

Modellierung – agil versus traditionell

Andreas Ditze, Susanne Mühlbauer, Philip Stolz

Genereller Einsatz von Modellierung

Eine wesentliche Eigenschaft eines jeden Modells ist, dass es immer nur einen Teil der Wirklichkeit abbildet, indem es ausgewählte Aspekte darstellt, andere jedoch gänzlich weglässt. Diese Eigenschaft hilft uns dabei, mit Modellen gezielter kommunizieren zu können und durch bewusstes Weglassen bzw. Darstellen von Aspekten den Überblick zu behalten.

Abhängig vom Formalisierungsgrad und der Abstraktionsebene können wir Modelle mit unterschiedlichen Zielsetzungen einsetzen. Unterschiedliche Abstraktionsebenen ermöglichen es uns, sowohl den Überblick zu behalten, als auch dabei, den Problemraum vom Lösungsraum – das Was und Warum vom Wie – zu trennen.

Gibt es überhaupt eine Agile Modellierung?

Um diese Frage zu beantworten, ist es unerlässlich, einen Blick in das agile Manifest zu werfen. Hier sind Werte und Prinzipien beschrieben, die bei agilem Vorgehen zu beachten sind. Ein genaues Vorgehensmodell oder eine Beschreibung, wie diese Werte und Prinzipien erreicht und eingehalten werden können, finden wir dort nicht. Etabliert haben sich Praktiken wie z.B. Scrum, die agiler Entwicklung einen Rahmen geben. Aber auch dort ist das konkrete „Wie“ bewusst offen gehalten. Sowohl das agile Manifest als auch Scrum treffen keine Aussage zu Modellierung, lassen diese dadurch natürlich implizit zu. An dieser Stelle möchten wir beispielhaft einige Prinzipien des agilen Manifests herausgreifen, die durch Modellierung sinnvoll unterstützt werden können.

Kommunikation und Konversation – ein Bild sagt mehr als tausend Worte

„Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht (Agiles Manifest).“ Agile Entwicklung versucht auf umfassende schriftliche Vorabspezifikationen zugunsten von Konversation und schnellem Feedback zu verzichten.

Dieses Prinzip basiert also auf dem persönlichen Gespräch zwischen Fachbereich/Kunde und Entwicklung, um das Problem genau zu erfassen (das „Was“) und ebenso auf persönlicher Kommunikation innerhalb der Entwicklung, um das Software-Design (das „Wie“) als

Basis der Implementierung zu erarbeiten. Modelle können, je nach Zielsetzung, mit unterschiedlichen Formalisierungsgraden eingesetzt werden. Für Kommunikation und Konversation bietet es sich an, Modelle gemeinsam in Form einer Skizze am Flipchart zu entwerfen, um ein einheitliches Verständnis zu erzielen.

Existieren bereits Modelle, so können die betroffenen Aspekte ausgedrückt und ebenfalls gemeinsam verändert und erweitert werden. Dieses Vorgehen eignet sich für Analyse- und Design-Aufgaben gleichermaßen.

Mit den erstellten Skizzen kann nun je nach Kontext unterschiedlich verfahren werden:

- Sie werden entsorgt, nachdem das gemeinsame Verständnis erzielt wurde
- Sie werden in eine elektronische Form übertragen (z.B. in ein Modellierungswerkzeug), die eine weitere Bearbeitung der Modelle zulässt:
 - Auswirkungsanalyse der betrachteten Teilaspekte auf das Gesamtsystem (falls dieses bereits als Modell repräsentiert vorliegt)
 - Ableitung des Software-Designs durch das Entwicklungsteam
 - Implementierungsvorgabe für modellbasierten Code
 - Verfeinerung und Aktualisierung als Systemdokumentation vor/während/nach der Implementierung

Weniger ist mehr - Das Richtige richtig tun

„Einfachheit -- die Kunst, die Menge nicht getaner Arbeit zu maximieren -- ist essenziell (Agiles Manifest).“

Dieses Prinzip schreit geradezu nach Modellierung. Modellgetriebene Vorgehensweise hilft dabei Routinearbeiten zu sparen durch:

- Code-Generierung aus Design-Modellen
- Dokumentations-Generierung aus allen Modellen
- Erstellung von Benutzerdokumentation aus ausgewählten Modellen
- modellbasiertes Testen

In Verbindung dazu sind auch die folgenden beiden Prinzipien zu sehen:

„Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen (Agiles Manifest).“

„Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne. (Agiles Manifest).“

Diese beiden Forderungen lassen sich nur erfüllen, wenn sich die Entwicklung auf die wirkliche Entwicklungsleistung und Innovation konzentrieren kann und Änderungen am Gesamtsystem jederzeit schnell und ohne riskante, unvorhersehbare Auswirkungen durchgeführt werden können.

