Klassische versus agile Softwareentwicklung – Kriterien für die Wahl der geeigneten Methode

als
Abschlussarbeit des CAS ICT-Beschaffungen
an der
Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Universität Bern

eingereicht bei
Reto Maduz und Dr. Matthias Stürmer
Forschungsstelle Digitale Nachhaltigkeit
Institut für Wirtschaftsinformatik

von
Dippon, Martin A.
von Suhr (AG)

Studienadresse
Drusbergstrasse 13
8703 Erlenbach ZH

Bern, 28. August 2015

Zusammenfassung

Neben den mittlerweile etablierten, klassischen Methoden zur Vorgehensweise in Softwareentwicklungs-Projekten kamen in den letzten Jahren agile Methoden auf. Die Erfahrung zeigt, dass richtig eingesetzt, die Vorgehensweise die Aussicht auf Projekterfolg markant steigern kann. In sogenannten „hybriden“ Methodenansätzen werden die Stärken aus dem klassischen und agilen vereint und neu kombiniert. Nach welchen Gesichtspunkten lässt sich jedoch festlegen, welches Verfahren im konkreten Fall besser geeignet ist?

In dieser Arbeit werden innerhalb des hybriden Ansatzes nach HERMES 5 elf Kriterien zur Differenzierung zwischen klassischem und agilem Ansatz herausgeschält, nach ihrer Bedeutung gewichtet und in einem Entscheidungskatalog zusammengefügt. Die Auswertung anhand des numerischen Resultats aus diesem Hilfsmittel gibt Auskunft über das am besten geeignete Verfahren zur Umsetzung des betrachteten Projekts, aber auch allfällige weitere Hinweise zum Vorhaben.

Anhand von drei Projekten aus dem Projektportfolio von MeteoSchweiz wird die Tauglichkeit des Entscheidungskatalogs demonstriert und anschliessend beurteilt.

Inhaltsverzeichnis

ZUSAMMENFASSUNG III
INHALTSVERZEICHNIS V
ABBILDUNGS- UND TABELLENVERZEICHNIS VI

1 EINLEITUNG 1
1.1 Problemstellung 1
1.2 Zielsetzung & Schwerpunkt 2
1.3 Aufbau & methodisches Vorgehen 2
2 METHODENANSÄTZE IM VERGLEICH 3
2.1 Klassische Vorgehensweise 3
2.2 Agiler Methodenansatz 5
2.3 Hybrider Methodenansatz 7
3 ENTWURF EINES ENTSCHEIDUNGSKATALOGS 8
3.1 Zusammenstellung der Beurteilungskriterien 8
3.2 Gewichteter Entscheidungskatalog 9
3.3 Umsetzung in ein elektronisches Hilfsmittel 11
4 DER ENTSCHEIDUNGSKATALOG IM EINSATZ 12
4.1 NinJo 12
4.2 COSMO-next 13
4.3 SM/\RTneo 15
5 SCHLUSSFOLGERUNGEN 16

ANHANG A: GRUNDZÜGE DES WASSERFALLMODELLS 19
ANHANG B: DAS AGILE MANIFEST 21
ANHANG C: DIE GRUNDZÜGE VON SCRUM 22
ANHANG D: GRUNDZÜGE VON HERMES 5 24
ANHANG E: HERMES 5 AGIL MIT SCRUM 26

LITERATURVERZEICHNIS 29
SELBSTÄNDIGKEITSERKLÄRUNG 31
VERÖFFENTLICHUNG DER ARBEIT 33

Abbildungs- und Tabellenverzeichnis

Abbildung 1: elektronische Umsetzung des Entscheidungskatalogs 11
Abbildung 2: Beurteilung aus Entscheidungskatalog für NinJo 13
Abbildung 3: Beurteilung aus Entscheidungskatalog für COSMO-next 14
Abbildung 4: Beurteilung aus Entscheidungskatalog für SMARTneo 15
Abbildung 5: Erweitertes Wasserfall-Modell nach Royce 19
Abbildung 6: Abläufe in SCRUM 23
Abbildung 7: Module in HERMES 5 24
Abbildung 8: Phasen und Meilensteine in HERMES 5 24
Abbildung 9: Verteilung von Ergebnissen auf die Phasen in HERMES 5 25
Abbildung 10: HERMES 5 mit agiler Entwicklung 26

Tabelle 1: Stärkenprofil klassischer Methoden 4
Tabelle 2: Stärkenprofil agiler Methoden 6
Tabelle 3: Mögliche Ausprägungen der ausgewählten Kriterien 9
Tabelle 4: Gewichtung der ausgewählten Kriterien 10
Tabelle 5: Bezeichnung Methodenelemente HERMES - SCRUM 27
Tabelle 6: Bezeichnung Rollen HERMES - SCRUM 27

1 Einleitung

Bis in die späten 60-er Jahre wurden Informatikprojekte ohne Planung und Methode durchgeführt. Ein in der Softwareentwicklung als „Code and Fix“ bezeichneter Ansatz bewirkte, dass bei grösseren Projekten die Möglichkeit der Softwareerweiterung oder der Fehlersuche rasch verloren ging (Weiss & Tan 2004, S. 3). Im Bereich der Softwareentwicklung werden daher seit den 70-er Jahren Methoden zur Strukturierung des Entwicklungsprozesses aufgebaut. Definierte Prozesse und Methoden ermöglichen dabei eine effektivere und planbarere Softwareentwicklung. Heute besteht für (ICT-)Projekte eine grosse Auswahl an Methodenansätzen. Diese werden in phasen-/prozessorientierte („klassische“) und agile Ansätze unterteilt.

Agile Vorgehensweisen haben in den vergangenen Jahren an Bedeutung gewonnen und sich auf Kosten der klassischen Ansätze immer mehr durchgesetzt (Wagener 2015, S. 3). Nahezu dogmatisch wurden die beiden unterschiedlichen Ansätze einseitig verteidigt und gegeneinander ausgespielt (Kraft & Kirchhof 2011, S. 10), anstatt die Vorteile gewinnbringend in hybriden Vorgehensmodellen kombiniert. Mit HERMES 5.1 verfügt die Bundesverwaltung über eine hybride Projektmanagementmethode. Es zeigt sich, dass mit der jeweils besser geeigneteren Vorgehensweise nicht nur die Effizienz gesteigert und Aufwände optimiert werden können, sondern auch dass die Wahl erfolgskritisch für das gesamt Projekt sein kann (Schneegans 2012, S. 4). Eine von der Deutschen Gesellschaft für Projektmanagement in Auftrag gegebene Studie in Erfahrung gebracht, dass klassische Vorgehensweisen eine Steigerung der Erfolgswahrscheinlichkeit auf rund 40% und agile Vorgehensweisen bis zu 67% bewirken können (Vgl. Toth et al. 2009, S. 7).

1.1 Problemstellung

Wenn die Vorgehensweise einen so hohen Einfluss auf die Erfolgswahrscheinlichkeit hat, dann sollte die Methodenwahl objektiv gut begründet durchgeführt werden. Zumal beispielsweise die Projektaufsetzung oder auch der Zeitplan je nach Vorgehensweise anders ausfällt. Nach welchen Ge-

sichtspunkten lässt sich jedoch der am besten geeignete Ansatz festlegen? Auf welche Kriterien können sich Projektleitende von ICT-Vorhaben für die Wahl abstützen?

1.2 Zielsetzung & Schwerpunkt

Ein Entscheid zur Wahl der geeigneten Vorgehensweise innerhalb eines (ICT-)Vorhabens basiert weitgehend auf subjektiven Kriterien. In dieser Arbeit wird eine auf Kriterien basierte, objektivierte und somit auch nachvollziehbare Entscheidungshilfe aufgebaut. Der Schwerpunkt der vorliegenden Arbeit ist daher aussagekräftige Entscheidungskriterien zu bestimmen. Diese werden versehen mit beurteilbaren Ausprägungen und einer Gewichtung, um so die am besten geeignete Vorgehensweise festlegen zu können.

