Hybride Vorgehensmodelle in der Versionserstellung – ein Praxisbeitrag

Dimitri Petrik[1]

Abstract: abilex GmbH ist ein KMU, welches neben anderen Dienstleistungen „Softwaretest as a Service“ anbietet. Einen großen Teil der Beauftragungen bilden Tests spezifischer produktdatenintensiver Point-of-Sale Software im Enterprise Bereich. Diesen Service fasst abilex als „Versionserstellung“ zusammen, wobei die Aufbereitung jeder neuen Version von abilex als ein Projekt betrachtet wird. Dabei werden, zum Teil bewusst und zum Teil unbewusst, sowohl klassische- als auch agile Vorgehensmodelle des Projektmanagements angewandt. Daher hat dieser Praxis-Beitrag das Ziel, die Herausforderungen und Besonderheiten der Anwendung von hybriden Projektmanagement-Methoden im Bereich des Managed Software Tests in unternehmensübergreifenden Konsortien zu liefern.

Keywords: Projektmanagement, hybride Vorgehensmodelle, hybrides Projektmanagement, agiles Projektmanagement, Projektmanagement in Teilprojekten, Projektmanagement im Konsortium, Projektmanagement im Wertschöpfungsnetzwerk, Software-Test, Testing as a Service, Managed Software Testing, Versionserstellung

1. Einleitung

1.1 Allgemeines

Viele Projektmanagement-Ansätze, sind in der IT-Branche entstanden und entweder als klassische oder als agile Ansätze bekannt. Um Gründe für Probleme in Projekten aufzuzeigen, wurde das sequentielle Wasserfallmodell entwickelt, das als klassisches Modell gilt [Sc10]. Später entstanden agile Methoden, um bekannte Probleme klassischer Modelle zu lösen. Ein bekannter Vertreter des agilen Projektmanagements ist „Scrum“, das sich am agilen Manifest für die schlanke Softwareentwicklung orientiert. [He10]. Doch die agilen Methoden haben ebenfalls ihre Grenzen, so bieten sie bspw. weniger Hilfe in Form von unterstützenden Tools und Prozessen oder einer expliziten Risikosteuerung während eines Projekts. Einen Überblick über mögliche Einschränkungen beider Vorgehensmodelle liefert die Übersicht von Sandhaus, Berg und Knott, welche bekannte agile und klassische Vorgehensmodelle vergleicht.

[1] abilex GmbH, MBKS-Versionserstellung, Löffelstr. 4, 70597 Stuttgart, Dimitri.Petrik@abilex.de

Kriterium	Wasserfall modell	V-Modell	XP	Scrum
Phasenmodell	√	√	X	X
Skalierbarkeit	X	X	√	√
Flexibilität	X	X	√	√
Vorgehen	Sequentiell	Sequentiell	Iterativ	Iterativ
Methoden	X	√	√	√
Werkzeuge	X	√	X	X
Unterstützende Prozesse	X	√	X	X
Praxisnähe	X	√	√	√
Risikosteuerung	X	√	X	X
Qualitätssicherung	√	√	√	√

Abb. 1: Stärken und Schwächen von klassischen und agilen Vorgehensmodellen im Projektmanagement [KK14]

Die Praxis zeigt, dass sequentielle Modelle wie das V-Modell nach wie vor in vielen Projekten eingesetzt werden. Unterschiedliche Anforderungsspezifitäten erfordern unterschiedliche Vorgehensmodelle für den erfolgreichen Projektabschluss [Hü14] – so eignen sich klassische Modelle besser bei unerfahrenen Teams und eher klar definierten Anforderungen, wohingegen ein eingespieltes Team selbst bei unklaren Anforderungen oder neuen Technologien im agilen Vorgehen bessere Ergebnisse liefert [Hü14]. Doch findet eine strikte Trennung in der Praxis statt? Oftmals werden vorteilhafte Merkmale beider Vorgehensweisen im gleichen Projekt angewendet. Die Vermischung ist unter dem Begriff „hybride Vorgehensmodelle“ bekannt. [SBK14]. Dies zeigt eine 2011 durchgeführte Untersuchung.

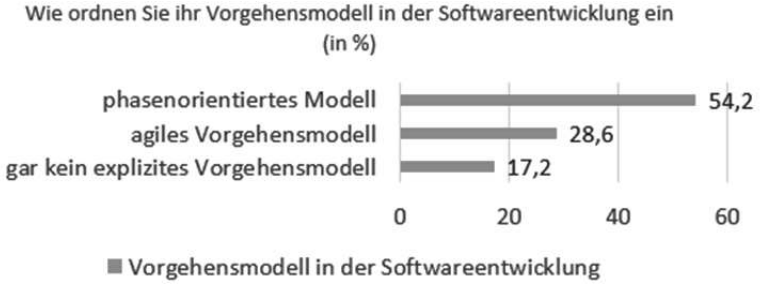

Abb. 2: Umfrage zur Verbreitung von Vorgehensmodellen bei der Softwareerstellung [SU11]

Dabei dominiert bei sequentiellen Modellen das allg. V-Modell mit 36,6% und Scrum bei den agilen Modellen mit 57%. An zweiter Stelle kommen bei beiden Vorgehensarten eigene Modelle vor, was auf hybride Vorgehensmodelle hindeutet. Mit diesem Beitrag

möchte der Autor, anhand der Vorgehensweise in Projekten seines Arbeitgebers im Bereich des Softwaretests, die Anwendung der hybriden Methoden untersuchen und die Herausforderungen oder gar Konflikte zwischen der Praxis und der Theorie zu hybriden Vorgehensmodellen im Projektmanagement aufzeigen.

1.2 Einordnung der Versionserstellung im PM und in der Praxis

Die abilex GmbH ist ein KMU, das aus dem Umfeld des Produktinformationsmanagements entstanden ist. Die Datenimplementierung und der begleitende Softwaretest von produktdatenintensiver Point-of-Sale-Software (PoS-Software) in der Automobilindustrie wurden als Kernkompetenz erarbeitet. In der Dienstleisterrolle untersucht abilex die Kundensoftware auf Fehlerwirkungen und Defekte, es findet jedoch kein Debuggen im Sinne der Beseitigung der Fehlerzustände statt. [SL06] Die gefundenen Fehlerwirkungen und Defekte werden analysiert und anschließend an den Kunden oder an den Softwarehersteller kommuniziert. Die Dienstleistung bietet somit im Rahmen des Outsourcings den Software-Test als Service an. Geschuldet den besonderen Softwaremerkmalen der A-Kunden, hat sich abilex im Rahmen des Softwaretests ebenfalls auf die Produktdatenprüfung und –aufbereitung spezialisiert. In diesem Zusammenhang ist der Begriff der Versionserstellung entstanden – als Sonderform des produktdatenintensiven Softwaretests. So wird korrekten Datenbeständen innerhalb der Versionserstellung eine Top-Priorität eingeräumt. Besonders wichtig sind marktspezifische Anforderungen an die Software, da sie international eingesetzt wird und marktspezifische Unterschiede aufweist, wie zum Beispiel landesspezifische Gesetzgebungen, Währungen, Steuern, Ausstattungen oder Fahrzeugkonfigurationen. Deshalb gehören zur Versionserstellung von PoS-Software die korrekte Produktdatenaufbereitung und die Lokalisierung der marktspezifischen Anforderungen.

