemacht haben, ein Buch über agiles Lernen zu schreiben, wurde uns schnell klar: Ein Buch mit agilem Inhalt sollte auch agil geschrieben werden. Aber was heißt es, ein Buch agil zu schreiben? In unserem Fall bedeutet das:

Das Buch hat viele Autoren: Von der ersten Idee, über das gemeinsame Brainstorming im Team, über die kritischen Nachfragen von Kollegen, bis hin zum Tippen und Gegenlesen – dieses Buch entstand aus der unglaublich anregenden Zusammenarbeit von vielen, vielen Menschen

bei QualityMinds, auch wenn auf dem Buchdeckel nur zwei Namen stehen. Für uns war diese Art des Schreibens neu, sie hat aber sehr viel Spaß gemacht und jeden weitergebracht.

Der Inhalt ist organisch gewachsen: Recht schnell hatten wir eine grobe Struktur des Inhalts, doch immer wieder haben Gespräche oder Teammeetings dazu geführt, dass Abschnitte umgeschrieben, gelöscht oder neue Teile aufgenommen wurden. Und dieser Prozess geht immer noch weiter, denn ...

Das Buch ist nicht fertig: Wir wollen auch in Zukunft an den Themen und Inhalten dieses Buchs arbeiten. Dank Book on Demand können wir dann z. B. die neuen Erfahrungen und Studien in eine aktuelle Ausgabe einfließen lassen. Dabei kommen auch Sie ins Spiel, denn ...

Auch Sie können mitmachen: Wir freuen uns über Feedback, Anregungen und Hinweise für die künftige Weiterentwicklung des Ansatzes und dieses Buchs. Haben Sie bereits Erfahrungen mit agilem Lernen gesammelt oder interessante Informationen? Dann lassen Sie künftige Leser und uns teilhaben und schreiben Sie uns unter: learning@qualityminds.de

Dank

Abschließend möchten wir allen Unterstützern, Helfern, Freunden und Kollegen danken, die dieses Buch möglich gemacht haben: Carina, Elena, Rick, Markus, Sanne, Susanne, Tine und dem gesamten QualityLearning-Team für das gemeinsame Nachdenken und Entwickeln; Madeleine

und Rosetta für den Satz und die graphische Gestaltung; Bettina, Eva, Tamara und Ralph für den Input und das Gegenlesen; Michael und Robert für die vorbehaltlose Unterstützung, das kritische Nachfragen und den nötigen Freiraum. Und last but not least: Danke an die ganze QualityMinds-Community: Ihr seid einfach wunderbar!

Vera Gehlen-Baum und Manuel Uli im Frühjahr 2019

Noch eine Anmerkung: Gleichstellung und Gleichberechtigung sind uns wichtig. Die weibliche Form (Lernerin, Mitarbeiterin etc.) soll der männlichen Form in diesem Buch gleichgestellt sein; lediglich aus Gründen der leichteren Lesbarkeit wurde fast durchgehend die männliche Form gewählt.

2. Agiles Lernen - was ist das?

Etwas zu lernen und dann anwenden zu können, z. B. eine neue Fremdsprache, ist sehr motivierend. Agil und dynamisch agieren zu können, selbst wenn eine Situation neu und herausfordernd ist, auch das ist etwas Großartiges. Wir möchten im folgenden Abschnitt zeigen, dass man Lernen und Agilität so kombinieren kann, dass etwas ganz Neues entstehen kann. Dieses Neue kann sogar unser Verständnis davon ändern, wie wir Menschen lernen, leben und arbeiten (Gehlen-Baum & Hoppenz, 2018).

Fangen wir zu nächst einmal mit dem schillernden Wort ›Agilität‹ an. Das deutsche Adjektiv ›agil‹ leitet sich über das Französische von dem lateinischen Wort ›agilis‹ ab, was so viel wie »leicht beweglich, behände, aktiv« bedeutet. Das Konzept, über das heute im Bereich der Unternehmensführung und der Organisationsentwicklung so viel gesprochen wird, ist aber wesentlich jünger.