1.3 Aufbau & methodisches Vorgehen

In Kapitel 2 erfolgt zunächst die Aufbereitung der Methodengrundsätze aus der Theorie heraus. Dabei werden die jeweiligen Stärken herausgeschält. Auf eine vollständige SWOT-Analyse wird verzichtet, da für die hier angestellten Betrachtungen nur die Stärken relevant sind. Die Stärken werden in der Folge mit Ausprägungen ihres Auftretens ergänzt zu aussagekräftigen Entscheidungskriterien und mit einer begründeten Gewichtung versehen. Es entsteht somit eine numerische Gegenüberstellung der Vorgehensmethoden, basierend auf dem ausgearbeiteten Kriterienkatalog.

Anhand von Praxisbeispielen wird in Kapitel 4 der Einsatz des Kriterienkatalogs demonstriert und die Aussagekraft der Entscheidungshilfe verifiziert. Schlussfolgerungen zum Einsatz der Entscheidungshilfe und somit der Wahl der geeigneten Vorgehensmethode erfolgen in Kapitel 5.

Vor- und nachgelagerte Prozesse innerhalb der Softwareentwicklung im Rahmen eines Projekt werden in dieser Arbeit nicht berücksichtigt.

2 Methodenansätze im Vergleich

2.1 Klassische Vorgehensweise

2.1.1 Allgemeine Grundprinzipien

Der Begriff “klassische Vorgehensweise“ lässt sich nicht allgemeinverbindlich definieren. Gemäss Wagener ist „klassisch“ im Zusammenhang mit Projektmanagement eher als „nicht agil“ zu verstehen (Vgl. Wagener 2015, S. 3). Weitgehend identisch ist bei dieser Vorgehensweise, dass ab Beginn nach SMART[1] formulierte Ziele vorliegen, welche in einem Projekt mit Meilensteinen und weitgehend voneinander getrennten Phasen abgewickelt werden (Schneegans 2012, S. 3). Auf die Einhaltung des weitgehend strikten Prozesses wird Gewicht gelegt und die Planungsphase fällt ausführlich aus. In allen Phasen wird eine ausführliche Projektdokumentation geführt, welche in der jeweils folgenden Phase als Grundlage dient (Weiss & Tan 2004, S. 3).

Klassische Methoden beschreiben somit, „wie“ etwas während eines Projektes gemacht werden soll und „was“ genau in welchem Dokument zu einer Zeit beschrieben sein muss, so dass zum Beispiel Software entwickelt werden kann. Durch die eher dokumentenlastige, klar strukturiert und zeitlich gründlich festgelegte Planung werden klassische Vorgehensweisen auch als „heavyweight methods“ bezeichnet (Neus, S. et al. 2013a).

2.1.2 Vertreter des klassischen Methodenansatzes

Als typische Vertreter des klassischen Ansatzes gelten, das Wasserfallmodel, das Spiralmodell oder der Rational Unified Process (RUP). Das Wasserfallmodell ist in seinen Grundprinzipien in Anhang A erläutert.

2.1.3 Stärkenprofil

Aufgrund der bisher gemachten Aussagen und den Erläuterungen in Anhang A können Methoden nach dem klassischen Ansatz in ihren Stärken im Allgemeinen und vereinfacht wie folgt charakterisiert werden:

[1] SMART steht für spezifisch, messbar, aktiv, realistisch und terminiert

	Aussage
Fachinhaltliche Aspekte	
Auftrag	• durch Auftraggeber zugeteilt • weitgehend konstant über das Gesamtprojekt
Ziel	• ab Beginn nach SMART festgelegt • robust und einzig über formelle Anträge veränderbar
Lieferobjekte	• Dokumentation • Produkt basiert auf Anforderungen der Konzeptphase
Schnittstellen	• beliebige Anzahl und beliebig verteilt
Technologie-stand	• eher basierend auf bewährtem Standard
Organisatorische Aspekte	
Steuerung	• hierarchisch für das Gesamtprojekt • Austausch möglich ohne wesentlichen Einfluss auf Ziele und Vorgehen
Führung	• weitgehend straffe Führung bis auf Stufe Team • Austausch Beteiligte jederzeit möglich
Ausführung intern	• beliebige Teamgrösse und beliebig verteilt • Austausch Beteiligte jederzeit möglich
Ausführung extern	• beliebige Teamgrösse und beliebig verteilt • Austausch Beteiligte jederzeit möglich
Stakeholder / Kunden	• Kontakt mehrheitlich bei Erreichen von Meilensteinen und/oder Zielabweichungen
Vorgehensplan	
Zeitplan	• Planung als zentraler Aspekt • Phasen sind einzuhalten, da aufeinander aufbauend • Meilensteine oftmals im Sinne von Quality-Gates
Finanzen	
Aufwand-kontrolle	• in der Regel jeweils bei Phasenabschluss
Änderungs-kosten	• je länger das Projekt bereits andauert, umso höher die Kosten für Änderungen

Tabelle 1: Stärkenprofil klassischer Methoden

2.2 Agiler Methodenansatz

2.2.1 Allgemeine Grundprinzipien

Die Motivation des agilen Ansatzes wird bereits durch die Namenswahl verdeutlicht: Agil bedeutet flink oder beweglich. Agile Methoden konzentrieren sich darauf, kurzfristig auf Begebenheiten reagieren zu können. Im Hinblick auf Anforderungsmanagement zeigt sich, dass agile Methoden grundsätzlich keine Dokumentationsweise festlegen. Es ist festgehalten, „was“, nicht „wie“ etwas durchgeführt werden soll. Neue Anforderungen und die Änderung von Anforderungen sind zu jeder Zeit willkommen (Neus et al. 2013a). Diesen Umstand belegt auch eines der grundlegenden Prinzipien des Agilen Manifests: „*Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage"* (Vgl. Beck et al. 2001a). Das vollständige Agile Manifest ist in Anhang B aufgeführt.

2.2.2 Vertreter des agilen Methodenansatzes

Als typische bzw. verbreitete Vertreter des agilen Ansatzes gelten SCRUM, PRINCE2 oder auch Extreme Programming (XP). Die Grundprinzipien des Ansatzes nach SCRUM sind in Anhang C erläutert.

2.2.3 Stärkenprofil

Aufgrund der bisher gemachten Aussagen und den Erläuterungen in Anhang B und C können Methoden nach dem agilen Ansatz in ihren Stärken im Allgemeinen und vereinfacht wie folgt charakterisiert werden:

	Aussage
Fachinhaltliche Aspekte	
Auftrag	• wird zusammen mit dem Kunden definiert und regelmässig überprüft bzw. angepasst • oftmals mit vielen Freiheitsgraden versehen
Ziel	• weitgehend dynamisch, oftmals basierend auf just-in-time Spezifikation • häufige Anpassungen

Lieferobjekte	• Produkt-Inkremente • Endprodukt basierend auf regelmässigen kundenseiteigen Rückmeldungen
Schnittstellen	• direkter Kontakt zu Kunden/Auftraggeber • sonst möglichst wenig Abhängigkeiten
Technolgiestand	• gut verträglich mit hohem Innovationsgrad
Organisatorische Aspekte	
Steuerung	• agile (Teil-)Projekte mit nahezu Blackbox-Elementen • Austausch jederzeit möglich jedoch mit allfälliger Änderungen der Prioritäten und Zielvorstellung verbunden
Führung	• bewahrt Team vor Ablenkungen/Hindernissen • Hüter der Methode
Ausführung intern	• selbstorganisiertes Arbeiten • möglichst frei von Linienaufgaben • effiziente Teamgrösse bei rund 5 Mitgliedern
Ausführung extern	• möglichst wenige, dafür kleine oder keine externen Teams, andernfalls geringe bis keine Abhängigkeiten
Stakeholder / Kunden	• regelmässige Absprache mit Kunde • beeinflusst Produkt-Inkremente stark
Vorgehensplan	
Zeitplan	• Qualität wichtiger als Termin • Gesamtzeitplan kaum von Bedeutung
Finanzen	
Aufwandkontrolle	• Rückschlüsse über die Burndown Chart
Änderungskosten	• weitgehend konstant über die Projektdauer

Tabelle 2: Stärkenprofil agiler Methoden

2.3 Hybrider Methodenansatz

2.3.1 Allgemeiner Grundansatz

Üblicherweise werden Projekte entweder agil oder klassisch aufgesetzt und durchgeführt. Dabei stehen sich die jeweiligen Befürworter meist äusserst skeptisch gegenüber. Kraft & Kirchhof (2011, S. 7) sehen jedoch in der Anwendung von hybriden Vorgehensmodellen, also einer Mischform zwischen klassischem und agilem Ansatz, den erfolgskritischen Aspekt in der Integration einzelner Teilprojekte in ein Gesamtprojekt und der Zusammenarbeit zwischen Teilprojekten. So werden im Hybrid die jeweils optimalen Ansätze nach klassischem und agilem Ansatz in geeigneter Form kombiniert.