Die Versionserstellung entspricht allen Merkmalen eines Projekts nach DIN-Norm und wird deshalb in Projektform geführt.[2] Fehlerfreie Markteinführung markiert die Zielvorgabe, während der Kunde eine Zeitvorgabe definiert. Ebenfalls werden der Preis und der Leistungsumfang im Vorfeld definiert und priorisiert und die begrenzte Anzahl der Projektmitglieder für die Versionserstellung vereinbart. Zudem wird die Versionserstellung klar von anderen Aktivitäten des Unternehmens abgegrenzt. Somit eignet sich die Versionserstellung als valides Beispiel für ein Projekt.

[2]Vorhaben, das im Wesentlichen durch die Einmaligkeit aber auch Konstanz der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie zum Beispiel Zielvorgabe, zeitliche, finanzielle, personelle und andere Begrenzungen; Abgrenzung gegenüber anderen Vorhaben; projektspezifische Organisation.

Insgesamt umfasst die Versionserstellung folgende sequentiellen Prozesse: Vorbereitung, Dateneingangsprüfung, Aufbereitung von Datenständen, Test und die Marktauslieferung. Die nächste Abbildung ordnet die Versionserstellung innerhalb der Softwareerstellung ein und verdeutlicht die einzelnen Aufgaben - sog. Arbeitspakete innerhalb des Versionserstellungsprojekts.

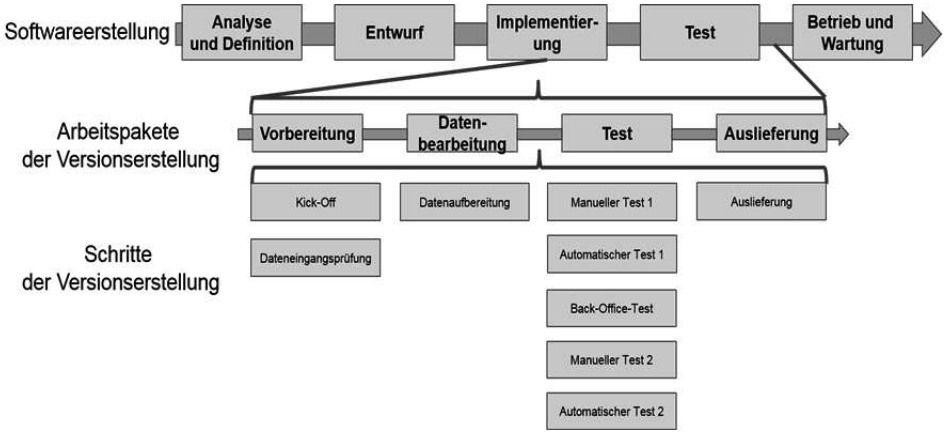

Abb. 3: Einordnung im Softwareerstellungsprozess und Darstellung der einzelnen Subtätigkeiten im Rahmen der Versionserstellung

Dabei wirft die sequentielle Vorgehensweise folgende Fragen auf:

- Können in der Vorbereitung alle Fragen geklärt werden?
- Was passiert wenn Bugs erfasst werden?
- Ist ein einmaliger Testdurchlauf ausreichend?

Die Bewältigung dieser offenen Punkte erfordert eine iterative Vorgehensweise im Projekt. Deshalb ist eine Untersuchung dazu sinnvoll, wie agil die sequentielle Vorgehensweise in der Versionserstellung ist. Darüber hinaus ordnet die Abbildung 3 die Versionserstellung als Teilprojekt innerhalb des ganzheitlichen Softwareprojekts der PoS-Software aus der Kundensicht ein, was zur Forschungslücke führt.

1.3 Forschungslücke

Die Wissenschaft konzentriert sich bei hybriden Vorgehensmodellen auf ganzheitliche Softwareerstellungsprojekte – die Möglichkeit einer horizontalen Kooperation zwischen unterschiedlichen Serviceanbietern wird nicht beachtet. Doch genau solche Konstellationen, bestehend aus rechtlich unabhängigen, agilen KMUs, kommen in der Realität oft vor. Die in der Literatur dargestellte Situation findet man eher bei simplen Software-Projekten (simple Smartphone Apps) oder bei großen Softwareanbietern, welche in der Lage sind, die Softwareerstellung ganzheitlich allein zu bewältigen. Aus vielerlei be-

kannten Gründen wird den Unternehmen dennoch empfohlen, sich auf die Kernkompetenzen zu konzentrieren. Unter dieser Prämisse agiert auch abilex als Anbieter von Managed Software Testing in einem Verbund aus unterschiedlichen Unternehmen und übernimmt für seine Kunden aus der Automobilindustrie den Softwaretest. Neben Kostenvorteilen hat diese in der Praxis durchaus verbreitete Lösung den Vorteil von höherer Qualitätssicherung der Software – unabhängige Testteams liefern in der Regel Software von höherer Qualität ab, weil trotz eines relativ hohen Einarbeitungsaufwands, externe Experten spezielles Know-How besitzen und eine emotionale Bindung gegenüber den Entwicklerteams nicht gegeben ist [SL05]. Im vorliegenden Praxisbeitrag wird von Großkonzernen als Kunden der Software-Test als eigenständiges Projekt an Dienstleister vergeben. Diese Aufgabenverteilung zwischen Unternehmen begünstigt die Anwendung hybrider Vorgehensmodelle. Da aus der Sicht des Autors die Anwendung hybrider Projektmanagementmethoden bei Unternehmenskooperationen in der Wissenschaft bisher wenig behandelt wurde, deutet dies auf eine vorhandene Forschungslücke hin.

