



Agil

Das ganzheitliche Projektmanagementsystem für die öffentliche Verwaltung

Herausgebende Instanz

Bundesverwaltungsamt (BVA)

Abteilung VM

Barbarastraße 1

50728 Köln

Kontakt

Kompetenzzentrum (Groß-)Projektmanagement (CC GroßPM)

www.grosspm.bund.de

grosspm@bva.bund.de

Dokumentinformation

Version 1.01 | Juni 2023

Lizenzhinweis

Der PMflex-Agil-Leitfaden wurde erstellt auf Grundlage des PM² Agile Guide 3.0.1 der Europäischen Kommission, lizenziert unter CC BY 4.0. <https://creativecommons.org/licenses/by/4.0/>

Bildnachweis

Vectorjuice auf Freepik. Dieses Deckblatt wurde mit Ressourcen von Freepik.com erstellt

ZUSAMMENFASSUNG

Agilität gilt als die höchste Form der Anpassungsfähigkeit einer Organisation. Projekte der öffentlichen Hand und insbesondere Digitalisierungsvorhaben sind zunehmend durch vielfältige und sich im Projektverlauf schnell ändernde Anforderungen gekennzeichnet. Als öffentliche Einrichtung ist es daher wichtig, schnell und effektiv auf die Bedürfnisse der Gesellschaft reagieren zu können. Die traditionellen bürokratischen Prozesse und Strukturen können in vielen Fällen zu schwerfällig sein, um den Bedarf an schnellen und flexiblen Lösungen zu decken.

Aufgrund der sich schnell verändernden Gesellschaft in einer dynamischen Welt mit vielen verschiedenen Einflüssen und Besonderheiten in der öffentlichen Verwaltung ist der vorliegende Leitfaden relevanter denn je. Durch agile Arbeitsmethoden können Verwaltungsprozesse agiler gestaltet werden, was bedeutet, dass Teams schneller und flexibler arbeiten, um auf Veränderungen zu reagieren. Agiles Arbeiten kann dazu beitragen, die Effektivität, Effizienz und Servicequalität der öffentlichen Verwaltung zu verbessern und somit zu einer positiven Wahrnehmung führen.

Besonders hervorzuheben sind die Chancen agilen Arbeitens in der öffentlichen Verwaltung. Dazu zählen die Förderung der Kunden- und Nutzerzentrierung, Anpassungs- und Reaktionsfähigkeit, die Förderung innovativen Denkens und Arbeitens, der Abbau von Silodenken, die Erhöhung der Motivation und Identifikation mit der Arbeit der Mitarbeitenden, die Unterstützung einer positiven Fehlerkultur sowie eine Verbesserung des Wissensaustauschs unter Kolleginnen und Kollegen durch cross-funktionale Teams.¹

Der vorliegende Leitfaden übernimmt die PM²-Agile-Methodik der Europäischen Kommission unverändert, die viele andere agile Projektmanagement-Methoden, wie z. B. Scrum oder SAFe, miteinschließt und damit die Klammer für einen ganzheitlichen Projektmanagementansatz in der öffentlichen Verwaltung bildet. Der Leitfaden besteht aus zwei Teilen:

- Teil I: Agilität in der öffentlichen Verwaltung: In diesem Teil werden die Besonderheiten Herausforderungen von Agilität in der öffentlichen Verwaltung dargestellt. Außerdem werden die Erfolgsfaktoren des agilen Arbeitens anhand des agilen Manifests dargestellt. Darüber hinaus bietet Teil I zwei Praxisbeispiele, welche zeigen, wie eine erfolgreiche Implementierung von agilen Methoden in der Praxis der öffentlichen Verwaltung aussehen kann. Die Praxisbeispiele wurden ausgewählt, da sie einige wesentliche Elemente der agilen PM²-Methodik verdeutlichen und dennoch eine Abwandlung und Anpassung ermöglichen, so wie es das PMflex-System für den individuellen Einsatz vorsieht.
- Teil II: PM²-Agile – Die Methode der Europäischen Kommission: Im zweiten Teil wird die PM²-Methodik für Agilität der Europäischen Kommission dargestellt.

Das CC GroßPM legt mit dem PMflex-Agil-Leitfaden einen Standard für agiles Vorgehen in der öffentlichen Verwaltung vor. Er erläutert die wichtigsten agilen Methoden und Techniken für die Projektarbeit und soll Sie dabei unterstützen, Agilität in Ihrer Organisation erfolgreich zu implementieren. Er ist ein Teil des ganzheitlichen Projektmanagementsystems PMflex, welches neben vier Leitfäden (Projekt-, Programm- und Portfoliomanagement sowie Agil) auch ergänzende Bausteine zur Verbreitung und

¹ vgl. Theobald, S./Guckenbiehl, P. (2021). Agilität in der Verwaltung – Agiles Arbeiten als Chance für die moderne Verwaltung. ([Link](#))

Wissensvermittlung (z. B. Netzwerke, PM-Wissenspool und Lehrprogramm) umfasst. Alle PMflex-Leitfäden und Artefakte sowie weitere Arbeitshilfen stellt das CC GroßPM auf seiner Webseite zum Download zur Verfügung.

