

Agiles V-Modell - ein Widerspruch?

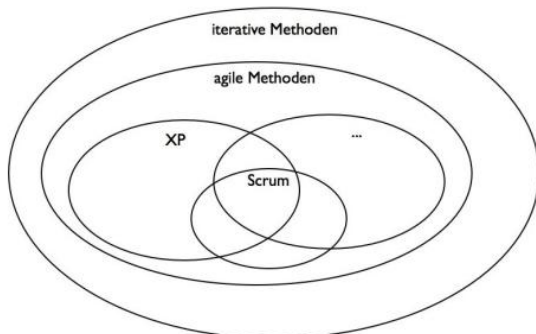
› Die agile Entwicklung (Agile Development) und das V-Modell XT gelten unter Kosten- und Qualitätsaspekten jeweils als effiziente Wege der Softwareerstellung. Die Frage liegt nahe, ob und wo sich beide Vorgehensweisen kombinieren lassen.

Von Ingo Kriescher und Josef Markgraf (16.07.2009 10:00:00)

Angesichts sinkender IT-Budgets steht die Softwareentwicklung auf dem Prüfstand. Diverse Studien unter anderem von [Gartner](#)¹ zeigen auf, dass 80 Prozent der IT-Budgets für Wartung ausgegeben und lediglich 20 Prozent in wettbewerbsdifferenzierende Lösungen investiert werden. Dabei sind laut Chaos Report der [Standish Group](#)² nur 32 Prozent der Projekte erfolgreich (termingerecht, budgetgerecht, volle Funktionalität), 44 Prozent sind mit Problemen (Verspätung, Budgetüberschreitung und/oder Fehlen versprochener Funktionalität) behaftet, und 24 Prozent schlagen fehl.

Einige Anwender sehen ihr Heil in [Standardsoftware](#)³, oft zu Lasten der Differenzierung von der Konkurrenz und auch des Budgets, wenn man dann doch nicht beim vermeintlichen Standard bleiben kann. Als Alternative zu ausufernden IT-Entwicklungsbudgets, vor allem für die Innovation und Entwicklung von Produkten und Verfahren, sehen sich Anbieter [agiler und iterativer Verfahren](#)⁴ (Agile Development) - und das in Deutschland, dem Land der Erfinder des [V-Modells](#)⁵, das durch das V-Modell XT ein Revival erlebt.

Die Diskussion über agile Entwicklung versus V-Modell wird dabei oft emotional geführt, vielfach auch aus Unwissenheit. Einerseits vergleicht diese Debatte unter Umständen Äpfel mit Birnen, da es das eine agile Vorgehensmodell nicht gibt. Bei Agilität handelt es sich vielmehr um ein Wertesystem, eine Projektkultur und um eine Familie von [Vorgehensmodellen](#)⁶, deren Werte und Prinzipien im [Manifesto for Agile Software Development](#)⁷ niedergeschrieben sind.



Agile Vorgehensmodelle

Andererseits lässt sich eine Gegenüberstellung vertreten, da auch das [V-Modell XT](#)⁸ ein "Baukasten" ist, der für jeden Projektkontext ein entsprechendes "Tailoring" benötigt und einen Leitfaden darstellt. Diese Flexibilität ermöglicht in einem innovativen und mutigen Umfeld eine erfolgversprechende Verknüpfung der beiden Vorgehensarten. Geschickt kombiniert, können sich die beiden mitunter konträren Ansätze ergänzen, so dass sich knappe Budgets effizienter nutzen lassen. Gerade bei komplexen, schwer beschreibbaren Aufgaben und in einem entsprechenden organisatorischen Umfeld kann die Kombination eine Lösung erst ermöglichen.

» Agile Development kontra V-Modell: die Unterschiede

Beide Ansätze zur Entwicklung von Software haben das Ziel, Projekte erfolgreich und kostengünstig umzusetzen, sie folgen dabei jedoch mitunter komplett unterschiedlichen Vorgehensweisen und Grundsätzen. Das V-Modell XT ist ziel- und ergebnisorientiert, im Mittelpunkt stehen die Produkte. Damit gemeint sind alle zu erstellenden Artefakte, die durch das Tailoring zu Projektbeginn vorgegeben wurden. Als Ergebnis erhält das Projekt einen Handlungsrahmen, der eindeutig definiert, in welcher Reihenfolge die Produkte erzeugt und abgesichert werden.