Durch die Kombination der beiden methodischen Ansätze in einem Hybrid erscheint jedoch das Projektmanagement auf den ersten Blick komplizierter, werden doch parallel unterschiedliche Vorgehensweisen praktiziert und scheinbar unvereinbare Führungsansätze angewendet (Schneegans 2012, S. 9). Durch geschickte Kommunikation des jeweiligen Arbeitsauftrags und dem Gewähren von Freiheiten bei der Arbeitserledigung, ohne dabei jedes Detail als Projektleiter nachvollziehen zu können und einem gesunden Mass an Vertrauen, lassen sich die Methoden in der Praxis dennoch gut kombinieren (Kraft & Kirchhof 2011, S. 11).

2.3.2 Stärkenprofil

Die in Kapitel 2.1.3 und 2.2.3 identifizierten Stärken stehen sich im hybriden Ansatz oftmals diametral gegenüber. Dies bedeutet, dass in falscher Kombination eine vermeintliche Stärke in Tat und Wahrheit zu einer Schwäche wird und so ein Risikofaktor für die effiziente Zielerreichung darstellt. Die Herausforderung bei der hybriden Vorgehensweise ist daher, die am besten geeignete Kombination der jeweiligen Vorteile im konkreten Anwendungsfall zu erkennen (Kraft & Kirchhof 2011, S. 2).

2.3.3 Vertreter des hybriden Methodenansatzes

Hybride Methodenansätze werden nach ihrem Ursprung aus dem klassischen oder agilen Umfeld bezeichnet und haben in der Regel keine eigenen Namen. So kommt zum Beispiel in der Bundesverwaltung grundsätzlich HERMES 5, ein Vertreter des klassischen phasenorientierten Ansatzes, zum Einsatz. HERMES 5 ist seit der Version 5.1 mit Agilität (SCRUM) ergänzt und kann so auch als Hybrid eingesetzt werden (ISB 2014, S. 165). Die Grundzüge von HERMES 5 können Anhang D entnommen werden, und in Anhang E ist der Einsatz von SCRUM innerhalb von HERMES 5 erläutert.

3 Entwurf eines Entscheidungskatalogs

3.1 Zusammenstellung der Beurteilungskriterien

In Kapitel 2.1.3 und 2.2.3 identifizierte Stärken können grundsätzlich als Beurteilungskriterien betrachtet werden. Um anhand der Auswahlkriterien eine Vorgehensentscheidung herbeiführen zu können, sind jedoch in der weiteren Betrachtung nur noch jene Kriterien massgebend, welche eine klare Differenzierung zwischen den zur Auswahl stehenden Methodenansätzen innerhalb einer hybriden Umsetzung erlauben.

Die verbleibenden Auswahlkriterien sind um die jeweilige Ausprägung zu ergänzen. Dabei ist zu berücksichtigen, dass der Vorgehensentscheid in der Praxis oftmals mit noch vagen Projektinformationen getroffen werden muss. Die möglichen Entscheidungskriterien und deren Ausprägung sind dahingehend geschärft und in unten aufgeführter Tabelle entsprechend ergänzt.

Auch gilt es im Unternehmen sicherzustellen, dass die Voraussetzungen für den Einsatz von verschiedenen Methodenansätzen auch aus praktischer Sicht gegeben sind. Zum Beispiel muss vorausgesetzt sein, dass zu beiden Methoden sowohl genügend Fachwissen als auch die organisatorischen Strukturen vorhanden sind. So sind denn auch die Auswahlkriterien darauf ausgerichtet eine unmittelbare Umsetzung vornehmen zu können und nicht die Voraussetzungen für die bereits getroffene Methodenwahl zu schaffen.

	Ausprägung	
	spricht für klassisches Vorgehen	spricht für agiles Vorgehen
Fachinhaltliche Aspekte		
Auftrag	hohe Stabilität	hohe Flexibilität
Ziele	weitgehend fix (SMART)	hohe Änderungsdynamik
Lieferobjekte[2]	Dokumentation	Inkremente
Schnittstellen	viele	wenige
Innovationsgrad	tief	hoch
Organisatorische Aspekte		
Steuerung	hierarchisch	Nähe zur Ausführung
Führung	Institutionelles Rollenverständnis	koordinative Aktivitäten
Ausführung intern	abhängig von Führung	autonom und isoliert
Ausführung extern	viele und mit teileweise hohem Einfluss	möglichst wenige oder keine und wenig Einfluss
Stakeholder / Kunden	Kontakt bei Meilensteinerreichung & Abweichungen	häufiger & regelmässiger Kontakt mit internem Team
Vorgehensplan		
Zeitplan	von zentraler Bedeutung	Qualität vor Zeit
Finanzen		
Änderungskosten	je länger, je höher	weitgehend konstant

Tabelle 3: Mögliche Ausprägungen der ausgewählten Kriterien

3.2 Gewichteter Entscheidungskatalog

Die Gewichtung hat zum Ziel, dass die entscheidenden Kriterien in der jeweiligen Punktwertung mehr Bedeutung bekommen. Sie ist pro Kriterium festgelegt und für beide methodischen Ansätze symmetrisch angesetzt. Ausschlaggebend für die Gewichtung ist der geschätzte generische Einflussgrad

[2] Unterschiede während dem Projekt, d.h. abgesehen vom Endprodukt

auf die erfolgreiche Zielerreichung innerhalb der jeweils gegebenen äusseren Rahmenbedingungen.

Einzelne bisher aufgeführte Kriterien stehen in direkter Abhängigkeit zueinander. Um zu vermeiden, dass einzelne Kriterien in unterschiedlichen Rubriken mehrfach beurteilt werden, sind die unten aufgeführten Kriterien zusätzlich auf direkte Abhängigkeiten geprüft und angepasst. So fällt zum Beispiel das Kriterium „Änderungskosten“ aus dem Entscheidungskatalog, da der Aspekt der Änderungen bereits implizit in der Zielfixierung abgedeckt ist.

Der verbleibende Katalog umfasst somit noch die folgenden 11 Kriterien:

	Gewichtungsfaktor			Begründung
	3	2	1	
Fachinhaltliche Aspekte				
Auftragsfixierung	X			wesentliche Differenzierung zwischen den beiden Vorgehensweisen
Zielfixierung	X			
Lieferobjekte	X			
Schnittstellen		X		Charakterisierung eher schwierig
Innovationsgrad	X			wesentliche Differenzierung
Organisatorische Aspekte				
Steuerung			X	Steigerungspotential bez. Projektkultur bei MeteoSchweiz nicht ausgeschöpft
Führung			X	
Ausführung intern		X		Dito Steuerung/Führung, jedoch mittelbarerer Einfluss auf die Lieferobjekte
Ausführung extern		X		
Stakeholder / Kunden	X			wesentliche Differenzierung sowie Zweckgebundenheit des Vorhabens
Vorgehensplan				
Zeitplan		X		Aus der Theorie klare Differenzierung, jedoch weniger deutlich in der Praxis

Tabelle 4: Gewichtung der ausgewählten Kriterien

Durch die Gewichtung beträgt die maximal erreichbare Punktezahl der verbleibenden Kriterien 25 Zähler (5 x 3 und 4 x 2 und 2 x 1). In der Anwendung des gewichteten Entscheidungskatalogs ist somit grundsätzlich jene Methode besser geeignet, welche in der Beurteilung eine höhere Punktzahl erreicht.

