

Scrum in 20 Minuten verstehen

von Frank Schatz

Inhaltsverzeichnis

Warum dieses Buch?	3
Was ist Agilität?	5
Das agile Manifest	5
Scrum	10
Der Product Owner:	11
Das Development-Team:	11
Den Scrum Master:	11
Fokustechnik Timebox	12
Events in Scrum	12
Der Sprint	12
Ablauf eines Scrum Sprints	12
Sprint Planning	13
Daily Scrum	14
Sprint Review	14
Sprint Retrospektive	15
Artefakte in Scrum	16
Das Product Backlog	16
Die Definition of Done	16
Sprint Backlog	16
Das Product Increment	17
Burn-Down Chart	17
Rechtliches	19

Warum dieses Buch?

Scrum zu lernen ist kein Hexenwerk. Die Anwendung von Scrum ist eine emotionale, psychische und intellektuelle Herausforderung. Dieses Buch ist auf die steigende Nachfrage entstanden - *mal eben* Scrum zu erklären.

Der [aktuelle Scrum-Guide](#) ist sicherlich die beste Quelle für unverfälschte Informationen zu Scrum.

Ich habe den Inhalt des Scrum-Guides komprimiert und die für das Verständnis wesentlichen Informationen zusammengefasst. So hat auch noch das agile Manifest mit seinen 4 Werten und 12 Prinzipien einen Platz gefunden.

Ich wünsche mir, dass dir dieses E-Book nützt und der Inhalt verständlich ist.

Ich freue mich über Feedback per E-Mail an feedback@agileonline.de

Bleib agil Erfolgreich!

Frank Schatz, *Mallorca, im Herbst 2020*

WIDMUNG

Für alle zertifizierten Scrum-„irgendwas“ Menschen, die nicht verstanden haben, dass die Zertifizierung lediglich ihre Schuhe schnürt und erst dann wertvoll wird, wenn die Sohlen durchgelaufen sind.

Was ist Agilität?

Agilität ist eine Denkweise - ein Mindset.

Wie wird diese Denkweise in der Softwareentwicklung angewendet?

Auf dem Markt existieren viele Frameworks die ein agiles Mindset unterstützen. Es gibt jedoch keine Silberkugel. Jedes eingesetzte Framework muss inhaltlich individuell auf die jeweiligen Bedürfnisse und Ziele angepasst werden.

Die Kunst der Anpassung besteht darin, dass die Regeln, Werte und Prinzipien des Frameworks nicht verletzt werden.

Das agile Manifest

Agile Frameworks basieren auf dem agilen Manifest. Das Manifest entstand aus der Erkenntnis, dass bei einigen Projekten die Ergebnisse konstant besser waren, als bei anderen. Die Leiter dieser Projekte trafen sich 2001 in einem Skigebiet in Utah und analysierten, was in den gut funktionierenden Projekten anders war.

Sie formulierten aus den Erkenntnissen dieser Analyse das agile Manifest und deren 12 Prinzipien. Der bis dahin verwendete Begriff 'leichtgewichtig' wurde durch das Wort 'Agile', auf Vorschlag von Mike Beedle ersetzt.

Der Wortlaut des agilen Manifest ist:

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.

Die 12 Prinzipien lauten:

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 essenziell.

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.

Teilnehmer waren unter anderem Ken Schwaber und Jeff Sutherland - die Erfinder von Scrum.

Prädiktive, also Vorhersage- Modelle, wie zum Beispiel das Wasserfall-Modell, basieren auf zwei Annahmen:

1. Die Anforderungen sind zu Beginn exakt bekannt und
2. die Umsetzung erfolgt reibungslos

Wenn diese Annahmen nicht zutreffen, wird es um so teurer, je später diese Feststellung getroffen wird. Das wirkt sich negativ auf die Performance eines Projekts aus.

Agile Projekte basieren auf Validierung und Anpassung (Inspect & Adapt) und eine iterative und inkrementelle Lieferweise.

Es geht darum, in kurzen Zyklen funktionierende und potenziell auslieferbare Software zu erstellen die bei jeder Lieferung mehr Funktionalität enthält als die vorhergegangene Lieferung. Die Basis für diesen Prozess sind selbstorganisierte Entwicklungsteams und die enge Zusammenarbeit aller Beteiligten inkl. der Stakeholder.

