

Software Management in turbulenter Umwelt – Warum IT-Projekte agil werden müssen!

Wandel der Vorgehensmodelle im Zeitalter der digitalen Transformation

Eckhart Hanser¹

Abstract: Der folgende Artikel wurde als vielbeachtete Keynote zum 25-jährigen Geburtstag der Fachgruppe Vorgehensmodelle für die betriebliche Anwendungsentwicklung (WI-VM) der Gesellschaft für Informatik e.V. an der Tagung PVM2018² in Düsseldorf gehalten. Er beschäftigt sich mit dem Wandel von Vorgehensmodellen in Software-Projekten aufgrund der beginnenden Digitalisierung um die Jahrtausendwende und beschreibt die Entstehung von agilen und hybriden Modellen als Reaktion auf sich schnell ändernde Projekt- und Produkt-Anforderungen im Zeitalter der digitalen Transformation von Geschäftsmodellen.

Keywords: Agile und hybride Vorgehensmodelle, Digitalisierung und digitale Transformation, Scrum, Kanban, MAP.

1 Einleitung – Historie der Vorgehensmodelle

Auf den großen GI-Tagungen zum Thema Vorgehensmodelle und Projektmanagement ist seit etwa 10 Jahren ein Umbruch in der Diskussion zu spüren. Stand z.B. 2008 noch die Industrialisierung des Software-Managements im Vordergrund [HM08] war 2013 bereits das agile Vorgehensmodell Scrum Gegenstand einer Keynote [HMF13]. Seitdem dominieren hybride Projektstrukturen und soziale Aspekte die Tagungen³. Was ist in der Zwischenzeit geschehen?

Die Basis für die Erklärung für den Vormarsch agiler Techniken, die allesamt auf das „Agile Manifest“⁴ zurückgehen, bildet die Analyse der IT-Projekte der jeweiligen Zeit.

Die 1970er Jahre wurden dominiert vom „Wasserfall-Modell“ [Fa85]. Dieses lineare Modell passt hervorragend zum Typ der damaligen Projekte. Man flog z.B. zum Mond oder schrieb ein neues Großrechner-Betriebssystem. Die Anforderungen waren wohldefiniert und unterlagen über die zum Teil mehrjährige Projektdauer keinen oder

¹ DHBW Lörrach, Kompetenzzentrum für agile IT-Prozesse (KAP) im Studienzentrum Informatik (SZI) (<http://www.dhbw-loerrach.de/kap.html>)

² Die PVM2018 fand vom 15.-16. Oktober 2018 an der FOM - Hochschule für Oekonomie und Management in Düsseldorf statt (siehe auch <http://www.pvm-tagung.de>).

³ Siehe z.B. in den Tagungsbänden der PVM-Tagung ab 2013, veröffentlicht in den Lecture Notes for Informatics, <https://dl.gi.de/handle/20.500.12116/21> .

⁴ Die agilen Grundprinzipien sind beschrieben unter <http://agilemanifesto.org/> .

nur wenigen Änderungen. Folglich bestand der Software-Entwicklungsprozess nur aus vier relevanten Phasen: Analyse des Problems, Design der Lösung, Implementierung (Kodierung) der Lösung und Systemtest. Danach fand eine „Wartungsphase“ statt, in der kleinere Anpassungen vorgenommen wurden. Interessant ist auch, dass in einer Phase zwar verifiziert wurde, ob die Resultate der vorhergegangenen Phase entsprachen, aber Änderungen der Produkte der vorherigen Phasen nicht vorgesehen waren [Fa85]. Aufgrund der „Stabilität“ der Projekte war dies auch nicht nötig. War doch einmal eine größere Anpassung notwendig, wurde einfach ein neues Wasserfall-Projekt aufgesetzt.

