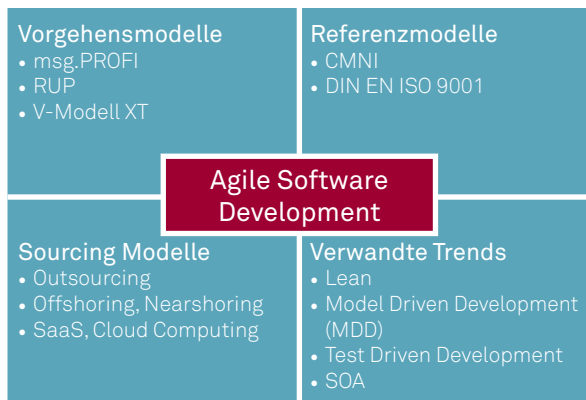


Agile Software-Entwicklung

Agile Software-Entwicklung ist der Oberbegriff für den Einsatz von Agilität in der Software-Entwicklung. Sie zeichnet sich durch schnelle Reaktion auf Veränderungen, effiziente Kommunikation mit allen Beteiligten, starke Einbeziehung des Kunden und Betonung sozialer Teamaspekte aus. Je nach Kontext bezieht sich der Begriff auf Teilbereiche der Software-Entwicklung (z.B. Agile Modeling) oder auf den gesamten Software-Entwicklungsprozess (z.B. Extreme Programming).

Definition

Die Agile Software-Entwicklung ist eine Gegenbewegung zu den oft als schwergewichtig und bürokratisch angesehenen traditionellen Software-Entwicklungsprozessen wie dem Rational Unified Process oder dem V-Modell. Dabei ist das Ziel, den Software-Entwicklungsprozess flexibler und schlanker zu machen, als das bei den klassischen Vorgehensmodellen der Fall ist. Man möchte sich mehr auf die zu erreichenden Ziele fokussieren und auf technische und soziale Probleme bei der Software-Entwicklung eingehen.



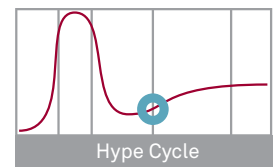
Prozessverbesserung an; alternative Sourcing-Modelle suchen neue Wege um die Beschaffungskosten von Software zu reduzieren.

Das msg-eigene Vorgehensmodell msg.PROFI (PROzessrahmen Für IT-Projekte) beinhaltet wesentliche agile Kernelemente (wie kurze Iterationszyklen, Kundentransparenz, Reaktionsfähigkeit auf kurzfristige Änderungen). Darüber hinaus unterstützt es Tailoring-Verfahren, risiko- und testgetriebene Entwicklung, systematische Codereviews sowie angemessene Qualität. msg.PROFI ist dabei kein Vertreter der vollständig agilen Methoden.

Agile Software-Entwicklung ist ein möglicher Ansatz, um Software-Entwicklung effizienter zu machen und dadurch Kosten und Time-to-Market zu reduzieren. Dazu legt Agile Software-Entwicklung besonderen Wert auf verstärkte Kooperation im Team -inklusive Kunde- und regelmäßige Validierung der Anforderungen anhand lauffähiger Software. Im Gegensatz dazu setzen z.B. Referenzmodelle à la CMMI bei der

Reifegrad

Agile Methoden werden bereits heute in „klassischen“ Projekten angewendet. Dabei werden einzelne agile Praktiken nutzbringend eingesetzt. Vollständig



Agile Methoden wie Scrum können in Kundenprojekten nicht einfach 1:1 angewendet werden. Es ist für den Projekterfolg entscheidend, zusätzliche bekannte Best Practices aus Software Engineering und Projektmanagement einzusetzen. Das Skalieren und Anpassen der Methoden auf die jeweiligen Gegebenheiten ist auch weiterhin ein entscheidender Erfolgsfaktor, der viel Erfahrung verlangt. In der Produktentwicklung kann Scrum prinzipiell mit weniger Anpassungen angewendet werden.

Marktübersicht

Scrum ist der bedeutendste Vertreter der rein agilen Methoden und deckt den Bereich agiles Projektmanagement ab. Darüber hinaus gibt es viele weitere Methoden, die jedoch im Kundenprojektgeschäft weniger Bedeutung haben.



Alternativen

Neben den klassischen – im agilen Kontext als „plan-driven“ bezeichneten – Vorgehensweisen gehen viele Kunden den Weg, agile und Plan-getriebene Methoden zu kombinieren. Neben **Hybrid-Methoden** wie Service-Oriented Software Development existieren hierfür verschiedene Ansätze, um kontext- und projektspezifische Vorgehensweisen zu entwickeln.

Referenzszenario

Ein typisches Einsatzgebiet für vollständig agile Methoden ist die Software-Produktentwicklung einer

neuen, sehr innovativen Lösung mit einer hohen Priorität auf Time-to-Market und sehr unklaren Anforderungen zu Beginn der Entwicklung. Dagegen sprechen die folgenden Kennzeichen eines Projektes für eher plangetriebene Elemente in der Vorgehensweise:

- Verteilte oder sehr große Teams
- Regulatorische Anforderungen an den Entwicklungsprozess (z.B. automobiler Steuergeräte)
- Hohe fachliche und/oder organisatorische Komplexität
- Unzureichender Reifegrad (z. B. Skill) bei Projektbeteiligten und Organisation

Business Impact

Ob und in welchem Maße der Einsatz agiler Methoden in der Software-Entwicklung eine Reduktion von Kosten und Verbesserung von Time-to-Market bewirkt, hängt vom jeweiligen Projektkontext ab. Darüber hinaus konnten weitere Vorteile, wie eine verbesserte Manövrierfähigkeit, in der Produktentwicklung ein-drucksvoll verwirklicht werden.

Pro	Contra
Verbesserte Manövrierfähigkeit (schnelle Reaktion auf veränderte Marktbedingungen, zügige und kostengünstige Umsetzung)	Leicht zu erlernen, aber schwer zu beherrschen; agile „Missverständnisse“ bei unerfahrenen Teams können zu erheblichen Problemen führen
Regelmäßige Überprüfung und Anpassung der Ziele (Anpassungen und Umpriorisierung der Anforderungen ist integraler Bestandteil)	Geringer Nutzen, wenn Anforderungen im Vorfeld bereits vollständig erhoben; im Festpreis-Kontext problematisch
Hoher „Takt“ in der Entwicklung (hohe Motivation im Team, effektive Kommunikation, Konzentration auf wertschöpfende Tätigkeiten, kontinuierliche Verbesserung)	Eher Entwickler-zentriert als Anwender- oder Endbenutzer-zentriert; gutes Produktdesign ist nicht im Fokus; Kreativität wird durch hohe Taktung behindert

msh systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München
 Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113
 www.msh-systems.com | info@msh-systems.com