1.4 Aufbau und Literaturanalyse

Im vorliegenden Praxis-Beitrag wird die Versionserstellung aus der Projektmanagementsicht beschrieben und systematisch auf das Vorhandensein der Merkmale klassischer und agiler Vorgehensmodelle untersucht, um daraus die Anwendung hybrider Modelle abzuleiten. Im Unterschied zu bereits erschienen Publikationen wird ein Bottom-Up Ansatz verfolgt, da der Schwerpunkt der Untersuchung am Beispiel der Versionserstellung als Dienstleistung und Teilprojekt betrachtet werden Manged Software Tests als Dienstleistung stllen einerseits als Kernkompetenz einen hohen Wert für abilex dar und leistet einen hohen Qualitätsbeitrag zum erfolgreichen Abschluss des Gesamtprojekts aus der Kundensicht – der erfolgreichen Markteinführung einer neuen PoS-Software. Die Projektuntersuchung umfasst ebenfalls die Rollen der Projektbeteiligten und die eingesetzten Tools. Zum Schluss wird das Vorgehen in der Versionserstellung schematisch dargestellt und in das ganzheitliche Projekt zur PoS-Software eingeordnet. Anschließend werden Empfehlungen für weitere Forschungsrichtungen zu hybriden Vorgehensmodellen im Projektmanagement abgeleitet.

1.5 Ergebnisse der Literaturanalyse

Der Mangel an Publikationen zur Anwendung hybrider Vorgehensmodelle in Unternehmensverbünden wird anhand der Literaturanalyse deutlich. Durchsucht wurden die Datenbanken Sciencedirect, IEEE, ACM und Ebscohost, wobei die Suche über Ebscohost auch die Suche bei Business Source Premier, Academic Search Premier und EconLit beinhaltet. Die Literaturrecherche umfasste nur Papers und Journals, erschienen ab 2004. Erste Literaturrecherche mit folgenden Suchbegriffen[3] ergab keine Treffer. Lediglich

[3] String1: „project management“ AND „hybrid approach“ AND „value network“

eine Vernachlässigung des Suchbegriffs „value network“ im Titel, Abstract und Keywords brachte Ergebnisse zu hybridem Projektmanagement. So beschreiben Lehman und Sharma eine hybride Projektmanagementform für „Softwareentwicklung as a Service“ bei IBM, welche agile Modelle für die Prototypenerstellung und die anfängliche Entwicklung anwendet und die dann nach Erreichen eines bestimmten Reifegrads zu klassischen nach Plan geführten Modellen übergeht [LS11]. Bedingt durch den Mangel an Veröffentlichungen wurde eine weitere systematische Literaturanalyse anhand der Titel, Abstracts und Keywords durchgeführt[4], wobei die Suche sowohl hybrides als auch agiles Vorgehen berücksichtigt hat. Überraschend war die unterschiedliche Einordnung des hybriden Projektmanagements. So sehen einige Paper hybride Vorgehensmodelle als Vorstufe zu agilem Projektmanagement, während andere hybrides Projektmanagement für geeigneter und daher notwendiger als rein agiles Vorgehen halten. So zeigen Serrador und Pinto in einer empirischen Untersuchung dass die Erfolgsaussichten eines Projekts mit dem Anteil agiler Elemente zusammenhängen und positiv korrelieren. In dem Artikel weist die Mehrheit der untersuchten Projekte (aus unterschiedlichen Branchen und zum Teil unternehmensübergreifend) von 65% agile Merkmale gemischt mit klassischem Vorgehen auf, womit sie als hybrid gelten. Dabei wird hybrides Vorgehen als temporärer Zustand auf dem Weg zu vollständig agilen Projekten angesehen [SP15]. Diese Einordnung wird ebenfalls von Stettina und Hörz vertreten, die in ihrer Untersuchung zum agilen Portfoliomanagement von zeitlichen Lernprozessen bei Unternehmen sprechen, die nötig sind um agil vorzugehen. Für agiles Vorgehen müssen Routinen konsolidiert werden und während dieses Lernprozesses wird nicht komplett agil vorgegangen. Somit werden hybride Vorgehensmodelle erneut als vorübergehender Zustand auf dem Weg zu vollständig iterativen Vorgehensmodellen betrachtet. [SH14] Weiteres Paper zu großen outsourcten Projekten liefert zwei wichtige Merkmale zur Übertragung auf die Versionserstellung. So behaupten die Autoren, dass agiles Projektmanagement neue Werte im Unternehmen integriert, wofür Zeit benötigt wird. Außerdem kommen die Autoren zur Ansicht, dass agiles Vorgehen weitere Prozesse im Unternehmen auslöst und sogar die Einbindung der Kunden in das Projekt verstärkt und die Kundenkommunikation agil werden lässt. [Da12]. Weitere Untersuchung zu Projekten mit Beteiligung mehrerer Agente von Domann et al. zeigt, dass klassische Methoden den Projekterfolg nicht sichern können und besonders zur Steuerung von Projekten mit einer Mehrzahl an Akteuren braucht es agile Werkzeuge und Tools. Laut dem Paper können klassische Vorgehensmodelle kleine agile Projekte mit kleinen Projektteams, dynamischen Anforderungen und vielen Stakeholdern schlecht steuern. Andererseits, passen Werkzeuge agiler Vorgehensmodelle oftmals nicht zu Projekten mit mehreren Akteuren (unabhängige Unternehmen oder Beziehungen zwischen Kunden und Dienstleistern) [Dj14]. Zur Effizienz von agilen Projekten sagen Serrador und Pinto, dass agile oder hybride Projekte ähnlichen Planungsaufwand im Vorfeld benötigen wie die klassischen Projekte verursachen. Sollte grundlegende Planung erst während des Projektablaufs stattfinden, so erhöht sich der Planungsaufwand bei agilem Vorgehen gegenüber dem klassischen

[4] String2: "Project" AND ("Agile" OR "Hybrid*" "network" "value*")

[SP15]. Martakis und Daneva haben mittels einer empirischen Untersuchung festgestellt, dass hybride Vorgehensmodelle am besten skalierbar sind und die Anforderungsabhängigkeiten als Risiko am besten lösen. Zudem wenden große Unternehmen Risikosteuerungsmethoden an, während kleine Unternehmen sich mehr auf Kommunikation zwischen Projektmitgliedern verlassen [MD13]. Spundak hingegen sieht klassische Vorgehensmodelle als besser geeignet zur Durchführung großer Projekte an, unabhängig davon ob die Projekte groß hinsichtlich ihrer Dauer, Teamgröße, Anzahl an Anforderungen oder Komplexität sind. Weiteres Paper von van Waanderburg und van Vliet beschreibt, dass auch große Unternehmen in Projekten versuchen agil vorzugehen. Die Autoren sehen hybrides Vorgehen bei großen Unternehmen als eine Koexistenzphase, während der Transformation von klassisch zu agil [WV13]. Diese Einordnung deckt sich mit der Sicht von Serrador und Pinto. Außerdem behaupten van Waanderburg und van Vliet, dass agiles Projektmanagement sich ebenfalls für große Unternehmen eignet, obwohl in großen Unternehmen Strukturen und Positionen existieren, die Mehraufwand während des Projektverlaufs generieren. Trotzdem können sie nicht so schnell umgestaltet werden. Aus diesem Grund seien hybride Vorgehensmodelle in IT-Projekten der befragten Großkonzerne verbreitet [WV13]. Aus dieser Einordnung heraus stellen die Autoren mangelndes „Business Involvement“ als Hindernis auf dem Weg zu agilem Projektmanagement dar, da es vom Management oder Kunden ausgeht. Weiterhin beschreiben die Autoren die Bottom-Up Ausbreitung des agilen Projektmanagements bei beiden befragten Großkonzernen – es erfolgte in einzelnen Teams, um später weitere Bereiche der Organisation zu beeinflussen. Weitere Merkmale klassischer, agiler und hybrider Vorgehensmodelle wurden anhand deutscher Quellen bei Springerlink und aus dem ProjektMagazin herausgearbeitet und finden Anwendung im Kapitel 3.