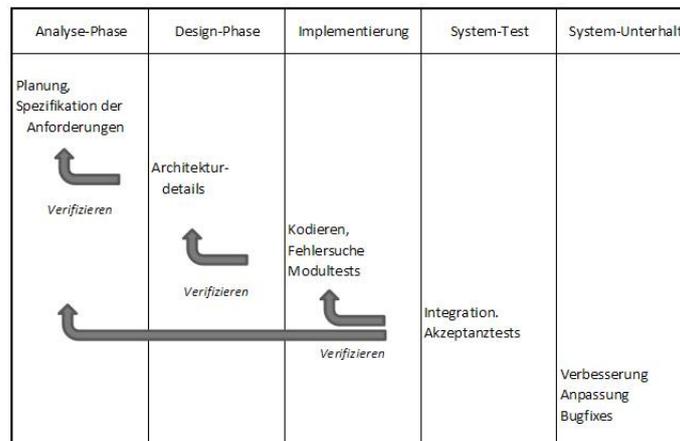


Abb. 1: Wasserfall-Modell [Ha10]

Ab 1979 setzte sich das V-Modell durch, welches in seiner ursprünglichen Form von Barry W. Boehm entwickelt wurde und die Zuordnung von Tests und Qualitätssicherungsmaßnahmen zum jeweiligen Level der Spezifikation ins Zentrum rückte⁵. Eine Iteration des Prozesses war allerdings in diesem ersten Ansatz immer noch nicht vorgesehen. Warum auch? Das V-Modell wurde hauptsächlich im „regulierten Umfeld“ eingesetzt, also z.B. in der Rüstung oder in der pharmazeutischen Industrie. Auch hier waren die Anforderungen in aller Regel stabil. Bis zur Jahrtausendwende änderte sich wenig. Die GI-Fachgruppe Vorgehensmodelle für die betriebliche Anwendungsentwicklung⁶ wurde 1993 gegründet und beschäftigte sich folgerichtig ab 1997 schwerpunktmäßig mit dem V-Modell⁷.

⁵ Eine sehr schöne Einführung in die Historie findet sich in [HH08], Kap.7.

⁶ Siehe auch <http://www.vorgehensmodelle.de>.

⁷ Empfohlen sei hier der Artikel von Ralf Kneuper zum 20-jährigen Jubiläum der Fachgruppe in [HMM13].

2 Digitalisierung und Digitale Transformation

Etwa um die Jahrtausendwende begann ein radikaler Umbruch: Die technischen Möglichkeiten der IT versetzten die Menschheit zum ersten Mal in die Lage, mehr Information digital als analog zu speichern. Üblicherweise spricht man vom Beginn des „Digitalen Zeitalters“ [HL11]. Damit konnte mehr als die Hälfte des Wissens der Menschheit informationstechnisch verarbeitet werden. Davor stand allerdings die Erfindung wichtiger technischer Voraussetzungen: die Erfindung der ersten Personal Computers wie Apple I (1976) oder IBM PC (1981) und die Entwicklung des „World Wide Web“⁸ durch Tim Berners-Lee. Durch die damit einhergehende (bezahlbare) Zurverfügungstellung dieser Technologien wurde die Digitalisierung überhaupt erst möglich.

Die einsetzende „Digitale Transformation“ ermöglichte bzw. erzeugte erhebliche Veränderungen des Alltagslebens, der Wirtschaft und der Gesellschaft durch den Einsatz der neuen digitalen Technologien. Neue IT-Produkte entstanden, wie auch neue Geschäftsmodelle. Exemplarisch seien hier Suchmaschinen wie Google, Webshops wie Amazon und ebay, sowie Social Media-Plattformen wie Facebook oder WhatsApp genannt. Um die Jahrtausendwende setzten auch immer Unternehmen auf das Internet als Möglichkeit des Verkaufs⁹.

Im Bereich der Software-Entwicklung und des Managements von IT-Systemen ergab sich durch die Digitale Transformation ein großes Problem: Immer mehr Projekte wurden unter immer größerem zeitlichen Druck angestoßen – z.T. ohne in der Lage zu sein, klare und vollständige Spezifikationen vorzugeben. Das Phänomen der „Moving Targets“ [Ha10] entstand.