Dementgegen zeichnet sich ein agiles Vorgehensmodell durch Werte und Prinzipien aus. Die Handlungen der Projektbeteiligten werden durch dieses Wertesystem angeleitet, seinen Leitsätzen folgen auch die konkreten Praktiken. Unter anderem folgende Eigenschaften bilden den methodischen Kernansatz:

- » Iteratives und inkrementelles Vorgehen innerhalb fester "Time Boxes".
- » Steuerung durch den Kundennutzen: Die Umsetzung der fachlichen Anwendungsfälle steht im Vordergrund und stellt den Projektfortschritt dar.
- » Risikoorientierung: Das Vorgehen orientiert sich stets an den schwerwiegendsten Risiken und versucht diese frühzeitig zu beheben.
- » Ständiges Feedback der Kunden und Anwender ist die Basis einer erfolgreichen Benutzung der Lösung.
- » Adaptive (rolling-wave) Planung hält das Projekt auf Kurs.

Damit werden die weitreichenden Unterschiede der beiden Vorgehensweisen deutlich, vergleichbar etwa mit dem Paradigmenwechsel beim Übergang von der [prozeduralen Programmierung](#)⁹ zur [Objektorientierung](#)¹⁰. Für konkrete Budget- beziehungsweise Projektplanungen ist es zudem wichtig, diese Unterschiede an Projektthemen zu spiegeln:

» Weniger Risiko und Kosten

Um eine wirtschaftliche Entwicklung zu gewährleisten, versucht das V-Modell XT, das Problem der hohen Fehlerkosten zu lösen, indem durch Redundanz und Dokumentation Fehler vermieden werden. Hierzu werden Dokumente erzeugt und abgesichert, bevor die eigentliche Lösung entsteht.

Im Fall des agilen Vorgehens werden die Fehlerkosten reduziert, indem sowohl Risiken frühzeitig verringert und im Softwaresystem abgesichert werden als auch das System so erstellt wird, dass sich Fehler kostengünstig und frühzeitig beheben lassen. Qualität ist ein Designkriterium. Die ständige Steuerung der fachlichen und technischen Qualität durch frühzeitiges kontinuierliches Prüfen, Testen und Integrieren gewährleistet den Projekterfolg. Die Kernarchitektur des Systems wird schrittweise innerhalb des Regelkreises (Iteration/Feedback) erstellt, zu einem frühen Zeitpunkt geprüft, kontinuierlich getestet und reift im Kern mit der Zeit.

» Requirement und Change

Anforderungen werden im V-Modell möglichst detailliert im Lastenheft erfasst und durch den Auftraggeber bestätigt. Jede nachfolgende Änderung ist dann als [Change Request](#)¹¹ dokumentiert und wird zu einem späteren Zeitpunkt in das Projekt eingesteuert. Das [Lastenheft](#)¹² wird in ein [Pflichtenheft](#)¹³ überführt, es folgt die Erstellung der nächsten benötigten Dokumente. Die entstehenden Produkte werden unter anderem von den Testern geprüft, um die Testprozeduren erstellen zu können. Die linke Seite des V-Modells verbindet sich mit der rechten (siehe Grafik).



Struktur der Systemerstellung

Das heißt zum Beispiel, dass die Tester während der Anforderungsanalyse nicht nur in die Reviews der Anforderungsdokumente eingebunden werden, sondern auch, dass sie bereits zu diesem Zeitpunkt die Testprozeduren und Testdokumente erstellen sollen. Dies trifft auf allen Ebenen des V-Modells zu.

Beim agilen Vorgehen werden die Anforderungen evolutionär erfasst und auf Basis von "Funktionalität-nach-Funktionalität" (Userstory-by-Userstory) in Iterationen umgesetzt. Ein Feature stellt hierbei die für den Kunden nutzbringende Anforderung dar, das heißt, die Anforderungen müssen dementsprechend auch strukturiert sein. Der Auftraggeber wird während der kompletten Projektlaufzeit stark eingebunden, damit existierende Anforderungen detailliert und neue beziehungsweise geänderte

Anforderungen gemeinsam in eine der nächsten Iterationen eingesteuert werden können. Die Anforderungen werden zum Beispiel als Tests detaillierter formuliert und unnötige Zwischenartefakte vermieden. Das Team arbeitet gemeinsam an der Umsetzung und Absicherung dieser Funktionen, redundante, mitunter temporäre Dokumente werden nicht benötigt. Auf diesem Weg lassen sich Fehler kostengünstig entfernen, und der Auftraggeber bestätigt die korrekte Umsetzung der Anforderung. Das verhindert schon in frühen Projektphasen Missverständnisse und erspart teure Umbauten.