3.3 Umsetzung in ein elektronisches Hilfsmittel

Um eine Berechnung zusammen mit einem visuellen Eindruck zu erhalten, wurden die verbleibenden 11 Kriterien zusammen mit den Ausprägungen in drei Kategorien auf eine Tabellenkalkulation übertragen. Versehen mit Schiebereglern, bei welchen die Gewichtung als Haupteigenschaft hinterlegt ist, lässt sich für jedes einzelne Kriterium die am ehesten zutreffende Ausprägung auswählen. Die jeweilige Punkteverteilung erfolgt anhand der Position des Schiebereglers. Dabei werden die Punkte pro Vorgehensmethode automatisch zusammengezählt. Abbildung 1 zeigt das so entstandene Hilfsmittel in seiner Grundform.

	Ausprägung "klassisch"		Ausprägung "agil"
1. Fachinhaltliche Aspekte			
Auftragsfixierung	*hohe Stabilität*		*hohe Flexibilität*
Zielfixierung	*fix (SMART)*		*hohe Änderungsdynamik*
Lieferobjekte	*Dokumentation*		*Inkremente*
Schnittstellen	*viele*		*wenige*
Innovationsgrad	*tief*		*hoch*
2. Organisatorische Aspekte			
Steuerung	*hierarchisch*		*nahe bei Ausführung*
Führung	*institutionelle Rollen*		*koordinative Aktivitäten*
Ausführung intern	*abhängig von Führung*		*autonom und isoliert*
Ausführung extern	*viele (und hoher Einfluss)*		*wenige bzw. kaum Einfluss*
Stakeholder- / Kundenkontakt	*bei MS & Abweichungen*		*häufig & regelmässig*
3. Vorgehensplan			
Zeitplan	*Einhalten Zeitplan im Fokus*		*Qualität vor Zeitplan*
Kriterienpunkte (max. 25)	**0**		**0**

Abbildung 1: elektronische Umsetzung des Entscheidungskatalogs

Anhand der Punktesumme lässt sich die voraussichtlich am besten geeignete Vorgehensweise ablesen. Durch die jeweilige Position der Schieberegler wird jedoch auch augenfällig, in welchen Ausprägungen die Aspekte jeweils vorliegen.

4 Der Entscheidungskatalog im Einsatz

In den folgenden Kapiteln wird der Entscheidungskatalog anhand von drei Projekten aus dem Portfolio von MeteoSchweiz zur Wahl des am besten geeigneten Vorgehens innerhalb des hybriden Ansatzes nach HERMES 5, Version 5.1, angewendet und das Resultat in Bezug auf die Aussagequalität bewertet.

4.1 NinJo

4.1.1 Fallbeschreibung

NinJo ist ein modernes meteorologisches Arbeitsplatzsystem zur Visualisierung verschiedenster meteorologischer Daten und zur Generierung von Produkten, wie z.B. von Wetterwarnungen. Es ist ein komplexes und hoch konfigurierbares System, das mittlerweile über 15‘000 Java Klassen mit rund 3.3 Mio. Zeilen Code und 3.5 Mio. Zeilen XML-Konfiguration umfasst. NinJo wurde im Rahmen einer internationalen Partnerschaft (NinJo-Konsortium) aufgebaut und wird in den kommenden Jahren noch weiterentwickelt. NinJo steht bei MeteoSchweiz seit 2010 im operationellen Prognosedienst während 24 Stunden an 365 Tagen im Jahr im Einsatz.

Folgende Herausforderungen sind in einem Projekt anzugehend:

- Performance-Issue durch Anzahl parallellaufender Prozesse;
- Standardisierung der Hardware-Architektur gem. NinJo-Konsortium;
- Implementierung des Konsortium-Releases 1.9;
- Anpassung der Betriebsorganisation gem. Vorgaben aus dem Business Continuity Management von MeteoSchweiz.

4.1.2 Festlegung des Vorgehens anhand des Entscheidungskatalogs

Eine erste Beurteilung anhand der oben aufgeführten Fallbeschreibung führt zu folgendem Resultat:

	Ausprägung "klassisch"		Ausprägung "agil"
1. Fachinhaltliche Aspekte			
Auftragsfixierung	*hohe Stabilität*		*hohe Flexibilität*
Zielfixierung	*fix (SMART)*		*hohe Änderungsdynamik*
Lieferobjekte	*Dokumentation*		*Inkremente*
Schnittstellen	*viele*		*wenige*
Innovationsgrad	*tief*		*hoch*
2. Organisatorische Aspekte			
Steuerung	*hierarchisch*		*nahe bei Ausführung*
Führung	*institutionelle Rollen*		*koordinative Aktivitäten*
Ausführung intern	*abhängig von Führung*		*autonom und isoliert*
Ausführung extern	*viele (und hoher Einfluss)*		*wenige bzw. kaum Einfluss*
Stakeholder- / Kundenkontakt	*bei MS & Abweichungen*		*häufig & regelmässig*
3. Vorgehensplan			
Zeitplan	*Einhalten Zeitplan im Fokus*		*Qualität vor Zeitplan*
Kriterienpunkte (max. 25)	**6**		**15**

Abbildung 2: Beurteilung aus Entscheidungskatalog für NinJo

4.1.3 Bewertung des Resultats

Die Verteilung der Punktzahlen auf die einzelnen Aspekte sowie auch die Gesamtpunktzahl lassen darauf schliessen, dass das Projekt NinJo für eine agile Vorgehensweise geeignet wäre. Die Rahmenbedingungen (NinJo-Konsortium) bedingen jedoch auch die Berücksichtigung von klassischen Aspekten, insbesondere bei der Pflege der Schnittstellen und in der Berücksichtigung des Zeitplans. Es gilt daher zu klären, ob z.B. Einfluss auf den Zeitplan genommen werden kann, um mehr Zeit für die Sicherstellung der Qualität zu erhalten.

4.2 COSMO-next

4.2.1 Fallbeschreibung

MeteoSchweiz hat das Projekt COSMO-NExT gestartet, um das System für numerische Wettervorhersagen weiter zu entwickeln. Das System soll 2016 den operationellen Betrieb aufnehmen und umfasst folgende Teilprojekte:

- COSMO-1 (deterministische Vorhersagen mit Maschenweite 1.1 km und Aktualisierungszyklus von achtmal täglich bis zu + 24 Std.)

- COSMO-E (Ensemble-Prognosen, welche zweimal täglich bis zu + 120 Std. mit einer Maschenweite von 2.2 km)
- Datenassimilation (neues Ensemble-basiertes Daten-Assimilationssystem sowie quasi-optimale und wetterabhängige Kombination von Beobachtungen und Modellprognosen basierend auf Fehlerstatistik)

4.2.2 Festlegung des Vorgehens anhand des Entscheidungskatalogs

Eine erste Beurteilung anhand der oben aufgeführten Fallbeschreibung führt zu folgendem Resultat:

1. Fachinhaltliche Aspekte		
Auftragsfixierung	*hohe Stabilität*	*hohe Flexibilität*
Zielfixierung	*fix (SMART)*	*hohe Änderungsdynamik*
Lieferobjekte	*Dokumentation*	*Inkremente*
Schnittstellen	*viele*	*wenige*
Innovationsgrad	*tief*	*hoch*
2. Organisatorische Aspekte		
Steuerung	*hierarchisch*	*nahe bei Ausführung*
Führung	*institutionelle Rollen*	*koordinative Aktivitäten*
Ausführung intern	*abhängig von Führung*	*autonom und isoliert*
Ausführung extern	*viele (und hoher Einfluss)*	*wenige bzw. kaum Einfluss*
Stakeholder- / Kundenkontakt	*bei MS & Abweichungen*	*häufig & regelmässig*
3. Vorgehensplan		
Zeitplan	*Einhalten Zeitplan im Fokus*	*Qualität vor Zeitplan*
Kriterienpunkte (max. 25)	**5**	**10**

