



Agile Softwareentwicklung – warum?

In der Zeit des stetigen Wachstums von neuen Softwareanwendungen (Mitte bis Ende der 90er Jahre des letzten Jahrhunderts) wurde die Erstellung von Software minutiös durchgeplant wie ein Ingenieurprojekt, z.B. eine Brücke. Daher konzentrierten sich viele Firmen unter Einsatz massiver Tools um eine intensive und aufwändige Planung aller Arbeitsschritte der Softwareerstellung und vor allem auch ihrer Dokumentation.

Im Gegensatz zu einer Brücke konnten sich viele Anwender jedoch nur schwer vorstellen, wie z.B. die bislang papiergestützte Buchhaltung auf einmal auf dem Rechner abgewickelt werden sollte. So hatten die Softwareentwickler oft die Speicherung und Verarbeitung der Daten im Fokus und konnten sich meist nicht in den Prozess der Bearbeitung durch den Sachbearbeiter hineinversetzen. In diesem Nebeneinander fokussierten sich die Entwickler auf ihre Tools und trieben einen erheblichen (zeitlichen und finanziellen) Aufwand, der den Kunden letztlich nicht interessierte. Konflikte waren vorprogrammiert.

Die fehlende Flexibilität, den Lernprozess des Kunden in den Entwicklungsprozess einfließen zu lassen, führte letztlich zu einer Reihe von neuen Ansätzen, die letztlich als agile Methoden in das Projektmanagement Eingang fanden.

2001 trafen sich führende Köpfe der Softwareentwicklung in Snowbird (Utah) und konnten sich letztlich auf das sogenannte Agile Manifest einigen, das in 12 Prinzipien mündete (<https://agilemanifesto.org/>):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Zu den agilen Verfahren zählten damals Scrum, Extreme Programming, Crystal Clear und andere.

Stellvertretend für diese agilen Methoden soll im Folgenden Scrum näher betrachtet werden.