» Projektsteuerung

Die jeweilige Projektsteuerung gestaltet sich entsprechend den bis jetzt beschriebenen Unterschieden. So stellt das V-Modell den Rahmen, in dem sich das Projektteam bewegen muss, mit klaren Abgrenzungen, wer wofür verantwortlich ist. Die Projektplanung wird im Detail durch den Projektleiter erstellt und vom Projektteam abgearbeitet. Frei nach dem Motto: Plan the Work - work the Plan. Abweichungen vom Plan werden als Fehler wahrgenommen und bekämpft, denn der Plan wird ungenügend angepasst.

Im Rahmen der agilen Softwareentwicklung wird dagegen eine adaptive Planung verwendet. Man plant auf verschiedenen Ebenen, und nur für die nahe Zukunft ist die Planung sehr detailliert. Nach jeder Iteration wird die Zielerreichung geprüft und der weitere Projektlauf nachjustiert. Dem Motto folgend: Der Plan ist nichts - Planung ist alles. Hierdurch ist es möglich, Änderungen wohlwollend in die Planung aufzunehmen, umzusetzen und das Projekt in Bezug auf Anforderungen oder unerwartete Ereignisse beweglich zu halten.

» Kundeneinbindung

Der letzte Gegensatz, der hier berücksichtigt werden soll, ist die Art und Weise, wie der Kunde (Auftraggeber) eingebunden wird und wie das Verhältnis zwischen Auftraggeber und Auftragnehmer gestaltet ist. Das V-Modell XT ist explizit für das Erstellen von Gewerken konzipiert worden. Die Vergabe von Dienstleistungen deckt es nicht ab. Eine der Neuerungen des V-Modells XT vertieft daher auch die konkrete Beschreibung der Zusammenarbeit von Auftraggeber und -nehmer und bezieht sich auf die zu erstellenden Produkte und deren Abnahme.

Auf der anderen Seite sind agile Vorgehensweisen für eine enge Kooperation von Auftragnehmer und Auftraggeber konzipiert und bevorzugen Dienstleistungsverträge. Die Wahl der Vertragsform und die Regelung der Zusammenarbeit durch den Auftrag ist nicht Bestandteil des agilen Vorgehens an sich. Der Ansatz sieht aber immer eine vertrauensvolle Zusammenarbeit vor, die weit über das Erstellen von Dokumenten und ihrer Reviews hinausgeht. Eine regelmäßige Einbindung des Auftraggebers beim Detaillieren der Anforderungen, der Planung und den Iterationsbewertungen wird vorausgesetzt: Der Auftraggeber muss zu einem guten Anteil mitarbeiten.

» V-Modell und Agile Dev. - kombiniertes Vorgehen

Ist eine Verbindung von V-Modell und agiler Entwicklung sinnvoll und nützlich? Diese Frage ist mit "Ja, aber..." zu beantworten. Die Ansätze schließen sich nicht aus, und im V-Modell XT ist bereits ein erster verbindender Schritt getan worden. Aber nicht für alle Projekte und in allen Unternehmen können agile Methoden mit dem V-Modell XT auf allen Ebenen sinnvoll kombiniert werden, da das agile Modell ein gewisses Maß an Vertrauen und Mut voraussetzt. Insbesondere bei unternehmenskritischen Systemen, die als Gewerk zum Festpreis ausgeschrieben sind, wird eine nutzbringende Kombination schwerer umsetzbar sein.

Ohne größere Probleme und Zweifel lassen sich dagegen einige konkrete agile Praktiken direkt in das V-Modell XT übernehmen, da es hier keine zwingende Richtlinie gibt. Unter anderem können dies das Konzept der [Test Driven Development](#)¹⁴ (TDD) auf Entwicklerebene, der kontinuierlichen [Builds](#)¹⁵ des Softwaresystems und der automatisierten Überprüfung der Lösung sein. Auch [Pair-Programming](#)¹⁶ wird in jedem Umfeld seinen Wert zeigen und ist nicht nur bei agilen Methoden hilfreich.

Ein besonderer Nutzen aus dem kombinierten Vorgehen ergibt sich bei der Lösung komplexer Probleme, die nicht einfach beschreibbar sind - und zwar sowohl für den Auftragnehmer als auch für den Auftraggeber, der eventuell sogar zur Verwendung des V-Modells XT verpflichtet ist. Bei dieser Symbiose bildet das V-Modell den vertrauten Rahmen für die Projektorganisation des Auftraggebers. Alle Ansprechpartner und Stellen innerhalb der Organisation können weiterhin bedient werden, und die Arbeitsteiligkeit der Organisation bleibt gewährleistet.