2. Hybride Vorgehensmodelle in der Versionserstellung

2.1 Projektablauf in der Versionserstellung

In diesem Kapitel wird der Projektablauf am Beispiel der Versionserstellung einer PoS-Software beschrieben. Wie in der Abbildung 4 zu sehen, wird bei abilex (auch Dienstleistungsunternehmen) das Projekt in vier Arbeitspakete unterteilt. Die Unterteilung sieht zwar nach einer statischen Planung mit sequentiellen Paketen aus, aber es besteht ebenfalls aus diversen Aspekten des agilen Projektmanagements.

Für den erfolgreichen Projektabschluss werden Kompetenzelemente gemäß der National Competence Baseline der GPM angewendet. Schon vor dem Projektbeginn werden durch die Versionserstellung als Dienstleister die Stakeholder analysiert - es sind überwiegend die Auftraggeber als Kontroll- und Koordinationsinstanz und die Verkäufer als zukünftige Nutzer der Software. Die Weiterentwicklung der PoS-Software im Konsortium wird vom Softwarelieferanten durchgeführt. Bereits im Vorfeld wird ein Release-Plan vom Kunden in Absprache mit dem Dienstleistungsanbieter vorbereitet und ein Rahmenplan für die Fertigstellung einer bestimmten Version erstellt. Aus der Kunden-

sicht ist es deshalb ein nach der Anforderungsspezifikation aufgestellter Plan, der aus der Sicht der Versionserstellung bereits vor Projektbeginn feststeht.

Für die Versionserstellung fängt das Projekt mit einem landesspezifischen Kick-Off Termin an, an dem landespezifische Vertriebspartner des Kunden teilnehmen. Beim Kick-Off werden mögliche Projektverzögerungen und Risiken besprochen – sei es aufgrund einer verspäteten Datenlieferung, welche vom ursprünglichen Release-Plan abweicht, dem Urlaub wichtiger Projektmitglieder oder nationaler Feiertage der jeweiligen Länder. Demnach werden im Kick-Off Risiken frühzeitig erkannt, [GPM08] um die qualitativen und zeitlichen Ziele des Versionserstellungsprojekts nicht zu gefährden. Aus der Kundensicht (Top-Down) ist der Kick-Off mit der Versionserstellung aufgrund der vorgelagerten Weiterentwicklung der Software und der Datenaktualisierung zwar mitten im Projektverlauf anzusiedeln, für abilex (Bottom-Up) markiert der Kick-Off jedoch den Projektanfang.

Nach dem Kick-Off erhält die Versionserstellung die benötigten Datenupdates oder auch eine neue Programmversion. Bei neuen Programmversionen oder auch nach Kundenwunsch fängt anschließend der Basistest an, um die Basisfunktionalitäten der Software zu prüfen. Wenn die Versionsersteller Fehlerwirkungen in der Basisfunktionalität nachweisen, kann es als Kriterium für den Abbruch der Versionserstellung gelten. Gefundene Fehlerwirkungen werden über ein Online-Ticket System an den Hersteller kommuniziert. Die Kommunikation kann auch der Kunde mitverfolgen und er kann so zeitnah über mögliche Verzögerungen informiert werden. Der Basistest deckt die Fehlerwirkungen sowohl in einzelnen Komponenten als auch im gesamten System auf. Abhängig von der Auslieferung des Entwicklungspartners, werden einzelne Komponenten der Software - ergänzt um Platzhalter - oder das komplette System inkl. Schnittstellen getestet.

Bei erfolgreich bestandenem Basistest oder bei einem Minor-Update (Datenaktualisierung aufgrund neuer Produkte, Preise, Steuern, Wechselkurse) fängt die umfassende Aufbereitung der Testversion an, welche zu Beginn automatische Tests beinhaltet und mit der Auslieferung der Testversion endet. Bei der Versionserstellung der Testversion wird ebenfalls die Datenlieferung vom Vertriebsdatenmanagement des Kunden aufbereitet und auf fehlerhafte Werte und Produktdaten getestet. Dabei wird der Datenstand zuerst manuell, anschließend automatisiert getestet. Sollten große Probleme im Datenstand aufgedeckt werden, werden diese über ein Ticketsystem dem Kunden umgehend mitgeteilt. So haben die Verantwortlichen für die Produktdaten auf der Kundenseite bereits während des ersten Tests die Möglichkeit, die Fehlerzustände der ersten Datenlieferung zu beheben.

Anschließend wird der Datenstand gemäß dem „Vier-Augen-Prinzip“ von einem unabhängigen Tester nochmals getestet. Das Testen durch zwei unterschiedliche Personen sichert die Unabhängigkeit der Testfallbewertung und verringert die Wahrscheinlichkeit eines falsch verstandenen Testfalls, womit es zum Erfüllen der Qualitätsziele beiträgt. Nach dem ersten Testdurchlauf wird die Testversion dem Vertriebsinnendienst des Kunden zum fachlichen Test übergeben. Hierbei wird gemäß der Anforderungsspezifikation

eine anwendernahe Sicht durch neue Testfälle simuliert und nach Abschluss mitsamt Feedback wieder an das Dienstleistungsunternehmen übergeben.