Abbildung 3: Beurteilung aus Entscheidungskatalog für COSMO-next

4.2.3 Bewertung des Resultats

Aufgrund der Verteilung der erreichten Punktzahl aus der Festlegung in Kapitel 4.2.2 lässt sich festhalten, dass eine agile Vorgehensweise zumindest für einzelne Aspekte oder Teilprojekte in Betracht gezogen werden sollte. Die relativ geringe Punktzahl in der agilen Ausprägung im Verhältnis zur möglichen Gesamtpunktzahl lässt aber vermuten, dass auch indifferente sowie klassische Aspekte vorliegen. Um eine klareres Bild zu erhalten wäre der nächste Schritt die Zerlegung des Projekts auf seine Teilprojekte und der anschliessenden Beurteilung der einzelnen Teilprojekte. Im Rahmen dieser Arbeit wird jedoch auf die Ausführung der nächsten Schritte verzichtet.

4.3 SM/\RTneo

4.3.1 Fallbeschreibung

SM/\RT (**S**ystem for **M**eteorological **A**utomated **R**epor**t**ing) ist eine Plattform für Echtzeit-Datenerfassung, -verarbeitung und -visualisierung. Folgende Funktionsbereiche werden unterschieden:

- Editor mit Systemvorschlägen und Überprüfung von manuellen Eingaben
- Viewer mit Visualisierung aller Mess- und aller berechneten Grössen
- Management Tool mit Sicht in das laufende System

Die Architektur SMART besteht aus 24 UNIX Servern. Hardware und Betriebssoftware sind ausnahmslos end of life und der Ersatz soll im Rahmen eines Projekts geplant, priorisiert und kontrolliert erfolgen. Die Ziele des Projekts lauten voraussichtlich:

- Zentralisierung der Server (mit Redundanz in ZRH und GVA)
- Migration Middleware auf LINUX (Strategie MeteoSchweiz)
- Bestehende hohe Betriebssicherheit aufrechterhalten

4.3.2 Festlegung des Vorgehens anhand des Entscheidungskatalogs

Eine erste Beurteilung anhand der oben aufgeführten Fallbeschreibung führt zu folgendem Resultat:

	Ausprägung "klassisch"		Ausprägung "agil"
1. Fachinhaltliche Aspekte			
Auftragsfixierung	*hohe Stabilität*		*hohe Flexibilität*
Zielfixierung	*fix (SMART)*		*hohe Änderungsdynamik*
Lieferobjekte	*Dokumentation*		*Inkremente*
Schnittstellen	*viele*		*wenige*
Innovationsgrad	*tief*		*hoch*
2. Organisatorische Aspekte			
Steuerung	*hierarchisch*		*nahe bei Ausführung*
Führung	*institutionelle Rollen*		*koordinative Aktivitäten*
Ausführung intern	*abhängig von Führung*		*autonom und isoliert*
Ausführung extern	*viele (und hoher Einfluss)*		*wenige bzw. kaum Einfluss*
Stakeholder- / Kundenkontakt	*bei MS & Abweichungen*		*häufig & regelmässig*
3. Vorgehensplan			
Zeitplan	*Einhalten Zeitplan im Fokus*		*Qualität vor Zeitplan*
Kriterienpunkte (max. 25)	**15**		**2**

Abbildung 4: Beurteilung aus Entscheidungskatalog für SMARTneo

4.3.3 Bewertung des Resultats

Sowohl die erreichte Punktzahl als auch die Verteilung der Punkte spricht in oben erstellter Beurteilung für ein klassisches Vorgehen. Aus dem ausgefüllten Entscheidungskatalog ist aber auch ersichtlich, dass offenbar ein grosses Bedürfnis an hoher Qualität besteht (Siehe Abbildung 4), was aufgrund des aviatischen Umfelds nicht weiter erstaunt. Dieser Umstand ist jedoch im Zeitplan entsprechend zu berücksichtigen. So sind zum Beispiel für Test- bzw. Korrekturverfahren und formelle Abnahmen genügend Zeit einzuplanen.

5 Schlussfolgerungen

Mehrere konsultierte Quellen weisen darauf hin, dass die Wahl der geeignetsten Vorgehensweise erfolgskritisch für ein ICT-Projekt sei. Eine eigentliche „falsche“ Methode gibt es nicht, jedoch nachweislich weniger gute, gute oder gar sehr gut geeignete Vorgehensweisen. Mit der „richtigen“ bzw. am besten geeigneten Vorgehensweise werden die Erfolgschancen markant erhöht. Mit dem Entscheidungskatalog steht ein Hilfsmittel zur Verfügung, welches basierend auf den zu Beginn des Projekts zur Verfügung stehenden Information einen objektiven Auswahlentscheid erlaubt, um die am besten geeignete Vorgehensweise festzulegen. Die Beurteilung mit dem Entscheidungskatalog liefert aber auch Hinweise, wo innerhalb des hybriden Ansatzes Aspekte aus dem weniger gut geeigneten Vorgehen zweckmässigerweise zu berücksichtigen sind.

Anhand der Bewertung der Resultate in Kapitel 4 kann geschlossen werden, dass der Entscheidungskatalog als Hilfsmittel taugt. Die jeweilige Auswahl der am besten geeigneten Methode scheint anhand der aufgeführten Beispiele als plausibel.

Der mögliche Einsatz der vorgestellten Methoden erfolgt fokussiert anhand der Betrachtung der jeweiligen Stärkenprofile. Es ist jedoch nochmals darauf hinzuweisen, dass die Methodenansätze auch Schwächen, Risiken sowie Chancen aufweisen, deren sich die Verantwortlichen bewusst sein sollten. Auch wurden Rahmenbedingungen vorausgesetzt, welche in einer Unter-

nehmung vorgängig aufzubauen sind. So gilt es zu beachten, dass nicht alle Projektleitenden im ICT-Umfeld die Fähigkeiten oder das Wissen mitbringen die angesprochenen Methodenansätze zweckmässig in Einzel- oder Hybridform anzuwenden. Neben der Ausbildung ist auch sicherzustellen, dass die Projektteams aus organisatorischer Sicht zweckmässig aufgestellt und eingebettet sind.

Anhang A: Grundzüge des Wasserfallmodells

Quelle: Neus et al. 2013b

Das Wasserfallmodell stammt ursprünglich aus dem Bau- und Produktionsprozess, wo späte Änderungen teuer oder gar unmöglich sind. Nachdem damals noch kein formaler Softwareentwicklungsprozess existierte, wurden die dort verwendeten Prozesse für Softwareentwicklung adaptiert. Die erste formale Beschreibung des Wasserfallmodells wird Winston W. Royce zugeschrieben, obwohl dieser in seinem 1970 erschienenen Artikel den Namen „Wasserfall“ nicht verwendete.

