

Das Agile Manifest – 4 Werte und 12 Prinzipien

Die vier Werte des agilen Manifests

Das agile Manifest besteht aus vier Grundwerten und zwölf grundlegenden Prinzipien, die den agilen Ansatz bei der Softwareentwicklung vorgeben. Bei jeder agilen Methode werden die vier Werte auf verschiedene Weise umgesetzt, aber alle nehmen sie als Grundlage zur Entwicklung und Bereitstellung qualitativ hochwertiger, funktionierender Software.

1. Individuen und Interaktionen mehr als Prozesse und Werkzeuge

Der erste Wert im agilen Manifest lautet „Individuen und Interaktionen mehr als Prozesse und Werkzeuge“. Menschen mehr Wertschätzung entgegenzubringen als Prozessen und Werkzeugen ist einfach zu verstehen, denn es sind Menschen, die auf geschäftliche Anforderungen eingehen und den Entwicklungsprozess vorantreiben. Wenn der Prozess oder die Werkzeuge für die Entwicklung verantwortlich sind, ist das Team bei Änderungen weniger ansprechbar und die Wahrscheinlichkeit, dass sie die Anforderungen des Kunden erfüllen können, fällt geringer aus. Kommunikation ist ein Beispiel für den Unterschied, den die Wertschätzung von Individuen im Vergleich zum Prozess ausmacht. Im Falle von Individuen erfolgt die Kommunikation organisch und bei Bedarf. Im Falle des Prozesses folgt die Kommunikation einem Zeitplan und setzt spezifische Inhalte voraus.

2. Funktionierende Software mehr als umfassende Dokumentation

Traditionell wurde enormer zeitlicher Aufwand betrieben, um das Produkt für die Entwicklung und abschließende Bereitstellung zu dokumentieren. Technische Spezifikationen, technische Anforderungen, der technische Ausblick, Schnittstellendesign-Dokumente, Testpläne, Dokumentationspläne und Genehmigungen waren jeweils Voraussetzung. Die Liste war lang und ein Grund für die massiven Verzögerungen während der Entwicklung. Beim agilen Ansatz wird die Dokumentation nicht gänzlich verworfen, aber auf eine Art optimiert, die dem Entwickler die nötige Zeit für seine Arbeit gibt, ohne dass er durch die Details ausgebremst wird. Beim agilen Ansatz werden Anforderungen als User Stories dokumentiert, die dafür ausreichen, dass ein Softwareentwickler die Entwicklung einer neuen Funktion beginnen kann. Das agile Manifest schätzt die Dokumentation, aber es schätzt funktionierende Software mehr.

3. Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

Verhandlungen sind der Zeitraum, in dem der Kunde und der Produktmanager die Details einer Bereitstellung herausarbeiten. Während der Erfüllung im Anschluss können an bestimmten Punkten in der Entwicklung Details neu verhandelt werden. Zusammenarbeit ist eine ganz andere Geschichte. Mit Entwicklungsmodellen wie dem Wasserfallmodell verhandelten Kunden die Anforderungen für das Produkt, bevor die Entwicklung begonnen wurde, und das häufig sehr detailliert. So war der Kunde in den Entwicklungsprozess eingebunden, bevor die Entwicklung begann und nachdem sie abgeschlossen wurde, aber nicht während der eigentlichen Entwicklung. Das agile Manifest beschreibt einen Kunden, der den Entwicklungsprozess aktiv mitverfolgt und sich dabei auch einbringt. So kann bei der Entwicklung wesentlich besser auf die Anforderungen des Kunden eingegangen werden. Bei agilen Methoden kann der Kunde in regelmäßigen zeitlichen Abständen konsultiert werden, etwa im Rahmen einer Demonstration, die Entwickler des Projekts können aber auch einfach einen Endbenutzer im Team haben, der an allen Besprechungen teilnimmt und so sicherstellt, dass das Produkt die geschäftlichen Anforderungen des Kunden erfüllt.

4. Reagieren auf Veränderung mehr als das Befolgen eines Plans

Bei der traditionellen Softwareentwicklung wurden Veränderungen als Ausgaben betrachtet, man versuchte also, sie zu vermeiden. Die Absicht war, detaillierte, ausgefeilte Pläne zu entwickeln, die einen festgelegten Satz an Funktionen umfassten. Im Allgemeinen hatte alles die gleiche Priorität und es gab eine große Zahl verschiedener Abhängigkeiten von der Bereitstellung in einer bestimmten Reihenfolge, damit das Team die nächste Aufgabe bearbeiten konnte. Beim agilen Ansatz ermöglicht die Kürze einer Iteration, dass Prioritäten zwischen Iterationen verschoben werden und der nächsten Iteration neue Funktionen hinzugefügt werden können. Die Auffassung beim agilen Ansatz ist also, dass Änderungen ein Projekt verbessern. Sie bieten einen Mehrwert.