Nach dem fachlichen Kundentest wird, um zeitnahe Datenkorrekturen ergänzt, ein finaler Test gestartet. Dieser dauert ebenfalls vier Wochen und beinhaltet die gleichen Testfälle wie der erste Testlauf. Am Ende des zweiten Testlaufs übernimmt die Versionserstellung das Release der Software und stellt die Software für die Zielmärkte des Kunden bereit Abschließend werden die im Projekt entstandenen Probleme und deren Lösungen in einem Review-Meeting zusammengefasst. Das Review-Meeting wird inkl. Protokollführung durchgeführt, um die Testfälle anzupassen und die Kernkompetenz im Managed Testing zu schärfen. So werden im Review-Meeting wenig effektive Testfälle angepasst oder aussortiert und der Vorgang im Sinne von „Lessons Learned“ festgehalten. Im Rahmen der Versionserstellung entstandene Probleme werden in einem umfassenderen Abschluss-Meeting diskutiert, wobei der Schwerpunkt nicht beim Testmanagement liegt, sondern alle Vorkommnisse des letzten Projekts besprochen werden. Zusätzlich werden über tägliche Meetings während der Projektlaufzeit - moderiert vom Teamleiter - der aktuelle Stand einzelner Versionsersteller ausgetauscht und mögliche Probleme diskutiert. Zusätzlich werden vom Kunden fünf bis sechs Scope-Meetings in Anwesenheit aller verantwortlichen Fachbereiche veranstaltet, um die Erreichung der qualitativen oder zeitlichen Ziele zu besprechen. An den Scope-Meetings nimmt der Teamleiter der Versionserstellung regelmäßig teil.

2.2 Wichtige Rollen während des Projektablaufs

Neben dem Versionsersteller sind im Projekt zwei weitere Personen von besonderer Bedeutung – der Testmanager und der Teamleiter bzw. der Projektleiter. Der Testmanager nimmt eine unterstützende Rolle ein und agiert während der Versionserstellung als Vermittler bei Problemen zum Testfallverständnis oder zu Modalitäten rund um eine Software-Version oder zu den Datenständen. Nach dem Abschluss des ersten Testdurchlaufs werden Berichte aus der Testmanagement-Software von Versionserstellern exportiert und dem Testmanager zur Auswertung weitergereicht. Am Ende des zweiten Testlaufs leitet der Testmanager das Review Meeting, um die Testfälle zu überarbeiten und die Änderungswünsche der Versionsersteller aus dem „Lessons-Learned“ zu priorisieren und bis zum nächsten Projekt zu implementieren. Damit trägt der Testmanager zur Qualitätssicherung bei, um wenig effektive Testfälle, welche keine Fehlerzustände aufdecken oder den Versionsersteller verunsichern (falsches Verständnis), zu eliminieren. Insgesamt agiert der Testmanager als fachlicher Ansprechpartner und Facilitator in Bezug auf Testprozesse. Er hat keine organisatorisch schützende Funktion über andere Tester während des Projekts und leitet auch keine Daily Meetings. Diese Aufgaben werden vom Teamleiter erledigt.

Der Teamleiter hat hingegen die Befugnis, Mitarbeiter vor anderen Aufgaben abzuschirmen, da die Mitarbeiter auch in anderen Projekten eingesetzt werden können. Zusätzlich beruft der Teamleiter die täglichen Meetings ein und leitet diese. Außerdem

werden vom Teamleiter zur Maximierung des Produktwerts die Anforderungen an die Versionserstellung durch Teilnahme an Kundenterminen priorisiert. Der Teamleiter pflegt ebenfalls die Kundenanforderungen und verantwortet die Aufnahme der Produktmerkmale. Da die Versionserstellung nach dem V-Modell am Ende des Gesamtprojekts anzusiedeln ist, hat der Teamleiter keinen Einfluss auf die Priorisierung der Kernfunktionalitäten der Software. Diese werden zwischen dem Kunden und dem Entwicklungsunternehmen definiert, womit die Priorisierung durch den Teamleiter in der Versionserstellung sich nur auf den Software-Test und die Datenimplementierung bezieht.

2.3 Tools zur Unterstützung des iterativen Vorgehens

Während der Projektlaufzeit zum Test der PoS-Software werden diverse Tools eingesetzt. So werden fachliche Verbesserungen in gemeinsam gepflegten Excel-Dokumenten aufgenommen und anschließend werden die Ergebnisse per E-Mail, kollaboratives Intranet oder in Teammeetings kommuniziert. Die Kick-Offs, die ständig verbesserten Testfälle, die Kommunikation mit dem Kunden und die Meetings während der Versionserstellung haben relativ feste Zeiten und führen normalerweise zu keinen Abweichungen. Im Falle einer längeren Krankheit kann ein anderer Versionsersteller die Vertretung übernehmen. Dazu trägt ebenfalls ein Wiki bei, welches Wissen rund um Testprozesse beinhaltet und Wissensmonopole verhindert. Über Opensource Ticketsysteme und Gruppen-E-Mail-Konten wird ein enger Kontakt zum Kunden und zum Softwarelieferanten gepflegt, um Projektverzögerungen minimal zu halten. Da die Prozesse im Projekt eine bestimmte sequentielle Abfolge haben, werden GANTT-Diagramme zur Verfolgung des Fortschritts verwendet. Zur Durchführung von Kick-Off Veranstaltungen werden Telefonkonferenzen mit Bildschirmübertragung durchgeführt, wodurch kollaborative Arbeit ermöglicht wird. Für die iterative Abstimmung zwischen Versionserstellern und dem Projektleiter wird im Intranet eine Anwendung zur Erfassung des Fortschritts und der Zeitangaben in Tabellenform gepflegt. Die Notwendigkeit eines für Scrum empfohlenen Burndown-Charts [Ha10] besteht hingegen nicht, da die Projektverzögerungen fast ausschließlich aufgrund der sequentiellen Arbeitspakete von vorgelagerten Stellen (Softwarelieferant oder Kunde) des Konsortiumprojekts verursacht werden und ein Burndown-Chart keine Lösung schafft, sondern nur bekanntes aufzeigt. Da das Projektvorgehen bei den Partnern aus der Sicht des Dienstleistungsunternehmens eine Black-Box ist, macht eine unternehmensübergreifende Kooperation den Einsatz dieses agilen Tools überflüssig. Klassische Scrum-Dashboards werden auch nicht eingesetzt, aber eine historisch entstandene Intranet-Anwendung übernimmt die Funktionen des Scrum-Dashboards.

3. Anwendung auf die Versionserstellung

Der zuvor beschriebene Projektablauf wird nun auf die Merkmale bekannter Vorgehensmodelle überprüft.