Was haben wir Entwickler davon?

Ein letztes der Prinzipien möchten wir gerne herausgreifen:

„Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams (Agiles Manifest).“

In der Praxis sehen wir leider, dass agile Teams kaum mit Modellierung arbeiten. Sie erkennen häufig den Nutzen nicht, der Ihnen durch Modellierung entstehen kann – Modellierung wird als Aufwand, nicht als Entlastung angesehen. Die Erklärungsversuche hierfür sind unterschiedlich:

- Für die Modellierung gibt es eine eigene Abteilung, es findet keine unmittelbare Zusammenarbeit mit den Teams statt
- Die existierenden Modelle sind veraltet und bilden nicht den realen Stand des Systems ab
- Die Tools sind nicht nahe genug an der Entwicklungsumgebung
- Es ist zu wenig Modellierungs-Know-how im Team vorhanden
- Modellierung wird als kompliziert empfunden, da immer der höchste Formalisierungsgrad angestrebt wird

Abschließend betrachtet, sind Methoden und Tools zur Modellierung also auch in einem agilen Umfeld sinnvoll einsetzbar. Es gibt daher keine „agile Modellierung“ im eigentlichen Sinne, sondern vielmehr eine agile Verwendung von Modellierung unter Berücksichtigung agiler Prinzipien und agiler Werte.

Was macht traditionelle Modellierung traditionell?

Prinzipiell unterscheiden wir im Folgenden die Phasen Analyse, Design und Implementierung. In der Analysephase machen wir uns Gedanken, Was getan werden muss und Warum (Ziel), in der Phase Design legen wir fest, Wie wir das Ziel erreichen und in der Phase Implementierung erfolgt die eigentliche Umsetzung.

Bei eher traditionellen, phasenorientierten Vorgehensweisen werden die einzelnen Phasen sequentiell bearbeitet (siehe Abbildung 1), wobei auch hier ein iterativer Ansatz natürlich nicht ausgeschlossen ist. Auch die Modellierungsaktivitäten richten sich dann nach diesen Phasen, wir unterscheiden zum Beispiel nach objektorien-



Susanne Mühlbauer ist Senior Consultant, Trainer und Coach bei der HOOD Group.



Andreas Ditze ist Geschäftsführer bei der MID GmbH. Er ist Leiter der Fachgruppe Modellierung im ASQF.



Philip Stolz ist Senior Consultant, Trainer und Coach bei der HOOD Group.