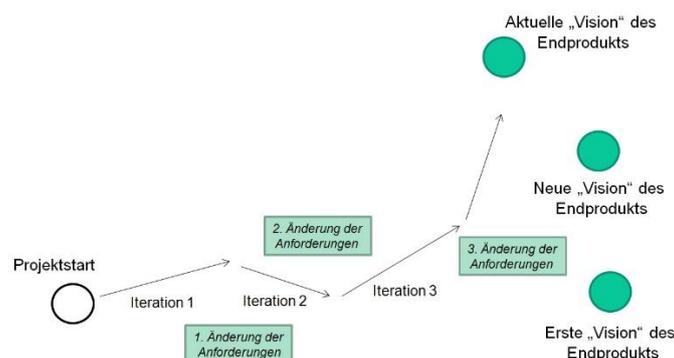


Abb. 2: Moving Targets – Problem der sich ständig ändernden Anforderungen

⁸ Tim Berners-Lee „erfand“ das WWW 1989 am CERN. Einen schönen Abriss der Geschichte findet man unter <https://webfoundation.org/about/vision/history-of-the-web/>.

⁹ In diesem Zusammenhang sei auf <http://www.online-redakteure.com/das-digital-vernetzte-ich/geschichte-der-online-shops/> verwiesen.

Neu war, dass meistens erst im Verlauf des Projekts die finale Vision des Produkts entstand. Vorgehensmodelle wie das V-Modell sind für solche Projekte ungeeignet!

3 Die Reaktion: Agile und hybride Vorgehensmodelle

Aus der Sicht des Autors ist es kein Zufall, sondern eine kausale Abhängigkeit, dass in etwa zeitgleich zur Digitalisierung das Agile Manifest entstand. Als Gegenpol zu „schwergewichtigen“, dokumentengetriebenen Modellen wie dem V-Modell steht die Interaktion und Kommunikation zwischen Kunden und Entwicklern im Mittelpunkt. Eine funktionierende Software ist wichtiger als ein Spezifikationsdokument. Die Reaktion auf sich ändernde Projektbedingungen steht im Fokus, nicht das plangetriebene Vorgehen¹⁰.

Ein weiterer interessanter Aspekt ist die „Sprengkraft“, die in der Digitalen Transformation steckt: Bestehende Geschäftsmodelle oder Märkte können zerschlagen werden, wie man an den Problemen des stationären Handels durch die Existenz des Online-Handels erkennen kann. Die Digitale Transformation kann „disruptiv“ sein¹¹. Damit stellen sich 2 Fragen:

1. Ist die digitale Transformation nicht nur für Geschäfts-, sondern auch für Vorgehensmodelle disruptiv?
2. Ersetzen (somit) agile Modelle die „schwergewichtigen“ Modellfamilien, wie z.B. das V-Modell?

Um Frage 1 zu beantworten, sollen im Folgenden zwei bekannte Vertreter der agilen Modelle analysiert werden:

¹⁰ Nicht umsonst dominieren bei der Postulierung des Agilen Manifests die Autoren der verschiedenen agilen Modelle. Sie leiten ihre Werte, Regeln und Praktiken direkt vom Agilen Manifest ab (siehe auch <http://www.agilemanifesto.org>).

¹¹ Siehe hierzu <http://www.gruenderszene.de>.

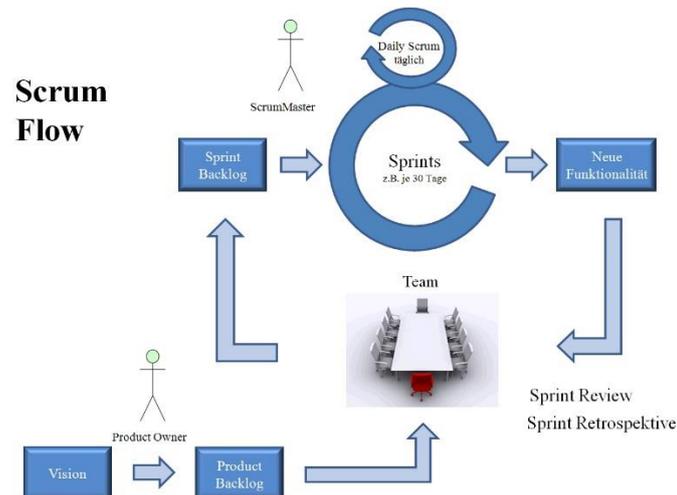


Abb. 3: Scrum

Das agile Vorgehensmodell Scrum [Sc04] ist klar disruptiv! Die neue Rolle des Product Owners, die Team-Rolle, die aus mehreren Entwicklern besteht und der ScrumMaster ersetzen zusammen die klassische Rolle des Projektleiters. Will eine IT-Entwicklungsorganisation Scrum einführen, müssen alle(!) Projektleiter „umgeschult“ werden. Ihre alte Aufgabe gibt es nicht mehr! Außerdem wird das Prinzip, Releases durch vorhandene Funktionalität des Software-Produkts zu definieren, umgestoßen: Releases (zumindest interne) entstehen einfach durch Beendigung eines zeitlich vordefinierten Sprints (Time-Boxing)¹². Zusammenfassend lässt sich sagen:

- Wer Scrum einsetzen will, muss neue Rollen einführen!
- Es gibt keinen „sanften“ Übergang von „Nicht-Scrum“ zu Scrum!
- Scrum schafft den Projektleiter ab!
- Scrum verändert den Anspruch an das Team!
- Scrum fordert Generalisten statt Spezialisten¹³.
- Scrum führt neue, kurze Dokumente ein und verzichtet auf aufwändige Spezifikationen.

Folglich fällt der Wechsel zu Scrum schwer!

Etwas „dezent“ zeigt sich die Disruption beim agilen Vorgehensmodell „Kanban“ [An11]. Kanban kennt keine Sprints, dafür aber einen kontinuierlichen Entwicklungsprozess ohne Time-Boxing durch Iterationen. Es gibt keine diskreten

¹² Für eine kurze Zusammenfassung von Scrum siehe auch [Ha10].

¹³ wenigstens meistens

Deployment-Termine, dafür dominiert die Methode der „kleinen Änderungen“. Für viele „Kanban-Tickets“¹⁴ wird der Aufwand und die Entwicklungszeit nicht mehr geschätzt, sondern eine (angestrebte) durchschnittliche Durchlaufzeit festgelegt.

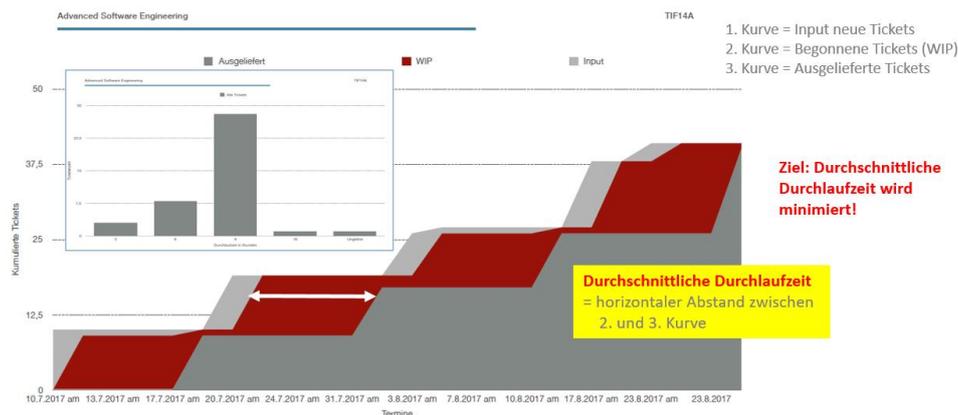


Abb. 4: Kanban – durchschnittliche Durchlaufzeit¹⁵

Somit ist auch Kanban disruptiv: Eine präzise Aufwandschätzung entfällt für die Serviceklassen „Standard“ und „unbestimmbare Kosten“ (Gelbe und grüne Tickets). Meilensteine (z.B. für Versionen) werden ersetzt durch statistische Aussagen zur Durchlaufzeit von Tickets (d.h. von den Arbeitspaketen). Der Fokus geht weg vom Projekt zur (Entwicklungs-) Organisation.

Somit kann zusammenfassend gesagt werden, dass die Digitale Transformation auch für Vorgehensmodelle disruptiv ist. Die Digitale Transformation schafft Umbrüche bei den Vorgehensmodellen durch sich schnell ändernde Anforderungen an Produkt und Team. Im Zeitalter der Digitalisierung müssen Vorgehensmodelle agil werden!