Klassisch Vorgehensmodelle	Agile Vorgehensmodelle
Wasserfall-Modell	Scrum
V-Modell	Extreme Programming
Spiral-Modell	
Hybride Vorgehensmodelle	

Abb. 4: Übersicht der klassischen und agilen Vorgehensmodelle

Der vom Kunden vorgegebene Release-Plan stellt für die Versionserstellung gleichfalls die Anzahl von Sprints fest und ist deshalb mit Scrum-Flows vergleichbar [Ha10]. Der Kick-Off mit Kundenbeteiligung ist typisch für den Ausgangspunkt für Projekte bei dem Dienstleistungsunternehmen und markiert zugleich den Anfang in agilen Vorgehensformen wie Scrum. Von Anfang an werden im Rahmen des Kick-Offs die Anforderungen festgehalten, um alle Beteiligten des Kick-Offs bzgl. der Projektanforderungen auf einen einheitlichen Stand zu bringen. Die festgehaltenen Anforderungen ähneln einem Product Backlog [Sc10]. Die Beteiligung des marktspezifischen Vertriebsinnendienstes, sowie der Produktdatendokumentation auf der Kundenseite und die Teilnahme des marktspezifischen Versionserstellers sowie des Teamleiters als Repräsentanten auf der Seite des Dienstleisters, markiert den Kick-Off als eine kollaborative Schnittstelle am Projektanfang. Deshalb ist der Kick-Off ein agiles Merkmal zur Planung mit Kundenbeteiligung von Scrum. Einerseits verhindert es mangelndes „Business Involvmenet“ [WV13], andererseits hilft es den Planungsaufwand während des iterativen Vorgehens im Laufe des Projekts zu reduzieren [SP15].

Nach dem Kick-Off und erfolgreich durchgeführten Basistest fängt aus der agilen Projektmanagementsicht der erste Sprint an. Für die Versionserstellung bedeutet der erste Sprint einen vollständigen erstmaligen Test einer Version, welcher die Vorbereitung der Version, die Aufbereitung gelieferter Daten und den manuellen Test der Version und den automatisierten Test beinhaltet. Diese Prozesse laufen sequentiell ab und die Dauer des ersten Versionserstellungslaufs beträgt vier Wochen, womit es genau der Sprint-Laufzeit im Scrum entspricht [Me14]. Zudem stellt die Zeitangabe ebenfalls eine Time-Box dar. Während der Sprintdauer werden im Format des Daily-Scrum Meetings durchgeführt (der aktuelle Stand einzelner Versionsersteller untereinander ausgetauscht und mögliche Probleme oder Fehler diskutiert [Sc10] [KK14]. Allerdings werden die Scrum-Meetings aufgrund der kleinen Projektgröße nicht täglich durchgeführt. Außerdem können aufgrund von kurzen Kommunikationswegen innerhalb der Versionserstellung kleine Probleme informell unter Projektmitgliedern ausgetauscht werden. Werden große Datenfehler aufgedeckt, werden sie über ein Ticketsystem dem Kunden direkt mitgeteilt. Damit haben der Kunde oder der Softwarelieferant während des ersten Sprints die Möglichkeit, die Fehlerzustände zu beheben – ganz im Sinne der Flexibilität von Scrum [Sc10] und

des Feedback-Wertes nach Extreme Programming [He10]. Da die Kommunikation mit dem Kunden mittels eines modifizierbaren Opensource-Tools erfolgt, ist die technische Infrastruktur für Bugtracking und unternehmensübergreife Kollaboration vorhanden. Die Möglichkeiten des Tools zur unternehmensübergreifenden Kollaboration werden jedoch durch die Compliance-Richtlinien des Kunden unterbunden, da Projektmitarbeiter des Dienstleisters die Tickets nicht an die Projektteilnehmer auf der Kundenseite zuweisen oder namentlich adressieren dürfen. Damit wird die These von van Waardenburg und van Vliet anhand der Versionserstellung als Beispiel gestützt, wonach agile Methoden im Einsatz in großen Unternehmen (Kunde) durch organisatorische Strukturen gebremst werden und dadurch zwangsläufig hybride Vorgehensmodelle entstehen, da es schwierig ist im Projekt agile Tools anzuwenden. [WV13]. Nach Sandhaus, Berg und Knot hilft zwar ein Vertrauensaufbau beim Überwinden von bürokratischen Hürden [SBK14], dennoch wurde dieser Grad des Vertrauens in der Versionserstellung entweder noch nicht erreicht, oder der erreichte Grad genügt als Bottom-Up Wandel nicht, um konzernweite Kommunikationsregelungen mit Externen beim Kunden zu ändern.

Bisher aufgeführte Risikosteuerungsmittel basieren ausschließlich auf Kommunikation und entsprechen der Aussage von Martakis und Daneva, wonach in kleinen Unternehmen die Kommunikation zur Risikosteuerung vorherrscht [MD13].

Die Einteilung in Arbeitspakete innerhalb des Sprints mit der Time-Box von vier Wochen ist ein agiles Scrummerkmal [KK14]. Das Vier-Augen-Prinzip ist ebenfalls ein Grundprinzip des agilen Extreme Programming Modells [Me14], welches beim Dienstleistungsunternehmen für den Softwaretest adaptiert wurde. Es verhindert den alleinigen Wissensaufbau eines einzelnen Versionserstellers, der zum Beispiel im Falle einer längeren Krankheit oder des Ausscheidens aus dem Unternehmen verloren geht und entspricht dem agilen Prinzip des Pair Programming. [SBK14] Zusätzlich beugt das Wiki im Intranet der Bildung von Wissensmonopolen vor.

Der Test durch den Vertriebsinnendienst des Kunden spiegelt das Real Customer Involvement (RCI) aus dem Extreme Programming wieder [SBK14]. Da der Test am Ende des ersten Sprints erfolgt und eine aktive Kundenintegration ermöglicht, ist es unübersehbar ein iteratives Merkmal einer agilen Vorgehensweise im Projektmanagement. Insgesamt besteht in der Versionserstellung, entsprechend iterativen und agilen Vorgehensmodellen, eine zeitnahe Stakeholder-Kommunikation mit Kunden und Softwarelieferanten.

Was die oben beschriebenen Positionen des Testmanagers und des Team-/Projektleiters angeht, so übernehmen sie nur zum Teil die Scrumrollen aus der Theorie. Die Rolle des Projektleiters ist teilweise mit der Rolle des agilen Scrummasters vergleichbar, teilweise trägt sie aber klassische und hierarchische Züge eines klassischen Projektmanagers. Die Rolle des Scrummasters [He10] lässt sich aufgrund der Organisationsstruktur im Unternehmen nicht vollständig anwenden und die Aufgaben werden zwischen dem Teamleiter und dem Testmanager aufgeteilt. Ein Grund hierfür ist die personelle Situation, die den Scrummaster nicht von operativen Aufgaben befreien kann – so hat der Testmanager bspw. selbst eine Version aufzubereiten, was wiederum zu Interessenskonflikten führen

kann. Es gibt durchaus Tailoring von Scrumrollen im Praxiseinsatz [Sc10]. In der Versionserstellung ist Tailoring aus Kostengründen notwendig (kleines Team) und der Vorgehensgeschwindigkeit, sowie der Compliance des Kunden geschuldet – da beispielsweise direkter Kontakt mit dem Kunden nur von Repräsentanten erfolgen kann und dafür Teamleiter-Befugnisse notwendig sind.