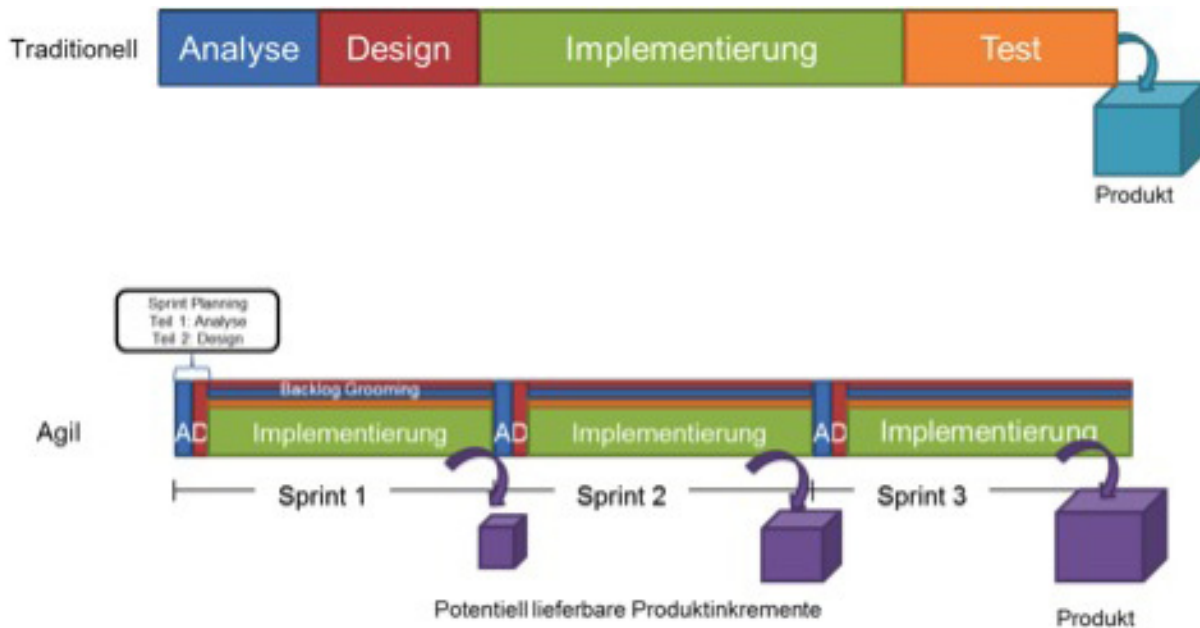


Abbildung 1: Gegenüberstellung agiler und traditioneller Ansätze

tierter Analyse (OOA) und objektorientiertem Design (OOD). Modellierungsnotationen wie die UML sind universell für alle Phasen und Abstraktionsebenen einsetzbar oder beziehen sich auf die Beschreibung konkreter Artefakte, wie die BPMN für Geschäftsprozesse und Workflows.

Agiles Vorgehen ist iterativ und inkrementell. Analyse, Design und Implementierung sind keine Phasen, sondern Aktivitäten, die laufend stattfinden. Das Ergebnis jeder Iteration (hier Sprint) ist jeweils ein potentiell lieferbares Produktinkrement. Im Rahmen der Sprintplanungsmeetings am Anfang jeden Sprints erfolgt Analyse und Design für die im aktuellen Sprint umzusetzenden Anforderungen. Während des Sprints erfolgen Analyse- und Designaktivitäten für künftige Sprints im Rahmen des Backlog Groomings.

Die Wahl der Modellierungsnotation ist hier also nicht abhängig von der Phase, sondern vielmehr von Aktivität, Einsatzzweck und erforderlichem Formalisierungsgrad.

Abbildung 1 stellt den Unterschied zwischen agiler und traditioneller Vorgehensweise als Prinzipdarstellung gegenüber. Der Vollständigkeit halber haben wir die Abbildung um die Phase bzw. Aktivität Testen erweitert.

Fazit

Für uns – und wir hoffen, wir konnten unsere Gedankengänge nachvollziehbar darstellen – sind Modellierungskennnisse für die professionelle Software-Entwicklung unabdingbar. Unabhängig davon, ob Sie agil oder traditionell vorgehen bietet Modellierung ausgezeichnete Möglichkeiten, um unter anderem

- ein gemeinsames Verständnis zu schaffen
- Problem- und Lösungsdarstellungen voneinander zu trennen
- Komplexität sichtbar und beherrschbar zu machen
- Routinetätigkeiten zu Automatisieren (Code-Generierung)
- Analyse-/Design-Entscheidungen und die erstellte Implementierung zu dokumentieren.

Für den Einsatz von Modellierung ist es dabei vollkommen unerheblich, welches Vorgehensmodell für die Systementwicklung gewählt wird.

In allen Phasen eines Produktlebenszyklusses kann Modellierung eingesetzt werden. Wie viel und mit welchen Notationen modelliert wird, ist vielmehr abhängig vom Kontext und vom Einsatzzweck.

Heute kommen überwiegend Stift und Flipchart zum Einsatz, wenn es um das Thema Modellierung in agilen Projekten geht. Dies dient dem Schaffen eines Verständnisses im Team, hat jedoch den Nachteil, dass viele Potentiale der Modellierung ungenutzt bleiben. In traditionell durchgeführten Projekten dagegen dominiert das Modell als Nachdokumentation der Implementierung. Besonders in agilen Projekten ist zu erwarten, dass sich werkzeuggestützte Modellierung erst dann durchsetzt, wenn Modellierungswerkzeuge und Entwicklungsumgebung derart integriert sind, dass Modell und Code zu jedem Zeitpunkt weiterentwickelt werden können und dabei automatisch konsistent zueinander bleiben.