2.1 Die Ursprünge der Agilität

Wissenschaftlich wurde der Begriff ›Agilität‹ in der Soziologie in den frühen 1950er Jahren untersucht. In ihrem Text »Working Paper in the Theory of Action« (1953) betrachteten die Autoren R. F. Bales, E. A. Shils und T. Parsons Gruppenverhalten, das Parsons (1978) dann in seinem systemtheoretischen AGIL-Schema bündelte.

Eine verbreitete Beschäftigung mit Agilität hat jedoch ihren Ursprung in den 1990er Jahren. Das war jene Zeit, in der

nicht nur der ›Personal Computer‹ Einzug in viele Haushalte hielt, sondern auch die Komplexität der Programme rasant anstieg. Deutlich wird das am Beispiel von Microsofts Windows: Hatte die erste Version (NT1.0) im Jahr 1993 etwa 4 bis 5 Millionen Zeilen Code, so umfasste Windows XP (NT 5.1) im Jahr 2001 bereits das Zehnfache, nämlich etwa 40 Millionen Zeilen Code (O'Brien, 2005).

Klassische Verfahren, die in anderen Kontexten, wie z. B. dem Automobilbau, etabliert waren, stießen hier schnell an ihre Grenzen. Im Jahr 1991 forderte das Iacocca Institute der Lehigh University unter dem Schlagwort ›Agilität‹ neue Produktionsformen und definierte den Begriff wie folgt: »A manufacturing system with extraordinary capability to meet the rapidly changing needs of the market place [...] ideally in real-time response to customer demand« (zitiert nach Costin et al., 2001, S. 632). Dasselbe Institut gründete kurz darauf das Agile Manufacturing Enterprise Forum (AMEF), um die geforderte dynamische Reaktivität zu propagieren.

Zu dem Aspekt der flexiblen Reaktion kamen Ende der 1990er Jahre die Aspekte proaktives Handeln, Kundenerfordernisse und Prozessadaptivität (Förster&Wendler, 2012). Softwareunternehmen griffen diese Aspekte schnell auf, denn sie mussten schon immer ihre Produkte in kürzeren Zyklen entwickeln und dabei auf überraschend aufkommende Anforderungen reagieren. Kaum etwas ist schlimmer, als eine Anwendung, die zwar ausgereift und stabil, aber zum Zeitpunkt ihres Rollouts bereits überholt ist und nicht zu den Bedürfnissen der User passt.

Im Februar 2001 trafen sich 17 führende Softwareentwickler in den verschneiten Bergen Utahs, um ihre Erfahrungen in einem gemeinsamen Lösungsansatz zusammenzutragen. Das Resultat war das »Manifest für agile

Softwareentwicklung« (oder kurz: »Agiles Manifest«), dessen Werte und Prinzipien bis heute eine ungeahnte Wirkung entfalten – und das in ganz unterschiedlichen Kontexten wie z. B. der Unternehmensführung und eben auch dem Lernen.

2.2 Das Agile Manifest

Das Agile Manifest beruht auf der Erkenntnis, dass sehr viele Faktoren über den Erfolg eines komplexen Prozesses entscheiden und dass deswegen eine wohlüberlegte Priorisierung unabdingbar ist. Faktoren können Interaktionen zwischen Aktanten, die Zusammenarbeit mit Kunden, Werkzeuge und Tools, das Reagieren auf Veränderung, funktionierende Software, die Dokumentation, Vertragsverhandlung, Planungen etc. sein.

Die Verfasser des Agilen Manifests trieb damals der Wunsch, eine klare Priorisierung der genannten Aspekte vorzunehmen, um die hoch komplexen Prozesse der Softwareentwicklung klarer, effizienter und für die Beteiligten angenehmer zu gestalten, ohne jedoch die Wertschöpfung aus den Augen zu verlieren. Das Manifest beruht auf vier Grundwerten, die genaugenommen eine Priorisierung darstellen:

»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.« (Beck et al., 2001)

Aufbauend auf diesen vier Grundsätzen formulieren die Verfasser des Manifests zwölf agile Prinzipien, die praktische Hilfestellungen für die konkrete Umsetzung der Werte sein sollen, z. B. »Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen« oder »In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann, und passt sein Verhalten entsprechend an.« (Beck et al., 2001)