Das erwähnte Review-Meeting zur permanenten Qualitätsverbesserung des Softwaretests, sowie das Abschluss-Meeting zu allgemeinen Vorkommnissen des Projekts, haben Ähnlichkeiten mit der agilen Sprint-Retroperspektive, da sie in einem begrenzten Zeitrahmen Verbesserungsvorschläge zu den letzten zwei Sprints zum Gegenstand haben [Ha10]. Da am Ende der Versionserstellung im Rahmen des Abschluss-Meetings die Versionen der PoS-Software nicht vordergründig besprochen und getestet werden, entspricht es deshalb formell nicht dem Sprint-Review nach Scrum [Ha10].

4. Fazit und Ausblick

Insgesamt verläuft die Versionserstellung als ein sequentielles Phasenmodell (Anlieferung, Aufbereitung, Test 1, Test 2, Auslieferung), das durch vielfältige agile Zusatzmerkmale angereichert wird, ganz i. S. v. Hüsselmann [Hü14]. Nach Sandhaus u. a. werden die Vorteile beider Vorgehen kombiniert und einzelne Iterationen entsprechen der Sprintdauer nach Scrum, angereichert um Daily Scrums zum iterativen Austausch, sowie Kick-Off zum Austausch mit dem Kunden [SBK14]. Für die Anwendung hybrider Vorgehensmodelle nach Lehman und Sharma, abhängig vom Reifegrad des Projekts, reicht die Dauer des Projekts nicht aus und ist daher im behandelten Praxis-Beispiel kaum anwendbar. Wenn das Gesamtprojekt aus der Gesamtprojektsicht des Kunden und nicht aus der Sicht vom Dienstleister betrachtet wird, so vollzieht sich das Softwarerelease nach klassischem V-Modell. Der Test fängt sequentiell nach dem Abschluss der Programmierarbeiten an und die Kommunikation mit dem Softwarelieferanten über ein Ticketsystem ist übertragbar auf das V-Modell. [SL05].

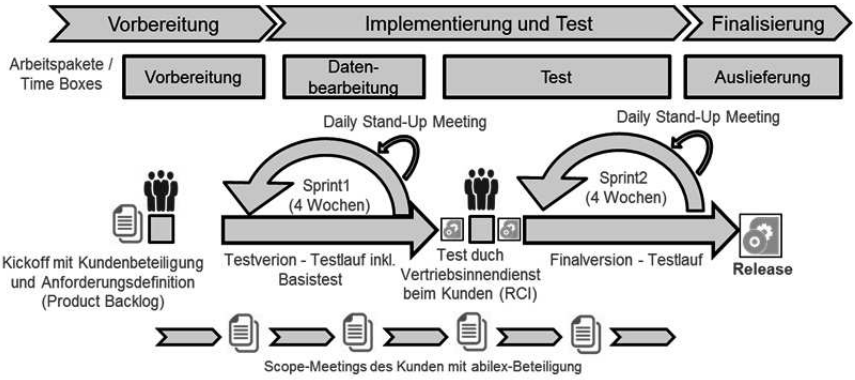

Abb. 5: Hybrides Vorgehen im Versionserstellungsprojekt bei abilex (Bottom-Up)

Der zweifache Testdurchlauf weist ebenfalls einige Merkmale des inkrementellen Spiral-Modells auf. Dennoch ist das Spiral-Modell mehr für komplexe und langwierige Entwicklungsprojekte gedacht. Die Versionserstellung ließe sich zwar in einer zweifachen Iteration darstellen, doch unterscheidet sich der Sinn der iterativen Durchläufe in dem Spiral-Modell von der Testwiederholung in der Versionserstellung. Während das Spiral-Modell eine beliebige Anzahl von Iterationen für ungewisse Risiken im Projektverkauf einhergehend mit einer Kostensteigerung zulässt, [Sc10] ist in der Versionserstellung ein zweifacher Durchlauf im Release-Plan vom Kunden vorgegeben, stellt für den Serviceanbieter für Managed Software Tests ein wichtiges Service-Merkmal dar und trägt zur Einhaltung von Projektzielen bei.

Bei einer ganzheitlichen Top-Down Betrachtung des Gesamtprojekts aus der Kundensicht, überwiegen eher klassische Merkmale, wobei abilex im sequentiellen Vorgehen des Kunden, nach dem klassischen V-Modell eingebunden ist und das Subprojekt der Versionserstellung agil steuert. Somit ist die Vorgehensweise im Gesamtprojekt auf jeden Fall hybrid – klassische Vorgehensweise im Gesamtprojekt und überwiegend agile Vorgehensweise in Teilprojekten.

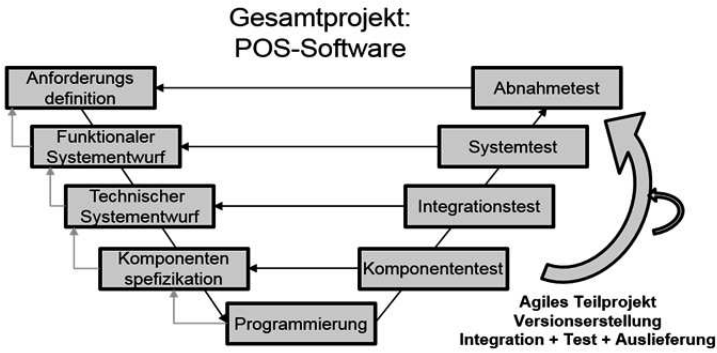

Abb. 6: Hybrides Vorgehen im Gesamtprojekt der PoS-Software (Top-Down)

Dies entspricht der Definition von hybriden Vorgehensmodellen [KK14] und dient als Beispiel für die gefundenen Thesen als Beispiel für den Einsatz von hybriden Methoden bei Großkonzernen. Diese These vertreten in ihren Artikeln Martakis und Daneva, van Waanderburg und van Vliet. Das Beispiel der Versionserstellung passt zu der These von Domann u. a., wonach Projekte mit mehreren Agenten durch klassisches Vorgehen nicht zu managen sind und den Einsatz agiler Tools erfordern. Dennoch können gewachsene Strukturen beim größten Konsortiumteilnehmer agiles Vorgehen wie unternehmensübergreifende Kollaborationstools oder die reine Rolle des Scrummasters von den Anforderungen oder den gewachsenen Strukturen des Kunden ausbremsen. Im Fall der Versionserstellung wird der Grad der Agilität somit vom Kunden (Management des Gesamtprojekts) vorgegeben und vom Dienstleister als KMU (Management des Teilprojekts) adaptiert.