Ein wesentlicher Vorteil von agilen gegenüber prädiktiven Entwicklungsmodellen, ist, dass Abweichungen, Fehler und Übersetzungsfehler der Anforderungen bei jeder Iteration aufgedeckt werden können. Es besteht die Möglichkeit einer schnellen und preiswerteren Korrektur.

Agile Modelle schaffen neue Herausforderungen.

So können sich Datenbankmodelle, Architekturen und Designs, häufig ändern.

Führungskräfte können das Gefühl haben, keine Kontrolle auf einer unvorhersehbaren Reise zu haben - was in gewisser Weise auch stimmt.

Die enge Zusammenarbeit zwischen Entwicklern und Kunden, die es in prädiktiven Modellen nicht gibt, ist ebenfalls eine neue Herausforderung.

Im Prinzip wird die gesamte Organisation herausgefordert. Wenn sich die Organisation oder Teile davon, gegen diese Veränderung ist, wird die Einführung eines agilen Modells scheitern. Gelingt die agile Einführung, ist eine gesteigerte Teamproduktivität, eine höhere Qualität der Produkte, ein schnellerer Return on Invest und eine verbesserte Kundenzufriedenheit zu verzeichnen.

Scrum

Ungeachtet weiterer agiler Modelle wie Extreme-Programming (XP), Lean Startup oder Kanban ist Scrum eines der am häufigsten eingesetzten agilen Frameworks.

Der Begriff "Scrum" wurde aus dem Rugby übernommen und bedeutet zu Deutsch, Gedränge. Im übertragenen Sinne, ist die enge Zusammenarbeit aller Beteiligten in einem Scrum-Projekt das Gedränge.

Wie bei einem Rugby-Spiel akzeptiert Scrum, dass das Ergebnis nicht vorherzusehen ist.

Scrum wurde von Ken Schwaber und Jeff Sutherland erfunden und wurde 1995 auf der OOPSLA-Konferenz der Öffentlichkeit vorgestellt.

Scrum basiert auf einem ein- bis vierwöchigen Zyklus, an dem die Teilnehmer*Innen definieren, entwerfen und entwickeln. Sie validieren das Product-Increment gemeinsam mit den Stakeholdern und Stakeholderinnen und liefern Stück für Stück ein funktionsfähiges Produkt.

Scrum kennt drei Rollen innerhalb eines Scrum Teams.

Der Product Owner:

Dem Product-Owner ist von der Organisation ermächtigt, sämtliche Entscheidungen bezüglich des Projektes allein zu treffen. Er verantwortet die erfolgreiche Umsetzung der Kundenanforderungen in einem zu erstellenden Produkt.

Das Development-Team:

Es erstellt aus den vom Product-Owner beschriebenen Anforderungen ein hochqualitatives Product-Increment. In einem Development-Team gibt es keine Titel wie Tester*Innen, Designer*Innen, Architekt*innen. Das Team ist eigenverantwortlich und muss interdisziplinär besetzt sein, um alle Anforderungen innerhalb des Teams mit technischer Exzellenz umzusetzen. Es verantwortet gemeinsam die Lieferung des Product-Increments.

Den Scrum Master:

Unterstützt und coacht alle Beteiligten in agiler Arbeitsweise und der Herstellung maximaler Transparenz. Dem Development-Team hilft der Scrum Master, die Events einzuhalten und richtig durchzuführen. Der Scrum Master unterstützt das Team bei der Beseitigung von Impediments (Hindernissen). Dem Product-Owner hilft der Scrum Master bei der Organisation des Product-Backlogs so das für das zu erstellende Produkt, maximaler Wert entstehen kann.

Fokustechnik Timebox

Timeboxing verfolgt den Ansatz, die zur Verfügung stehende Zeit für Aufgaben zu begrenzen. Die Einhaltung des festgelegten Zeitraums kann nur dann erfolgen, wenn fokussiert an dem Thema für die Timebox gearbeitet wird.

Timeboxing kann ausufernde Monologe oder sinnlose Diskussionen unterbinden. Zum Beispiel ist das Daily Scrum auf eine Timebox von 15 Minuten festgelegt. Bei einem 7-Personen Development-Team hat jeder im Schnitt rund 2 Minuten Zeit, seinen Beitrag zum Daily zu leisten. Nach Verstreichen der Timebox wird das Meeting hart abgebrochen.