Am besten beschreibt die positive Bedeutung von Veränderung beim agilen Ansatz wohl das Konzept namens Method Tailoring, das in An Agile Information Systems Development Method in Use wie folgt definiert wird: „Ein Prozess oder eine Funktion, in der menschliche Agenten einen Systementwicklungsansatz für eine bestimmte Projektsituation durch responsive Änderungen in und ein dynamisches Zusammenspiel zwischen Kontexten, Intentionen und Methodenfragmenten festlegen.“ Agile Methoden ermöglichen es dem agilen Team, den Prozess zu modifizieren und ihn an das Team anzupassen statt umgekehrt.

Die zwölf Prinzipien des agilen Manifests

Die zwölf Prinzipien sind die Leitsätze für die Methoden, die unter dem Titel „The Agile Movement“ zusammengefasst werden. Sie beschreiben eine Kultur, in der Veränderungen begrüßt werden und der Kunde bei der Arbeit im Mittelpunkt steht. Sie demonstrieren darüber hinaus die Intention der Bewegung, die Entwicklung in Einklang mit geschäftlichen Anforderungen zu bringen, wie von Alistair Cockburn, einem der Unterzeichner des agilen Manifests, beschrieben.

Die zwölf Prinzipien der agilen Entwicklung umfassen Folgendes:

1. **Zufriedenstellung des Kunden durch frühe und kontinuierliche Auslieferung von Software** – Kunden sind zufriedener, wenn sie funktionierende Software in regelmäßigen Abständen erhalten, anstatt längere Zeit zwischen Produktversionen warten zu müssen.
2. **Berücksichtigung sich ändernder Anforderungen beim Entwicklungsprozess** – Die Fähigkeit, Verzögerungen zu vermeiden, wenn sich eine Anforderung oder ein Funktionswunsch ändert.
3. **Lieferung von funktionierender Software in regelmäßigen, bevorzugt kurzen Zeitspannen** – Scrum beruht auf diesem Prinzip, da das Team in sogenannten Softwaresprints oder Iterationen arbeitet, die eine regelmäßige Bereitstellung funktionierender Software sicherstellen.
4. **Zusammenarbeit von Fachexperten und Entwicklern während des Projekts** – Es werden bessere Entscheidungen getroffen, wenn das Business-Team und das technische Team auf einer Linie sind.
5. **Sicherstellung von Unterstützung, Vertrauen und Motivation bei allen Beteiligten** – Bei motivierten Teams ist die Wahrscheinlichkeit, dass sie bestmöglich arbeiten, höher als bei unzufriedenen Teams.
6. **Ermöglichung von Interaktionen von Angesicht zu Angesicht** – Kommunikation ist erfolgreicher, wenn sich Entwicklungsteams persönlich austauschen können.
7. **Als wichtigstes Fortschrittsmaß gilt die Funktionsfähigkeit der Software** – Die Lieferung funktionierender Software an den Kunden ist der wichtigste Faktor bei der Messung des Fortschritts.
8. **Agile Prozesse zum Einhalten eines gleichmäßigen Arbeitstempos** – Teams setzen ein Tempo fest, das sie für die Bereitstellung funktionierender Software reproduzieren und einhalten können, und setzen dieses Tempo bei jeder neuen Version um.
9. **Erhöhte Agilität durch Augenmerk auf technische Details und Entwürfe** – Die richtigen Fähigkeiten und gutes Design stellen sicher, dass das Team das Arbeitstempo beibehalten, das Produkt ständig verbessern und Änderungen berücksichtigen kann.
10. **Einfachheit** – Entwickeln Sie gerade genug, um die Arbeit für den Moment zu erledigen.
11. **Die besten Architekturen, Anforderungen und Entwürfe entstehen in selbstorganisierten Teams** – Sachkundige und motivierte Teammitglieder, die Entscheidungen treffen können, Verantwortung übernehmen, regelmäßig mit anderen Teammitgliedern kommunizieren und Ideen haben, die qualitativ hochwertige Produkte ermöglichen.
12. **Selbstreflexion im Hinblick auf Steigerung der Effektivität** – Selbstverbesserung, Prozessverbesserung, die Steigerung von Fähigkeiten und Techniken helfen Teammitgliedern dabei, effizienter zu arbeiten.

Die Intention des agilen Ansatzes ist es, die Entwicklung an den geschäftlichen Anforderungen auszurichten, und der Erfolg dieses Ansatzes ist offensichtlich. Agile Projekte sind kundenorientiert und fördern Unterstützung und Teilnahme durch den Kunden. Deshalb hat sich der agile Ansatz zu einer allumfassenden Betrachtungsweise der Softwareentwicklung in der Softwarebranche entwickelt und ist selbst zu einer eigenen Industrie geworden.