Allerdings bleiben Rahmenbedingungen für Projekte, z.B. regulatorische Einschränkungen, behördliche Anforderungen oder Qualitätsansprüche bestehen. Die Folge sind "hybride" Modelle. Die Firmen schaffen sich eigene „Mischungen“ aus agilen und traditionellen Ansätzen. Allerdings werden dabei die agilen Anteile immer wichtiger und dominanter¹⁶. Diese Verbreitung agiler und hybrider Modelle in den Zeiten der Digitalisierung lässt sich belegen. In der großangelegten Studie „HELENA-Survey“ 2017/2018, durchgeführt durch Mitglieder des GI-Fachausschusses Management der

¹⁴ Arbeitspakete im Kanban-Projekt („Sticky Notes“)

¹⁵ Quelle: Studentisches Projekt 2017, Kompetenzzentrum für agile IT-Prozesse (KAP) der DHBW Lörrach

¹⁶ Die GI-Fachgruppen Vorgehensmodelle und Projektmanagement haben die Bedeutung hybrider Ansätze früh erkannt und in ihren Tagungen behandelt: PVM2015 mit dem Thema „Hybride Projektstrukturen erfolgreich umsetzen“ oder PVM2016 mit dem Thema „Arbeiten in hybriden Projekten: Das Sowohl-als-auch von Stabilität und Dynamik“ (<http://www.pvm-tagung.de>).

Anwendungsentwicklung und -wartung¹⁷ und seiner Fachgruppen, wird dieser Trend bestätigt. Selbst vorsichtig betrachtet, werden agile Vorgehensmodelle bereits heute in mehr als der Hälfte aller Software-Projekte eingesetzt [Ku17].

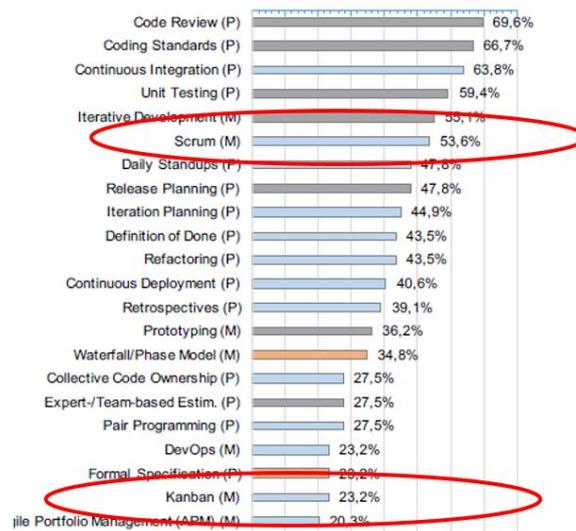


Abb. 5: HELENA-Survey 2017/18 (Auszug) [Ku17]

4 Der nächste Schritt in die digitale Zukunft

Wie kann noch schneller und flexibler auf sich ändernde Projektbedingungen reagiert werden? Sind Vorgehensmodelle – auch agile und hybride – überhaupt noch zeitgemäß? Braucht es nicht vielmehr die „optimal zusammengesetzten“ Teams mit ihrem eigenen „agilen Werkzeugkasten“, aus dem sie die Praktiken und Techniken holen, die im aktuellen Projekt geplant sind?

Einen Fingerzeig in die richtige Richtung geben die Systemischen Ansätze, die – aus der Organisationspsychologie abgeleitet – das Entwicklungsteam als System sehen, welches mit seiner „Umwelt“, seinen Stakeholdern und allen Projekt-Randbedingungen interagiert. Der 2010 vom Autor erstmals veröffentlichte MAP-Ansatz¹⁸ ist kein Vorgehensmodell mehr, sondern nur noch ein Meta-Modell, also einen Modellrahmen, in dessen Mittelpunkt die optimale „Komposition“ des Teams ist. Ausgehend von 6 „Typen“ von Team-Mitgliedern, die aus (erwünschten) Eigenschaften gebildet werden, werden die Projektrollen optimal besetzt. Das Team ist damit „empowered“, sich den

¹⁷ der auch die Tagung SWM2019 ausrichtet

¹⁸ Meta Agile Process Model [Ha10], Kompetenzzentrum für agile IT-Prozesse, DHBW Lörrach (<http://www.dhbw-loerrach.de/kap.html>)

aktuell optimalen Prozess aus dem „agilen und hybriden Werkzeugkasten“ schnell und effizient zusammensetzen.