Eine Timebox muss nicht bis zum Ende eingehalten werden. Meetings und Aufgaben können ***maximal*** bis zur Timebox-Grenze dauern. Die Zeitspanne kann auch kürzer sein, wenn das Ziel der Timebox erreicht wurde.

Events in Scrum

Der Sprint

Ein Sprint kann maximal 4 Wochen dauern. Innerhalb dieses Sprints wird vom Team ein Product-Increment hergestellt das tatsächlich "Fertig" im Sinne von "DONE" ist. In einem Scrum Projekt haben alle Sprints die gleiche Dauer. Eine Pause zwischen den Sprints ist nicht vorgesehen und unerwünscht.

Ablauf eines Scrum Sprints

Initial bei einem Scrum Projekt wird in einem Story-Workshop oder mittels einer anderen Methode ein erstes Product-Backlog erstellt. Die

Product-Vision ist dabei die wichtigste Eingangsgröße für dieses Event. Bei dieser Erstellung sind das gesamte Scrum-Team, sowie die Stakeholder, Anwender und Auftraggeber zugegen.

Mit dem Product-Backlog startet das Scrum Team in das Sprint Planning und definiert den Inhalt und Umsetzungsplan für den nächsten Sprint und legt ein Sprint Ziel mit Ausrichtung auf die Vision fest.

An jedem Arbeitstag eines Sprints, synchronisiert das Development Team seine Arbeiten und überprüft die Ausrichtung der Arbeiten auf das Sprint-Ziel.

Ist das Product-Increment Fertig - im Sinne von Done, wird dies im Sprint-Review den Stakeholdern präsentiert. Im Anschluss erfolgt die Sprint-Retrospektive für ein Inspect & Adapt des Scrum-Teams. Danach beginnt die nächste Iteration.

Sprint Planning

Im Sprint Planning wird geplant, welche Arbeit im kommenden Sprint enthalten ist und wie dieser Plan umgesetzt wird. Dieser Plan entsteht durch die gemeinschaftliche Arbeit des gesamten Scrum Teams.

Das Sprint Planning ist für einen einmonatigen Sprint auf maximal 8 Stunden befristet [Time Box].

Daily Scrum

Das Daily Scrum ist eine Time Box von 15 Minuten für das Development Team und findet an jedem Arbeitstag des Sprints statt. Das Development Team plant dabei die Arbeit für die nächsten 24 Stunden. Es überprüft die Arbeitsergebnisse seit dem letzten Daily Scrum und prognostiziert die im Sprint bevorstehende Arbeit, um die Zusammenarbeit und Leistung des Teams zu optimieren. Um die Komplexität zu reduzieren, wird das Daily Scrum an jedem Tag zur gleichen Uhrzeit am gleichen Ort abgehalten.

Im Daily Scrum überprüft das Development-Team seinen Fortschritt in Richtung des Sprint-Ziels und den Trend bei der Abarbeitung der Sprint-Backlog-Einträge. Das Daily Scrum erhöht die Wahrscheinlichkeit, dass das Development Team sein Sprint-Ziel erreicht.

Sprint Review

Am Ende eines Sprints wird ein Sprint Review abgehalten, um das Product-Increment zu überprüfen und das Product-Backlog bei Bedarf anzupassen. Während des Sprint-Reviews beschäftigen sich das Scrum Team und die Stakeholder gemeinsam mit den Ergebnissen des Sprints. Zusammen mit eventuellen Änderungen am Product-Backlog während des Sprints bieten diese die Basis für die gemeinsame Arbeit an möglichen neuen, den Wert des Produkts steigernden Punkten.

Beim Sprint Review handelt es sich um ein informelles Meeting, keinen Statusreport. Die Vorführung des Increments ist als Anregung für Feedback und die Basis für die Zusammenarbeit gedacht.

Für einen einmonatigen Sprint wird für dieses Meeting eine Obergrenze [Time Box] von vier Stunden angesetzt.

Sprint Retrospektive

Die Sprint-Retrospektive bietet dem Scrum Team die Gelegenheit, sich selbst zu überprüfen und einen Verbesserungsplan für den kommenden Sprint zu erstellen. Sie findet vor dem nächsten Sprint Planning statt.