Das Gesamtprojekt sollte zudem, je nach Projektgröße, innerhalb des V-Modells in mehreren Projektstufen (Releases) geplant werden. Dabei dient die erste Stufe dazu, die Architektur zu stabilisieren, die größten Risiken zu entschärfen und das adaptierte Vorgehen zu erproben. Bei einem passenden Rahmenvertrag für

die Vergabe der weiteren Projektstufen können Aspekte der evolutionären Anforderungsanalyse einfließen und somit das Projektrisiko inklusive der Kosten reduziert werden.

Die detaillierte Verbindung wird über die agile Projektdurchführungsstrategie erstellt. Innerhalb der angepassten Strategie wird entsprechend dem agilen Vorgehen die Lösung umgesetzt. Hierbei kann im V-Modell "Iteration geplant" als Einstiegspunkt dienen. Die Einbindung über die Durchführungsstrategie und eine dokumentierte Abweichung sollte jedoch in enger Abstimmung mit dem Auftraggeber stattfinden. Die Effizienz der Entwicklung lässt sich durch die Verwendung der agilen Prinzipien steigern, und Änderungswünsche lassen sich leichter in das Projekt einsteuern. Allerdings wird mit zwei unterschiedlichen kulturellen und methodischen Ansätzen gearbeitet, und die dadurch auftretenden Reibungsverluste müssen durch die Schnittstellen zwischen den Welten kompensiert werden. Zudem sind eine intensivere und kontinuierlichere Mitarbeit des Auftraggebers, eine empirische Projektsteuerung und ein modifiziertes Änderungs-Management erforderlich.

Für den Auftraggeber bedeutet das eine wesentlich höhere Projekttransparenz, weshalb es sich nur vermeintlich um Mehrarbeit handelt. Zudem kann der Projektstatus nach messbaren Fortschritten der Lösung, ausführbaren Tests sowie dem Maßstab einer früheren und kontinuierlicheren Integration beurteilt werden - und dies zu geringeren Kosten.

» Fazit

Bei innovativen Anwendungen, die sich im Vorfeld nur schwer beschreiben lassen, führt ein gemeinsam mit dem Auftraggeber verfolgter Lösungsansatz, bei dem agile Methoden in das etablierte V-Modell eingebettet werden, zu zeitnahen und kostengünstigen Lösungen. Das kann auch dort sinnvoll sein, wo aus Zeitnot eine Unterspezifikation in Kauf genommen werden musste und den Beteiligten bei der Vergabe klar war, dass die Aufgabe nur in enger Zusammenarbeit zu lösen sein würde. Der iterative Entwicklungsansatz lässt die Projektbeteiligten interdisziplinär denken. Auch beim Auftraggeber kommt es zu einem breiten Kompetenzaufbau, der ihn auf lange Sicht entlastet sowie Risiken und damit Kosten senkt.

¹ <http://www.gartner.com/>

² <http://www.standishgroup.com/>

³ <http://de.wikipedia.org/wiki/Standardsoftware>

⁴ http://de.wikipedia.org/wiki/Agile_Softwareentwicklung

⁵ <http://www.v-modell.iabg.de/>

⁶ http://de.wikipedia.org/wiki/Vorgehensmodell_%28Software%29

⁷ <http://agilemanifesto.org/>

⁸ <http://www.scribd.com/doc/2089960/VModell-XT>

⁹ http://de.wikipedia.org/wiki/Prozedurale_Programmierung

¹⁰ <http://de.wikipedia.org/wiki/Objektorientierung>

¹¹ <http://de.wikipedia.org/wiki/%C3%84nderungsanforderung>

¹² <http://de.wikipedia.org/wiki/Lastenheft>

¹³ <http://de.wikipedia.org/wiki/Pflichtenheft>

¹⁴ http://de.wikipedia.org/wiki/Testgetriebene_Entwicklung

¹⁵ http://en.wikipedia.org/wiki/Software_build

¹⁶ <http://de.wikipedia.org/wiki/Pairprogrammierung>

IDG Business Media GmbH

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium in Teilen oder als Ganzes bedarf der schriftlichen Zustimmung der IDG Business Media GmbH. DPA-Texte und Bilder sind urheberrechtlich geschützt und dürfen weder reproduziert noch wiederverwendet oder für gewerbliche Zwecke verwendet werden. Für den Fall, dass in Computerwoche unzutreffende Informationen veröffentlicht oder in Programmen oder Datenbanken Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht. Die Redaktion übernimmt keine Haftung für unverlangt eingesandte Manuskripte, Fotos und Illustrationen. Für Inhalte externer Seiten, auf die von Computerwoche aus gelinkt wird, übernimmt die IDG Business Media GmbH keine Verantwortung.