Die systematische Literaturanalyse und die Analyse des Projektablaufs in der Versionserstellung regen weitere Forschungstätigkeit zu folgenden Themen an. Die Literaturanalyse deckt unterschiedliche Einordnungen von hybriden Vorgehensmodellen auf. Deshalb ist es notwendig weitere Untersuchungen durchzuführen, um hybrides Projektmanagement systematisch einzuordnen, da hybrides Vorgehen von Unternehmen sowohl als Soll-Ziel, aber auch als vorübergehende Phase auf dem Weg zum agilen Vorgehen eingestuft wurde. Vor dem Hintergrund von unternehmensübergreifenden Projekten bietet es sich an, den Ausgangspunkt der Dynamik für hybrides oder agiles Vorgehen im Projektmanagement empirisch zu untersuchen, um einen Zusammenhang zwischen der Rolle eines Unternehmens in Kooperationen, der Unternehmenskultur und dem Vorgehen im Gesamtprojekt zu messen. Die Abhängigkeit zwischen der Partnerrolle und dem eingesetzten Vorgehensmodell bei Kunden-Partner-Beziehungen ist eine mögliche Forschungsrichtung. So wird im Praxisbeispiel der Versionserstellung das Vorgehensmodell bspw. nicht explizit vom Kunden vorgegeben, was aber in anderen Multistakeholder-Projekten anders sein kann. Dennoch hat abilex als Dienstleistungsunternehmen einige Elemente des agilen Vorgehens vom Kundenunternehmen übernommen. In diesem Zusammenhang müssen Einführungsstrategien für hybrides Projektmanagement erarbeitet werden, damit der Wandel von gewachsenen und traditionellen Organisationsstrukturen zwischen Kooperationspartnern gleichmäßig verläuft.

Die Ergebnisse der Literaturanalyse zur Notwendigkeit agiler Frameworks und deren Limitierung durch administrative Hürden, bietet die Grundlage für Untersuchung wie stark hybride Vorgehensmodelle von Tools unternehmensübergreifend unterstützt werden können und wie stark deren Einsatz von administrativen Hürden begrenzt wird. Manche Merkmale des agilen Vorgehens werden in der Versionserstellung nicht formal eingehalten, wie zum Beispiel die Meetings im Daily-Scrum-Format, die aufgrund der kleinen Projektgröße nicht täglich stattfinden. Daher bilden die Zusammenhänge zwischen Projektgröße und Skalierbarkeit agiler Tools weitere interessante Forschungsrichtungen. Anhand solcher Messungen können allgemeingültige Best Practices und Empfehlungen abgeleitet werden.

Resümierend sei angeführt, dass in unternehmensübergreifenden Projekten es zu Konflikten zwischen der Anwendung ausschließlich agiler Methoden und den gewachsenen Organisationsstrukturen, traditioneller Compliance oder den bestehenden vertraglichen Beziehungen kommen kann. Für solche Konstellationen sind hybride Vorgehensmodelle deshalb vorteilhafter als agile Methoden, da die traditionellen Komponenten zur Risikosteuerung und für einen reibungslosenreibungslosem Projektablauf bei Unternehmen unterschiedlicher Kulturen, Größen oder Branchen geeigneter sind.

Literaturverzeichnis

[Dj14] Domann, J. et.al.: An Agile Method for Multiagent Software Engineering, In: Procedia Computer Science 32, S. 928–934, 2014.

[Da12] Daneva, M. et.al.: Agile requirements priorization in large-scale outsourced system projects: An empirical study, In: The Journal of Systems and Software 86, S. 1333–1353, 2013.

[GP08] GPM, Deutsche Gesellschaft für Projektmanagement e. V., http://www.gpm-ip-ma.de/fileadmin/user_upload/Qualifizierung___Zertifizierung/Zertifikate_fuer_PM/National_Competence_Baseline_R09_NCB3_V05.pdf, Stand: 25.05.2015.

[Ha10] Hanse E.: Agile Prozesse: Von XP über Scrum bis MAP, Springer Verlag, Berlin, Heidelberg, 2010.

[Hü14] Hüsselmann C., http://www.scheer-management.com/wp-content/uploads/2014/07/Hybride-Vorgehensmodelle_Dr.-Claus-H%C3%BCsselmann.pdf, Stand: 14.05.2015.

[KK14] Kirchhof M., Kraft B.: Hybrides Projektmanagement. Agile und klassische Methoden im Projekt koordinieren. In: Spotlight – Agiles Management - ein Überblick, Berleb Media GmbH, Taufkirchen, S. 38–48, 2014.

[LS11] Lehman T. J., Sharma A.: Software Development as a Service: Agile Experiences, In: SRII Global Conference (SRII), 2011 Annual, S. 749–758, 2011.

[Ma13] Maximini D.: Scrum - Einführung in der Unternehmenspraxis, Von starren Strukturen zu agilen Strukturen, Springer Verlag, Berlin, Heidelberg, 2013.

[MD13] Martakis A., Daneva M.: Handling Requirements Dependencies in Agile Projects: A Focus Group with Agile Software Development Practitioners, In: Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference, S. 1–11, 2013.

[Me14] Meyerbröker P.: Agiles Projektmanagement – eine Einführung, In: Spotlight – Agiles Management - ein Überblick, Berleb Media GmbH, Taufkirchen, S. 13 – 22, 2014.

[SBK14] Sandhaus G., Berg B., Knott P.: Hybride Softwareentwicklung. Das Beste aus klassischen und agilen Methoden in einem Modell vereint., Springer Vieweg, Berlin, Heidelberg, 2014.

[Sc10] Schatten A., Biffl S., Demolsky M., Gotischa-Franta E., Östreicher Th., Winkler D.: Best Practice Software-Engineering, Eine praxisorientierte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen, Spektrum Akademischer Verlag, Heidelberg, 2010.

[SH14] Stettina C. J., Hörz J.: Agile portfolio management: An empirical perspective on the practice in use, In: International Journal of Project Management 33, S. 140–152, 2014.

[SL05] Spillner A., Linz T.: Basiswissen Softwaretest, Aus- und Weiterbildung zum Certified Tester, Foundation Level nach ISQTB Standard. dpunkt.verlag, Bremen, 2005.

[Sp14] Spundak M.: Mixed agile/traditional project management methodology – reality or illusion?, In: 27th IPMA World Congress, S. 939-948, 2014.

[SP15] Serrador P., Pinto J. K.: Does Agile work? – A quantitative analysis of agile project success, In: International Journal of Project Management 33, S. 1040–1051, 2015.

[SU11] Softwaretest-Umfrage 2011., http://www.softwaretest-umfrage.de/2011/Vorgehensmodelle.htm, Stand: 21.05.2015.

[WV13] van Waardenburg G., van Vliet H.: When agile meets the enterprise, In: Information and Software Technology 55, S. 2154–2171, 2013.