Die Sprint-Retrospektive wird durchgeführt, um

- zu überprüfen wie der vergangene Sprint in Bezug auf die beteiligten Personen, Beziehungen, Prozesse und Werkzeuge verlief,
- die wichtigsten gut gelaufenen Elemente und mögliche Verbesserungen zu identifizieren und in eine Reihenfolge zu bringen und
- einen Plan für die Umsetzung von Verbesserungen der Arbeitsweise des Scrum Teams zu erstellen.

Für einen einmonatigen Sprint wird hierfür eine Obergrenze von drei Stunden angesetzt. Bei kürzeren Sprints ist das Meeting in der Regel kürzer.

Artefakte in Scrum

Das Product Backlog

Das Product Backlog ist eine geordnete Liste von allem, von dem bekannt ist, dass es im Produkt enthalten sein soll. Es dient als einzige Anforderungsquelle für alle Änderungen am Produkt. Der Product-Owner ist für das Product-Backlog, seine Inhalte, den Zugriff darauf und die Reihenfolge der Einträge verantwortlich.

Die Definition of Done

Damit die Beteiligten ein gemeinsames Verständnis von „Done“ im Sinne von 'Fertig' haben, wird eine Definition of Done vom Team erstellt. Dies ist im Prinzip eine Checkliste die abgearbeitet sein muss, damit etwas als Fertig, im Sinne von Done, bezeichnet werden darf.

Diese Definition of Done kann sich von Scrum Team zu Scrum Team unterscheiden.

Sprint Backlog

Das Sprint Backlog besteht aus, vom Development Team gewählten Product-Backlog-Einträgen die im Sprint umgesetzt werden sollen.

Das Sprint-Backlog wird vom Development Team um einen Plan (Aufgaben) ergänzt, der beschreibt wie das Product-Increment geliefert und das Sprint-Ziel erreicht wird. Das Sprint Backlog ist eine Prognose darüber, welche Funktionalität im nächsten Inkrement enthalten sein werden und macht die gesamte Arbeit sichtbar, die das Development Team für notwendig hält, um das Sprint-Ziel zu erreichen.

Das Product Increment

Ein Product-Increment ist ein Gegenstand inspizierbarer, fertiger, auslieferbarer Arbeit und ein Schritt in Richtung einer Vision oder eines Ziels.

Es ist das Ergebnis aus allen Sprint-Backlog-Einträgen, die in einem Sprint fertiggestellt sind, plus dem Resultat der Inkremente aller früheren Sprints. Am Ende eines Sprints muss das neue Product-Increment die Definition of „Done“ des Teams erfüllen.

Burn-Down Chart

Ein Burn-Down Chart wird vom Team im Sprint als Sprint-BurnDown-Chart eingesetzt, um die im Sprint erledigte Arbeit transparent grafisch darzustellen. Der Product Owner verwendet einen Burn-Down Chart, als Release BurnDown Chart um die bereits erledigte Arbeit innerhalb eines Releases transparent darzustellen. Das Sprint BurnDown Chart wird täglich während des

Daily-Scrum aktualisiert. Das Release Burn-Down Chart wird nach dem Sprint-Review aktualisiert.

Rechtliches

Haftungsausschluss / rechtliche Hinweise

© 2020 by agile online - frank schatz

Dieses Material ist urheberrechtlich geschützt.

Jede Verwertung der Inhalte (z.B. in Form von Vervielfältigungen, Übersetzungen oder Weiterverarbeitung in elektronischen Systemen) ohne meine explizite Zustimmung ist nicht gestattet. Das Internet und insbesondere das Themengebiet "Agile" ist ein sehr dynamisches Umfeld. Ich habe die Informationen nach bestem Wissen zusammengestellt, kann jedoch Fehler nicht ausschließen. Daher kann ich keine Garantien, Verpflichtungen oder Gewährleistungen übernehmen. Ebenso lehne ich jede juristische Verantwortung oder Haftung, die eventuell aus der Nutzung dieses Materials entstehen, ab und übernehme keine Gewähr dafür, dass die beschriebenen Verfahren frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen und Warenbezeichnungen 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 dürfen.