



Das agile Manifest: Der Einzelne und sein Team planen zusammen

Agile Methoden sind ab Anfang der 1990er Jahre in der Softwareentwicklung aufgekommen und erlangten um 1999 erste Popularität mit dem Extreme Programming (Kent Beck). Die Bedeutung agiler Methoden nimmt in den letzten Jahren stark zu. Dies gilt vor allem – aber nicht mehr ausschließlich – für den IT-Sektor: 62% der Softwareentwicklungsprojekte arbeiten in Deutschland teils agil, 16% sogar durchgängig, während lediglich 22% aller Projekte ausschließlich mit konventionellen Methoden arbeitet.

Die agile Philosophie betont die Fähigkeit des Einzelnen und die Selbstorganisation der Entwicklungsteams. Mittels „Agiler Methoden“ wird versucht, die Unzulänglichkeiten der vorherigen Softwareentwicklungsmodelle aufzuheben, die als zu starr betrachtet werden. Weil eine starre und durchgehende Planung nicht mehr praktikabel ist, stehen die Anforderungen für die Software nicht von Anfang an fest. Sie entwickeln sich stattdessen mit dem wachsenden Verständnis des Softwaresystems, was vor allem durch eine intensive Kommunikation mit den Kunden erreicht werden soll. Um die Kommunikation zu erleichtern, nutzt u.a. Extreme Programming zur Verständigung sogenannte „User Stories“, also Anwendererzählungen zur Beschreibung von Systemeigenschaften bzw. wesentlichen Funktionen. Planung ist in Kooperation durchzuführen und keine separate Entscheidung des Projektmanagers. Das Zusammenspiel einer Gruppe zur Planung des Umfangs der nächsten (Software-)Freigabe oder Bereitstellung wird als „Planning Game“ bezeichnet. Planning Games und die Schätzung des Projektaufwandes werden zyklisch wiederholt (iterativer Prozess).

Bei einem Treffen in Utah 2001 ist der Begriff agil für diese Art der Softwareentwicklung gewählt und das „agile Manifest“ (www.agilemanifesto.org/iso/de/manifesto.html) formuliert worden: **„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** stehen über Prozessen und Werkzeugen
- **Funktionierende Software** steht über einer umfassenden Dokumentation
- **Zusammenarbeit mit dem Kunden** steht über der Vertragsverhandlung
- **Reagieren auf Veränderung** steht über dem Befolgen eines Plans.“

Agile Prinzipien sind Leitsätze für die agile Arbeit. Manchmal werden agile Prinzipien auch als Methode bezeichnet. **Im Agilen Manifest sind zwölf Prinzipien aufgelistet.**¹

- Zufriedenstellung des Kunden durch frühe und kontinuierliche Auslieferung von wertvoller Software
- Agile Prozesse nutzen Veränderungen (selbst spät in der Entwicklung) zum Wettbewerbsvorteil des Kunden.
- Lieferung von funktionierender Software in regelmäßigen, bevorzugt kurzen Zeitspannen (wenige Wochen oder Monate)
- Nahezu tägliche Zusammenarbeit von Fachexperten und Entwicklern während des Projektes (Bsp.: Gemeinsamer Code-Besitz (Collective Code Ownership))
- Bereitstellung des Umfeldes und der Unterstützung, welche von motivierten Individuen für die Aufgabenerfüllung benötigt wird
- Informationsübertragung nach Möglichkeit im Gespräch von Angesicht zu Angesicht
- Als wichtigstes Fortschrittsmaß gilt die Funktionsfähigkeit der Software

- Einhalten eines gleichmäßigen Arbeitstempos von Auftraggebern, Entwicklern und Benutzern für eine nachhaltige Entwicklung
- Ständiges Augenmerk auf technische Exzellenz und gutes Design
- Einfachheit ist essenziell
- Selbstorganisation der Teams bei Planung und Umsetzung
- Selbstreflexion der Teams über das eigene Verhalten zur Anpassung im Hinblick auf Effizienzsteigerung

Das Prinzip des nachhaltigen Tempos: Keine Überstunden



Eines der zwölf agilen Prinzipien ist das des nachhaltigen Tempos (sustainable pace). Bei Beck gibt es die „40-hour week“ (40-Stunden-Woche) als eine der zwölf Praktiken des Extreme Programmings. Überstunden sind ein Hinweis auf

ernste Probleme im Projekt, die zu lokalisieren und zu lösen sind. Denn aufgrund ständiger Überstunden sind Entwickler kaum mehr in der Lage, klar und übersichtlich zu codieren. Sie werden unproduktiv und programmieren mehr Fehler. Roman Pichler als ein prominenter Vertreter schreibt entsprechend: In Scrum – der am weitesten verbreiteten agilen Methode - sind regelmäßige Überstunden inakzeptabel.

»Agile Softwareentwicklung ist ein Sammelbegriff für eine Reihe von Methoden und Praktiken, die auf Werten und Prinzipien des Manifests Agiler Softwareentwicklung basieren.« (Agile Alliance, 2018)

Im **Scrum**, der am weitesten verbreiteten agilen Vorgehensweise, gibt es die Rollen Product Owner, Team und Scrum Master. Der Product-Owner nimmt die Sicht des Kunden ein und beschreibt die Anforderungen, die zur Erreichung des Projektziels umgesetzt werden müssen, im sogenannten Product Backlog. Dieses Dokument bearbeitet er kontinuierlich und priorisiert die Anforderungen abhängig vom Gesamtfortschritt.

Die Anforderungen werden in sog. stets gleich lange Sprints (2-4 Wochen) umgesetzt. Im Sprint Planning verständigt sich das Entwicklungsteam mit dem Product Owner über die Anforderungen. Das Team wählt diejenigen Aufgaben aus, die es im nächsten Sprint bearbeitet. Es schätzt die damit verbundenen Aufwände ab, um zu gewährleisten, dass dies in dem gewählten Zeitraum des Sprints machbar ist.

Die Ergebnisse des Sprints werden im Sprint Review dem Product Owner (ggf. auch Kunden/Anwendern) vorgestellt. Als erledigt gelten ausschließlich getestete und ‚lauffähige‘ Funktionalitäten.

Das Team organisiert sich für den gesamten Prozess vollständig selbst und muss über die entsprechenden Ressourcen und Fähigkeiten verfügen. In der Retrospektive nach dem Sprint reflektiert das Team den Arbeitsprozess und die Zusammenarbeit, um Verbesserungsmaßnahmen festzulegen. Der Scrum Master als Experte für die Methode unterstützt das Team dabei, die selbst gesetzten Regeln einzuhalten und Hindernisse aus dem Weg zu räumen.