Abb. 6: MAP – Landkarte der Typen im Team

Es kommt also insbesondere bei Digitalisierungs-Projekten in erster Linie auf die Team-Zusammensetzung an! Ein gutes Team schafft sich mittels seines „agilen Werkzeugkastens“ aufgrund der Projektanforderungen seinen eigenen hybriden Prozess! Mit MAP kann das erfolgreiche Team konstruiert werden. Tatsächlich wird durch MAP der Teamerfolg vorhersagbar: Eine Analyse von 14 konkurrierenden Teams zeigt, dass man das Zueinanderpassen („Matching“) der Team-Mitglieder durch einen „MAP-Koeffizienten“ des Teams noch vor dem Projektstart bestimmen kann. Beim Vergeben der Teamrollen ist das Ziel, diesen MAP-Koeffizienten zu maximieren. Tatsächlich korreliert im Vergleich der 14 Teams ein höherer MAP-Koeffizient mit besserem Teamerfolg. Teamerfolg wird also vorhersagbar!

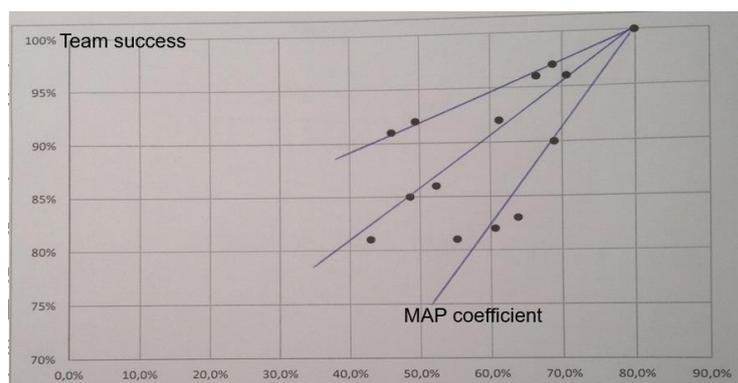


Abb. 7: MAP – Messung des Teamerfolgs

Das Beispiel des systemischen Ansatzes MAP zeigt, dass im Zeitalter der Digitalen Transformation nicht mehr das gewählte Vorgehensmodell entscheidend ist, sondern die Zusammensetzung des Teams und seine Fähigkeit, auf disruptive Anforderungen effizient und schnell zu reagieren. Im individuellen Teamprozess werden „reine“ agile Vorgehensmodelle wie Scrum und Kanban zu anpassbaren agilen Bausteinen auf einem problemorientierten Lösungsweg mit dem Ziel, selbst bei sich schnell ändernden Anforderungen das für die Kunden optimale IT-Produkt zu entwickeln.

Literaturverzeichnis

- [An11] Anderson, D.: Kanban – Evolutionäres Change Management für IT-Organisationen. Dpunkt, Heidelberg, 2011.
- [Sc04] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Washington, DC, 2004.
- [Fa85] Fairley, R.: Software-Engineering Concepts, Software Engineering Concepts. McGraw-Hill, New York, 1985.
- [Ha10] Hanser, E.: Agile Prozesse: Von XP über Scrum bis MAP. Springer, Heidelberg, 2010.
- [HH08] Höhn, R.; Höppner S.: Das V-Modell XT. Springer, Heidelberg, 2008.
- [HL11] Hilbert M.; López P.: The World's Technological Capacity to Store, Communicate, and Compute Information. sciencemag.org, 2011, Stand: 10.11.2018
- [HM08] Herzwurm G.; Mikusz M. (Hrsg.): Industrialisierung des Software-Managements. LNI Proceedings, 2008.
- [HMM13] Hanser E.; Mikusz M.; Fazal-Baqae M. (Hrsg.): Vorgehensmodelle – Anspruch und Wirklichkeit. LNI Proceedings, 2013.
- [Ku17] Kuhrmann M. et.al.: Hybrid Software and Systems Development in Practice: Waterfall, Scrum, and Beyond. In Proceedings of 2017 International Conference on Software and Systems Process, Paris, 2017 (ICSSP'17)