Agilität soll nach dem Wunsch der Verfasser des Manifests in einem ganz konkreten Kontext, nämlich der Softwareentwicklung, dazu beitragen, eine neue Balance zwischen technischen, strategischen, marktbedingten und kundenorientierten Anforderungen zu finden: »We want to restore a balance« lautet Ihr Credo, und weiter: »We embrace modeling, but not in order to file some diagram in a dusty corporate repository. We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. We plan, but recognize the limits of planning in a turbulent environment.«

Wie man erahnen kann, ging es den Verfassern nicht darum, Agilität zu einem Patentrezept oder einem Allheilmittel zu erklären. Die genannten Werte und Prinzipien sollten vielmehr für Programmierer, Softwarearchitekten, Manager, Projektleiter, Kunden – letztlich für alle Stakeholder des Prozesses – eine Anregung sein, wie man transparenter, effizienter und angenehmer zusammenarbeiten kann.

Agilität, so hört man auch oft, ist keine Methode, sondern ein Mindset (Hofert, 2018). Ein wesentlicher Teil davon ist das Selbst-Tun. Denn Prinzipien verlieren schnell ihre Legitimation, wenn sie von denen, die sie verkünden, nicht selbst praktiziert werden. Es geht beim Agilen Manifest also auch darum, den Blick für die wesentlichen Dinge zu schärfen und hinderliche Muster zu verabschieden – und zwar greifbar im täglichen Miteinander. Auf dieser agilen Philosophie beruhen konkrete Verfahren und Methoden, die dabei helfen sollen, Agilität praktisch umzusetzen. Ein solches Prozessmodell soll kurz vorgestellt werden, weil es auch das agile Lernen sehr gut unterstützen kann.

2.3 Die Agile Softwareentwicklung nach Scrum

Scrum ist neben vielen anderen Ansätzen, wie z. B. Kanban oder Extrem Programming, einer der weitverbreitetsten Ansätze, bei der Umsetzung agiler Prozesse. Das englische Wort »Scrum« stammt aus dem Rugby-Sport und bezeichnete ursprünglich das gemeinsame Nach-vorne-Bringen des Balls durch ein »eng gedrängtes Team«. Jeff Sutherland und Ken Schwaber, die schon am Agilen Manifest beteiligt waren, schufen Anfang der 2000er Jahre mit dem »Scrum Guide« ein ausgefeiltes Prozessmodell, das diesen Gedanken aufgreift (Sutherland & Schwaber, 2013).

Scrum beruht auf den drei Säulen Transparenz, Überprüfung und Anpassung, durch die der tatsächliche, d. h. empirisch messbare Fortschritt von komplexen Prozessen sichtbar und steuerbar wird. Ins Zentrum des Scrum-Prozesses stellen Sutherland und Schwaber das Scrum-Team, in dem eine klare Rollenaufteilung für transparente Verantwortlichkeiten sorgt (Gloger, 2016):

Product Owner: Er ist maßgeblich für die Vision des Produkts, für die Wertsteigerung und das große Ganze verantwortlich. Zu seinen Aufgaben zählt, für die Priorisierung von Anforderungen und Tasks sowie für die Kommunikation mit den Stakeholdern zu sorgen (also z. B. Management, Kunden, Anwender).

Entwicklungsteam: Das Team entwickelt die Software und ist interdisziplinär besetzt. Während in der klassischen Entwicklung unterschiedliche Personen fix zugewiesene Aufgaben und Zuständigkeiten wie z. B. Software-Architekten, Entwickler, Tester etc. hatten, die teils in unterschiedlichen Abteilungen organisiert waren, arbeiten die Teammitglieder in Scrum cross-funktional zusammen. Als Team sind alle gemeinsam dafür verantwortlich, was auf welche Weise und mit welchen Abnahmekriterien in einer Sprint-Phase entwickelt wird. Das bedeutet, dass jedes Mitglied eine breite Allgemeinqualifikation mit einer tiefgehenden Spezialqualifikation kombiniert und je nach Bedarf Aufgaben übernimmt.

Scrum Master: Seine Verantwortung ist es, dafür zu sorgen, dass das Entwicklungsteam möglichst frei von äußeren Hindernissen und Störungen arbeiten kann. Außerdem stellt er sicher, dass Scrum funktioniert, die Regeln eingehalten werden und offen kommuniziert wird.