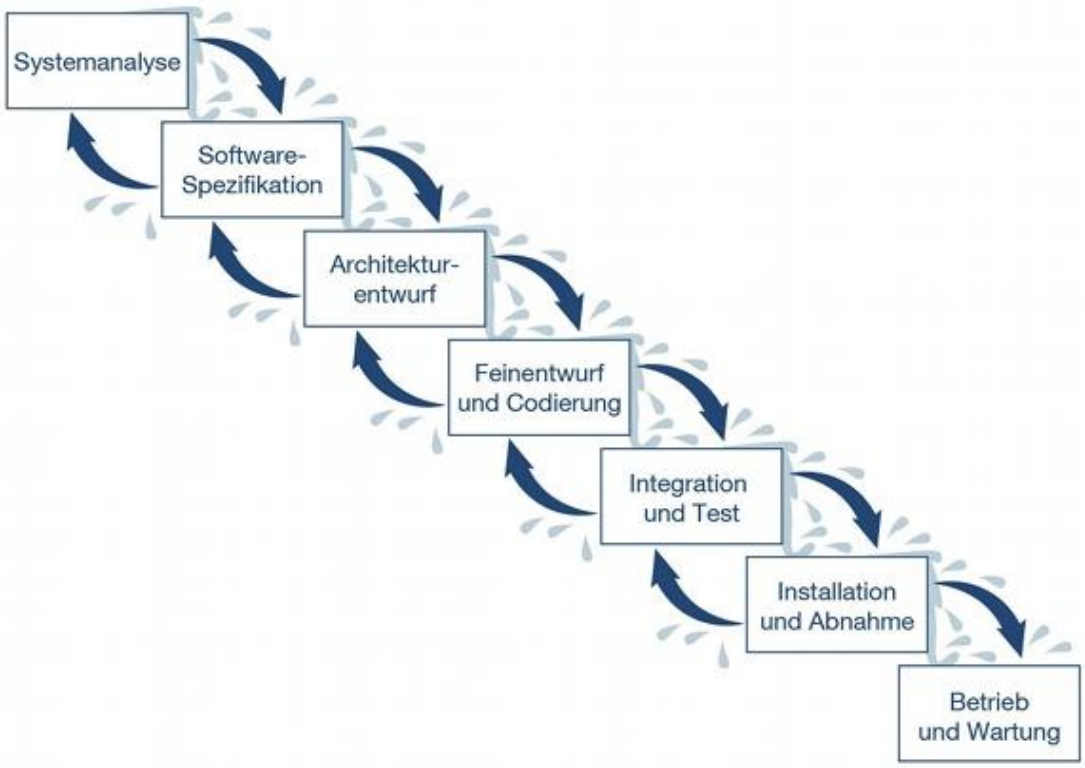

Abbildung 5: Erweitertes Wasserfall-Modell nach Royce[3]

Im Wasserfallmodell werden zu Beginn alle Anforderungen formuliert, alle weiteren Phasen bauen auf diesen Anforderungen auf. Jede Änderung bedeutet somit den Rücksprung in die vorherige Phase (Siehe Abbildung 5).

[3] Quelle: Neus et al. 2013b

Das Wasserfallmodell stellt den Ablauf zu einem realen, physischen Projekt dar, beispielsweise einem Hausbau. Zu Beginn wird das Fundament gelegt, danach die Mauern gezogen, am Ende das Dach aufgesetzt. An diesem Beispiel wird deutlich: Je später eine fehlerhafte Spezifikation erkannt wird, desto schwieriger und teurer ist die Behebung. Sollten in späten Phasen falsche Annahmen oder Mehrdeutigkeiten während der Analysephase bekannt werden, ist oft ein Projektabbruch die einzige Möglichkeit zur Schadensbegrenzung. Dies ist insbesondere bei umfangreichen Entwicklungen problematisch, da ein kompletter Durchlauf des Modells in diesen Fällen einige Jahre dauern kann. Spätestens mit Abschluss der Entwurfsphase kann kaum noch auf geänderte Anforderungen reagiert werden.

In Bezug auf Flexibilität von Anforderungen lässt sich das Wasserfall-Modell wie folgt kategorisieren:

- Die einzelnen Phasen müssen klar voneinander abzugrenzen sein und dürfen nur sequenziell nacheinander bearbeitet werden.
- Durch die sehr eingeschränkte Möglichkeit, die Ergebnisse bereits abgeschlossener Phasen nachträglich zu ändern, ist eine geringe Flexibilität auf geänderte Anforderungen möglich. Dies ist insbesondere bei langlaufenden Projekten, bei denen sich Anforderungen und auch Rahmenbedingungen ändern, problematisch. Das Modell ist »dokumentgetrieben«, es legt großen Wert auf die
- Dokumentation der Vorgänge und Ergebnisse. Dies kann einen großen Aufwand darstellen.

Anhang B: Das agile Manifest

Quelle: Beck et al. 2001a

Wir erschliessen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
- **Funktionierende Software** mehr als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
- **Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heisst, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Anhang C: Die Grundzüge von SCRUM

Im Kern basiert SCRUM auf einer inkrementellen Vorgehensweise, der Organisation von Entwicklungsabschnitten und Meetings in vordefinierten Zeitabschnitten (Time-Boxes) und der Erkenntnis, dass ein funktionierendes Produkt wichtiger ist als eine dreihundertseitige Spezifikation (Sutherland & Schwaber 2013, S. 7).

SCRUM kennt drei Rollen für direkt am Prozess Beteiligte:

- **Product Owner** (stellt fachliche Anforderungen und priorisiert sie),
- **ScrumMaster** (managt den Prozess und beseitigt Hindernisse) und
- **Team** (entwickelt das Produkt).

Daneben gibt es als Beobachter und Ratgeber noch die Stakeholders.

Die Anforderungen werden in einer Liste (Product Backlog) gepflegt, erweitert und priorisiert. Das Product Backlog ist ständig im Fluss. Um ein sinnvolles Arbeiten zu ermöglichen, wird monatlich vom Team in Kooperation mit dem Product Owner ein definiertes Arbeitspaket dem oberen, höher priorisierten Ende des Product Backlogs entnommen und komplett in Funktionalität umgesetzt. Dieses Arbeitspaket, das Increment, wird während der laufenden Iteration, des sog. Sprints, nicht durch Zusatzanforderungen modifiziert, um seine Fertigstellung nicht zu gefährden. Alle anderen Teile des Product Backlogs können vom Product Owner in Vorbereitung für den nachfolgenden Sprint verändert bzw. neu priorisiert werden.

Das Arbeitspaket wird in kleinere Arbeitspakete (Tasks) heruntergebrochen und mit jeweils zuständigem Bearbeiter und täglich aktualisiertem Restaufwand in einer weiteren Liste, dem Sprint Backlog, festgehalten. Während des Sprints arbeitet das Team konzentriert und ohne Störungen von aussen daran, die Tasks aus dem Sprint Backlog in ein Increment of Potentially Shippable Functionality, also einen vollständig fertigen und potentiell produktiv einsetzbaren Anwendungsteil, umzusetzen. Das Team gleicht sich in einem täglichen, streng auf 15 Minuten begrenzten Informations-Meeting, dem

Daily SCRUM Meeting, ab, damit jeder weiß, woran der andere zuletzt gearbeitet hat, was er als nächstes vor hat und welche Probleme es evtl. gibt. Während des Treffens stellt der ScrumMaster jedem Teammitglied die folgenden drei Fragen (Schweitzer 2003, S. 7):

- Was hast du seit dem letzten Treffen erledigt?
- Was wirst du von jetzt bis zum nächsten Treffen erledigen?
- Was hat dich bei deiner Arbeit behindert?

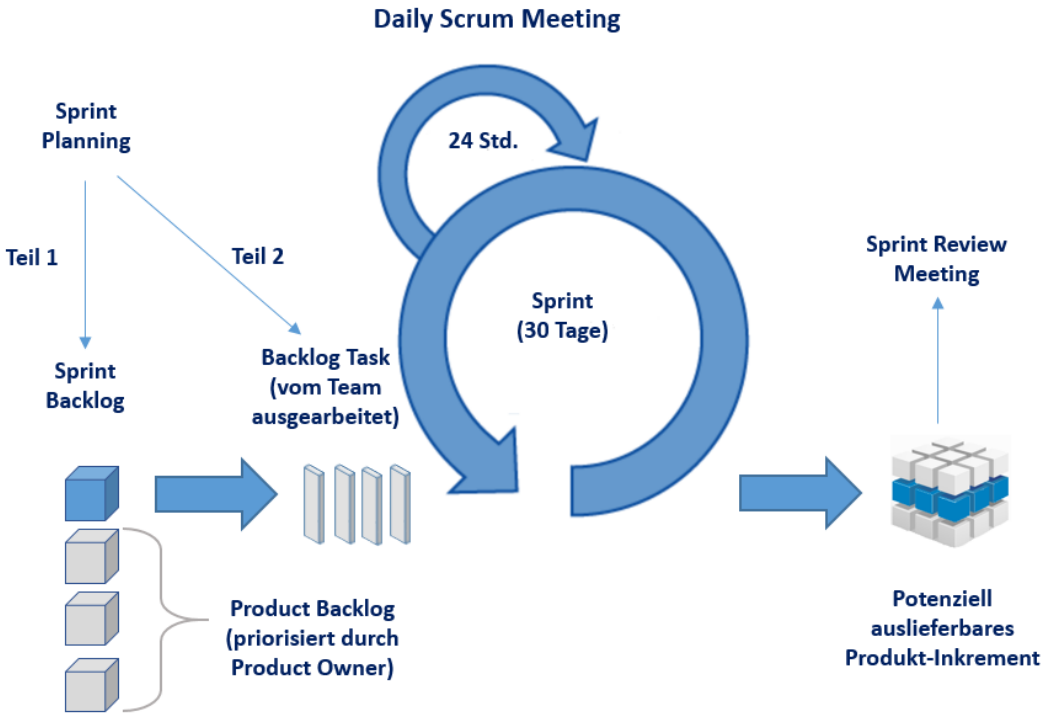

Abbildung 6: Abläufe in SCRUM[4]

Am Ende des Sprints präsentiert das Team dem Product Owner, den Stakeholders u.a. interessierten Teilnehmern in einem sog. Sprint Review Meeting live am System die implementierte Funktionalität. Das Feedback der Zuseher und die neuen Anforderungen des Product Owners für den kommenden Sprint fliessen dann wieder in das nächste Sprint Planning Meeting ein, und der Prozess beginnt von neuem (Kriegisch 2015).

[4] Quelle: Kriegisch 2015

Anhang D: Grundzüge von HERMES 5

Nachfolgend sind die Methodenelemente und ihr Zusammenspiel gemäss Referenzhandbuch HERMES 5.1 (ISB 2014, S. I, II und 1) beschrieben:

Szenarien

Ein Szenario ist auf die Durchführung von Projekten mit einer spezifischen Charakteristik ausgerichtet. Das Szenario beinhaltet genau diejenigen Methodenelemente von HERMES, welche für das Projekt von Bedeutung sind.

Module

Module sind wiederverwendbare Bausteine zur Erstellung von Szenarien. Ein Modul enthält die thematisch zusammengehörenden Aufgaben, Ergebnisse und Rollen. Sie sind den Phasen und Meilensteinen zugeordnet.

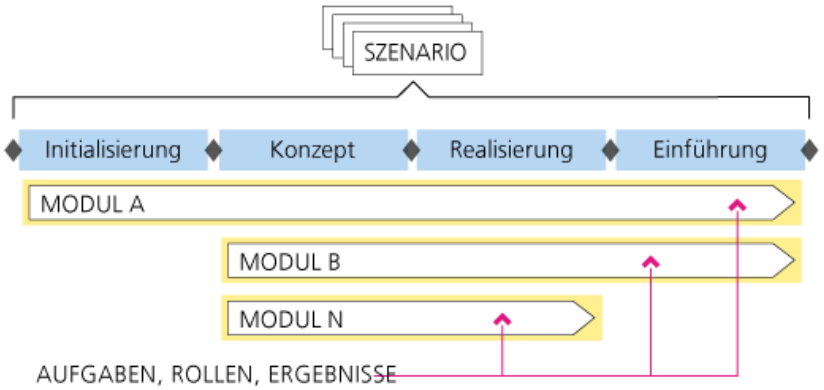

Abbildung 7: Module in HERMES 5

Phasen und Meilensteine

Das Phasenmodell bildet das Rückgrat des Projekts. Es schafft die Voraussetzung für das gemeinsame Verständnis der Projektbeteiligten betreffend den Projektablauf.

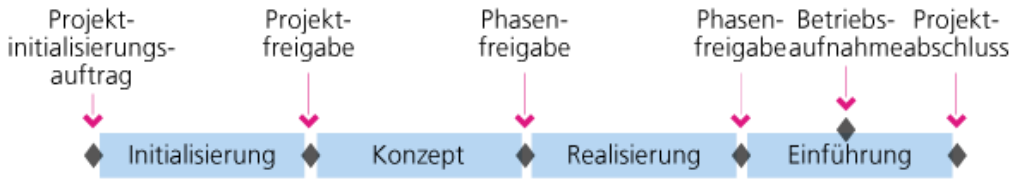

Abbildung 8: Phasen und Meilensteine in HERMES 5

Das Projekt beginnt mit der Phase Initialisierung beim Meilenstein Projektinitialisierungsauftrag und endet am Schluss der Phase Einführung beim Meilenstein Projektabschluss. Am Beginn und am Ende der Phasen stehen Meilensteine. Je nach Szenario gibt es weitere Meilensteine. Sie entsprechen Quality Gates, an denen über Ergebnisse und Vorgehen entschieden wird.

Rollen

HERMES unterscheidet Rollen aus Stamm- und Projektorganisation. Für jede Rolle der Projektorganisation gibt es eine Rollenbeschreibung. Sie definiert die Verantwortung, die Kompetenzen und die benötigten Fähigkeiten der Rolle. Jede Rolle in der Projektorganisation ist einer der Hierarchieebenen Steuerung, Führung oder Ausführung zugeordnet.

Aufgaben

Aufgaben dienen der Erarbeitung von Ergebnissen. Für jede Aufgabe gibt es eine Aufgabenbeschreibung. Sie definiert das generelle Vorgehen und die Aktivitäten, die unternommen werden, um die Ergebnisse zu erarbeiten. Jeder Aufgabe ist eine verantwortliche Rolle zugeordnet.

Ergebnisse

Ergebnisse stehen im Zentrum von HERMES.

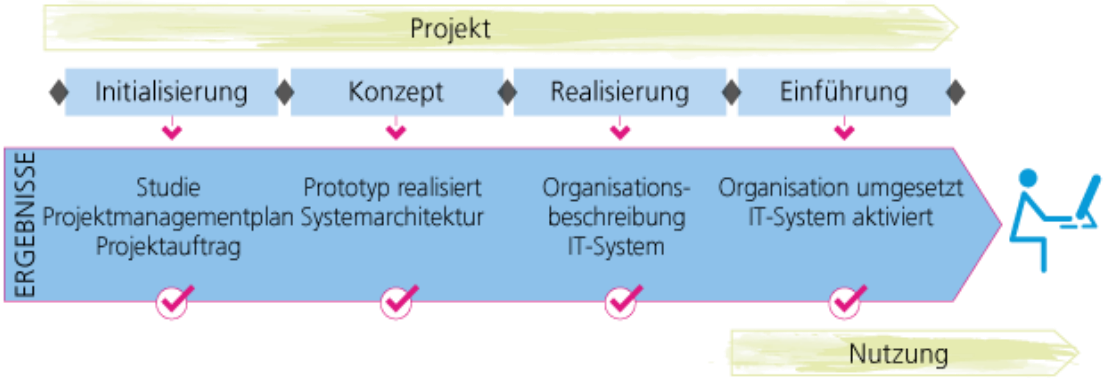

Abbildung 9: Verteilung von Ergebnissen auf die Phasen in HERMES 5

Jedem Ergebnis sind Rollen zugeordnet. Diese geben einen Hinweis auf die Beteiligung bei der Ergebniserstellung. Zudem sind Minimalergebnisse definiert, um die Anforderungen an die Projekt-Governance zu erfüllen.

Anhang E: HERMES 5 agil mit SCRUM

Quelle: ISB 2014, S. 165 - 170

HERMES deckt den gesamten Lebenszyklus des Projektes ab. SCRUM regelt die Organisation und die Steuerung des Entwicklungs-Teams. Für das Entwicklungs-Team müssen die Phasenübergänge jedoch nicht sichtbar sein. Das Entwicklungs-Team ist nur bedingt in die Steuerung des Projekts involviert, da die Steuerung des Entwicklungs-Teams über Product Backlog und Sprint-Backlog erfolgt.