Neben den Rollen legt Scrum auch eine klare zeitliche Struktur fest. Durch genau definierte Prozessphasen und Ereignisse werden verlässliche Prognosen über künftige Entwicklungen möglich. Das Risiko bleibt kalkulierbar, weil in kurzen Abständen greifbare Ergebnisse entstehen:

Sprint: Der Sprint ist ein fest begrenzter Zeitraum (max. 30 Tage, wobei mittlerweile der Trend verstärkt in die Richtung

von ein- oder zweiwöchigen Sprints geht), in dem ein nutzbares und potenziell auslieferbares Produktinkrement fertiggestellt wird. Sprints sollten in der Regel die gleiche Dauer und Struktur haben (auch wenn es dann und wann Sinn macht, Sprintlängen zu variieren).

Planning: Zu Beginn des Sprints setzt das Scrum-Team das Ziel der Sprint-Phase gemeinsam fest. Der Product Owner bereitet hierzu eine geordnete Liste an Anforderungen für das Produkt vor (= Product Backlog). Der Product Owner ist also für die Inhalte und die Priorisierung verantwortlich. Hierbei ist es wichtig, dass für jede Anforderung auch die Akzeptanzkriterien formuliert werden, unter denen eine Anforderung als zufriedenstellend umgesetzt gilt. Das Entwicklerteam wiederum schätzt die Komplexität, den Aufwand und das Risiko der verschiedenen Anforderungen im Product Backlog und wählt der Reihe nach so viele davon aus, wie im nächsten Sprint aller Voraussicht nach bearbeitet werden können. Hierbei kann das Entwicklerteam den Product Owner ggf. auch bitten, einzelne Punkte zu spezifizieren oder genauer zu formulieren. Das Team verpflichtet sich dann auf das anvisierte Ziel der Umsetzung und kann anschließend selbstständig wählen, wie es die Anforderungen realisiert möchte und wer welche Aufgabe bearbeitet.

Daily Scrum: Das ›Daily‹ ist ein tägliches, maximal 15-minütiges Treffen des Entwicklungsteams, in dem der aktuelle Fortschritt, die Herausforderungen und weitere Schritte besprochen werden.

Review: Gegen Ende des Sprints findet ein informelles Treffen des Scrum-Teams mit den Stakeholdern statt, bei dem das fertiggestellte Produktinkrement dem Product

Owner und falls gewünscht auch den Stakeholdern präsentiert und evaluiert wird. Das Thema ist das ›Was‹, also das Resultat des Sprints, und die evtl. nötigen Änderungen für künftige Sprints. Hierbei führt das Entwicklerteam vor, welche Funktionalität realisiert wurde- der Product Owner wiederum vergleicht das Präsentierte mit den im Planning vereinbarten Akzeptanzkriterien.

Retrospektive: Nach dem Sprint reflektiert das Scrum-Team die eigene Arbeitsweise und Kollaboration. Hier wird also das ›Wie‹ thematisiert mit dem Ziel, für die kommenden Sprints dazuzulernen und sich kontinuierlich zu verbessern. Wie schon im Agilen Manifest wird damit auch im Serum dem Thema Lernen eine besondere Bedeutung zugesprochen.

Product Backlog - Sprint Backlog: Der Serum Guide beschreibt zusätzlich zu Rollen und Ereignissen auch Artefakte, zu denen das ›Product Backlog‹ und das ›Sprint Backlog‹ zählen. Beides sind Listen, in denen die Aufgaben für das gesamte Projektvorhaben bzw. den gerade laufenden Sprint transparent verzeichnet und priorisiert werden. Für das agile selbstgesteuerte Lernen können diese Artefakte als hilfreiche Elemente übernommen werden. Mit ihnen können einerseits Lernziele konkret formuliert, aber auch allgemeine Lernthemen und Entwicklungsrichtungen im Blick behalten werden.

Aufgrund der klaren Zuständigkeiten (Rollen), zeitlichen Begrenzungen (Sprint-Phasen) und transparenten Artefakte können Ziele, Prozesse und zu erledigende Arbeitsschritte einerseits geplant werden und andererseits trotzdem offen bleiben für (auch kurzfristig) auftretende Veränderungen wie