Die HERMES-Phasen und die Entscheidungsaufgaben mit den Meilensteinen der Steuerung, Führung und Ausführung werden von Projektleiter und Auftraggeber nach wie vor durchlaufen. Auch bei einer agilen Entwicklung nehmen Auftraggeber und Projektleiter ihre Aufgaben der Projektsteuerung und Projektführung wahr, da diese nicht mit SCRUM abgedeckt werden.

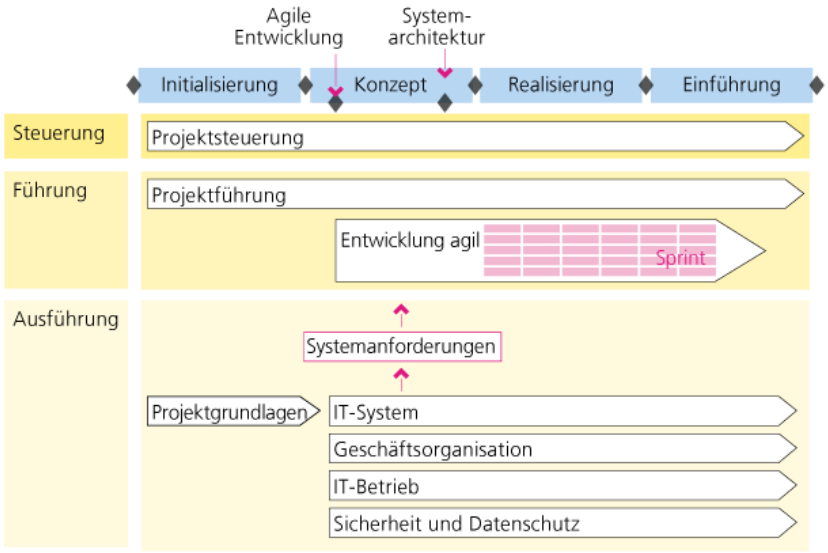

Abbildung 10: HERMES 5 mit agiler Entwicklung

Die Abbildung zeigt auf, dass entlang den HERMES-Phasen die Entwicklung agil geführt wird.

Bei der agilen Entwicklung mit SCRUM sind die Schwerpunkte des Entwicklungs-Teams in den Phasen wie folgt:

Der Schwerpunkt der agilen Entwicklung liegt in der Phase Realisierung. Bei der agilen Entwicklung erfolgt die Erarbeitung der Detailspezifikation zeitnah

zur Entwicklung. Die Entwicklung des Systems/Produkts erfolgt nach Vorliegen der jeweiligen Detailspezifikation in Sprints.
In der Phase Einführung werden weitere Sprints durchlaufen. Dabei werden z.B. Korrekturen und das Bugfixing agil geführt bis zur Systemabnahme.

HERMES bietet zwei Standardszenarien an, welche die agile Steuerung der Entwicklung mit SCRUM beinhalten:

- IT-Individualanwendung agil
- Dienstleistung/Produkt agil

Mit diesen Szenarien verfügt der Anwender über eine Methode zur agilen Steuerung der Entwicklung, die er sofort anwenden kann.

In einem Projekt, in dem die Entwicklung agil mit SCRUM geführt wird, wird SCRUM als Ergänzung zu HERMES eingesetzt.

HERMES und SCRUM bestehen aus ähnlichen Methodenelementen, sodass eine Integration des SCRUM-Frameworks in der HERMES-Methode einfach möglich ist.

Methodenelemente HERMES	Methodenelemente SCRUM
Ergebnis	Artefakt
Aufgabe	Ereignis
Rolle	Rolle

Tabelle 5: Bezeichnung Methodenelemente HERMES - SCRUM.[5]

Rolle HERMES	Rolle SCRUM
Projektleiter	Product Owner (inkl. Product Backlog-Manager)
Teamleiter oder Teilprojektleiter	ScrumMaster
(Teil-)Projektteam	Scrumteam

Tabelle 6: Bezeichnung Rollen HERMES - SCRUM[6]

[5] Quelle: ISB 2014, S. 170

[6] Quelle: Bachmann 2011, S. 16

Literaturverzeichnis

Publikationen

Bachmann, E. (2011): „*Steuerung komplexer IT-Projekte mit kombiniertem Einsatz der Methoden HERMES und SCRUM*“. Zertifikatsarbeit eingereicht der Universität Bern im Rahmen des Certificate of Advanced Studies in Public Administration (CeMaP).

Informatiksteuerungsorgan des Bundes ISB (2014): „*HERMES 5.1*“. Projektmanagementmethode für alle Projekte – Referenzhandbuch. 1. Auflage August 2014. Bern: BBL, Verkauf Bundespublikationen.

Kraft, B. & Kirchhof, M. (2011): „*Dogmatisches 'Entweder agil oder klassisch' im Projektmanagement hat ausgedient - die richtige Mischung macht's*“. Beitrag für das 28. Internationale Deutsche PM Forum 2011. GPM.

Schneegans, M. (2012): „*Klassisches versus agiles IT-Projektmanagement*“ Die Wahl der richtigen Vorgehensweise, amendos whitepaper. Hamburg: amendos gmbh.

Schweitzer, R. (2003): „Scrum – eine agile Methode zur Software Entwicklung“. Seminararbeit. Institut für Informatik der Universität Zürich.

Sutherland, J. & Schwaber, K. (2013): „ *Der SCRUM Guide*“, Der gültige Leitfaden für SCRUM: Die Spielregeln.

Toth et. al. (2009): „*Einfluss klassischer und agiler Techniken auf den Erfolg von IT-Projekten*“. 1. Ergebniszusammenfassung der Studie. Hamburg: oose.

Wagener, A. (2015): „*IT-Projektmanagement klassisch-agil*“. Agile Methoden in etablierte Strukturen einfügen. Gummersbach: Opitz Consulting.

Weiss, J. & Tan, E. (2004): „*Klassische vs. Agile Methoden der Softwareentwicklung*“. Erstellt für SWT Methoden und Werkzeuge zur Softwareproduktion.

Online-Quellen

Beck, K. et al. (2001a): „*Manifest für agile Softwareentwicklung*“, agilemanifesto.org/iso/de (Übersetzung aus dem Englischen), 05.08.2015.

Kriegisch, A (2015): „*Scrum - auf einer Seite erklärt*“. scrum-master.de /Was_ist_Scrum/Scrum_auf_einer_Seite_erklaert, 09.08.2015.

Neus, S. et al. (2013a): „*Vorgehensmodelle in der Softwareentwicklung*“, scrum-kompakt.de/grundlagen-des-projektmanagements/wasserfall-modell, 04.08.2015

Neus, S. et al. (2013b): „Wasserfallmodell“, scrum-kompakt.de/grundlagen-des-projektmanagements/wasserfall-modell, 04.08.2015

Selbständigkeitserklärung

„Ich erkläre hiermit, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe o des Gesetzes vom 5. September 1996 über die Universität zum Entzug des aufgrund dieser Arbeit verliehenen Titels berechtigt ist."

Bern, 28. August 2015 Martin A. Dippon

Veröffentlichung der Arbeit

- ☐ Hiermit erlaube ich, meine Arbeit auf der Website der Forschungsstelle Digitale Nachhaltigkeit zu veröffentlichen.
- ☐ Hiermit erlaube ich, meine Arbeit den anderen Teilnehmenden des CAS ICT-Beschaffungen (alle Jahrgänge) über ILIAS zugänglich zu machen.
- ☐ Ich möchte auf eine Veröffentlichung meiner Arbeit verzichten.

Die Benotung der Arbeit erfolgt unabhängig davon, ob die Arbeit veröffentlicht werden darf oder nicht.

Bern, 28. August 2015 Martin A. Dippon