INHALTSVERZEICHNIS

I.	Agilität in der öffentlichen Verwaltung.....	1
1.	PMflex-Agil als Teil des PMflex-Systems	1
2.	Herausforderungen	5
3.	Erfolgsfaktoren.....	7
4.	Praxisbeispiele	10
4.1	Agile Softwareentwicklung mit Scrum im PNR-Projekt im BVA.....	10
4.2	Implementierung von SAFe im Programm „Digitale Verwaltung Niedersachsen“	11
II.	PM ² -Agile - Die Methode der Europäischen Kommission	15
1.	Einführung	15
2.	Was ist Agile?	18
2.1	Agile Werte.....	19
2.2	Agile Teams.....	22
2.3	Kontinuierliche Verbesserung und validiertes Lernen.....	23
2.4	Mythen über Agilität	26
3.	Überblick über PM ² -Agile	28
3.1	Prinzipien von PM ² -Agile.....	28
3.2	Agilität und die PM ² -Methodik	29
3.3	Das PM ² -Agile-Modell	31
3.3.1	Lebenszyklus.....	33
3.3.2	Iterationen.....	37
3.3.3	Releases.....	40
3.4	Mindsets von PM ² -Agile	41
3.5	Artefakte und Dokumentation bei PM ² -Agile	43
3.6	Tailoring und Anpassen.....	45
4.	Themen von PM ² -Agile.....	47
4.1	Lean UX	47
4.2	Planung.....	49
4.3	Abstimmung und Berichterstattung	50
4.4	Anforderungen	52
4.5	Schätzung und Priorisierung.....	55
4.6	Risiken.....	56
4.7	Qualität.....	57
4.8	Weiterentwicklung und Veränderung	58
4.9	Architektur.....	60
4.10	Compliance und Sicherheit	61
4.11	Entwicklung.....	62
4.12	Software-Konfigurationsmanagement.....	64
4.13	Test.....	66
4.14	Deployment und Einführung	67
5.	Events	68
5.1	Iterationsplanung	69
5.1.1	Häufigkeit und Dauer.....	69
5.1.2	Struktur einer Iterationsplanung	69
5.1.3	Handlungsempfehlungen und Beteiligte.....	73
5.2	Daily Stand-up.....	74
5.2.1	Häufigkeit und Dauer.....	74
5.2.2	Struktur eines Daily Stand-up.....	75

5.2.3	Handlungsempfehlungen und Beteiligte.....	76
5.3	Release Planning.....	77
5.3.1	Häufigkeit und Dauer.....	78
5.3.2	Struktur des Release Planning.....	78
5.3.3	Handlungsempfehlungen und Beteiligte.....	81
5.4	Iterationsreview.....	85
5.4.1	Häufigkeit und Dauer.....	85
5.4.2	Struktur eines Iterationsreviews.....	86
5.4.3	Handlungsempfehlungen und Beteiligte.....	89
5.5	Iterationsretrospektive.....	91
5.5.1	Häufigkeit und Dauer.....	92
5.5.2	Struktur einer Iterationsretrospektive.....	93
5.5.3	Handlungsempfehlungen und Beteiligte.....	96
6.	Rollen und Zuständigkeiten.....	98
6.1	PM²-Agile-Projektorganisation.....	99
6.2	Stakeholderinnen und Stakeholder.....	99
6.3	Kernrollen.....	100
6.3.1	Teamkoordinatorin oder Teamkoordinator (TeCo).....	100
6.3.2	Product Owner (PrOw).....	100
6.3.3	Systemarchitektin oder Systemarchitekt (ArOw).....	101
6.3.4	Agiles Teammitglied (ATeM).....	102
6.4	Andere Rollen.....	102
6.5	Die Projektleitung (PM) und die Anforderungsmanagerin bzw. der Anforderungsmanager (BM) in einem PM²-Agile-Projekt.....	103
6.5.1	Projektleitung (PM).....	103
6.5.2	Anforderungsmanagerin bzw. Anforderungsmanager (BM).....	104
6.6	RAM (RASI) - Dokumentation von Verantwortlichkeiten.....	105
7.	Artefakte.....	107
7.1	IT-Governance-Artefakte.....	108
7.1.1	Business Case.....	108
7.1.2	Projektauftrag.....	109
7.1.3	Systemarchitekturübersicht.....	110
7.1.4	Geschäftsmodell.....	112
7.2	Agile-spezifische Artefakte.....	114
7.2.1	Entwicklungshandbuch.....	114
7.2.2	Entwicklungsarbeitsplan.....	117
7.2.3	Testplan.....	127
7.2.4	Bereitstellungsplan.....	129
7.3	Koordinierungs- und Berichtsartefakte.....	131
7.3.1	Entwicklungsstatusbericht.....	131
7.3.2	Projektprotokolle.....	134
7.3.3	Projektberichte.....	134
8.	Tools und Techniken.....	136
8.1	Bewertung der Teamarbeit.....	136
8.2	Selbst organisierte Teams.....	136
8.3	Design Blocks.....	136
8.4	Features und Storys.....	137
8.5	Zerlegen von User Storys.....	137
8.6	Definition of Done.....	138
8.7	Planning Poker.....	138
8.8	Kanban-Methode.....	138
8.9	Burndown-Charts.....	139
8.10	Testgetriebene Entwicklung.....	139

8.11	Pairing.....	140
	Anhang	141
1.	Tabellarische Übersichten und Diagramme der agilen Artefakte und Events	141
2.	Artefaktlandschaft der PM ² -Methodik und PM ² -Agile	142
3.	Überblick über die PM ² -Agile-Aktivitäten und -Artefakte	143
4.	Begriffsverzeichnis.....	144
5.	Abbildungsverzeichnis.....	146

I. AGILITÄT IN DER ÖFFENTLICHEN VERWALTUNG

1. PMflex-Agil als Teil des PMflex-Systems

Dieser Leitfaden ist ein Teil des ganzheitlichen Projektmanagementsystems PMflex, welches sich aus den in der nachfolgenden Abbildung dargestellten vier Leitfäden, dem sogenannten PMflex-Standard, sowie weiteren ergänzenden Bausteinen zur Verbreitung und Wissensvermittlung (z. B. Netzwerke, PM-Wissenspool und Lehrprogramm) zusammensetzt.

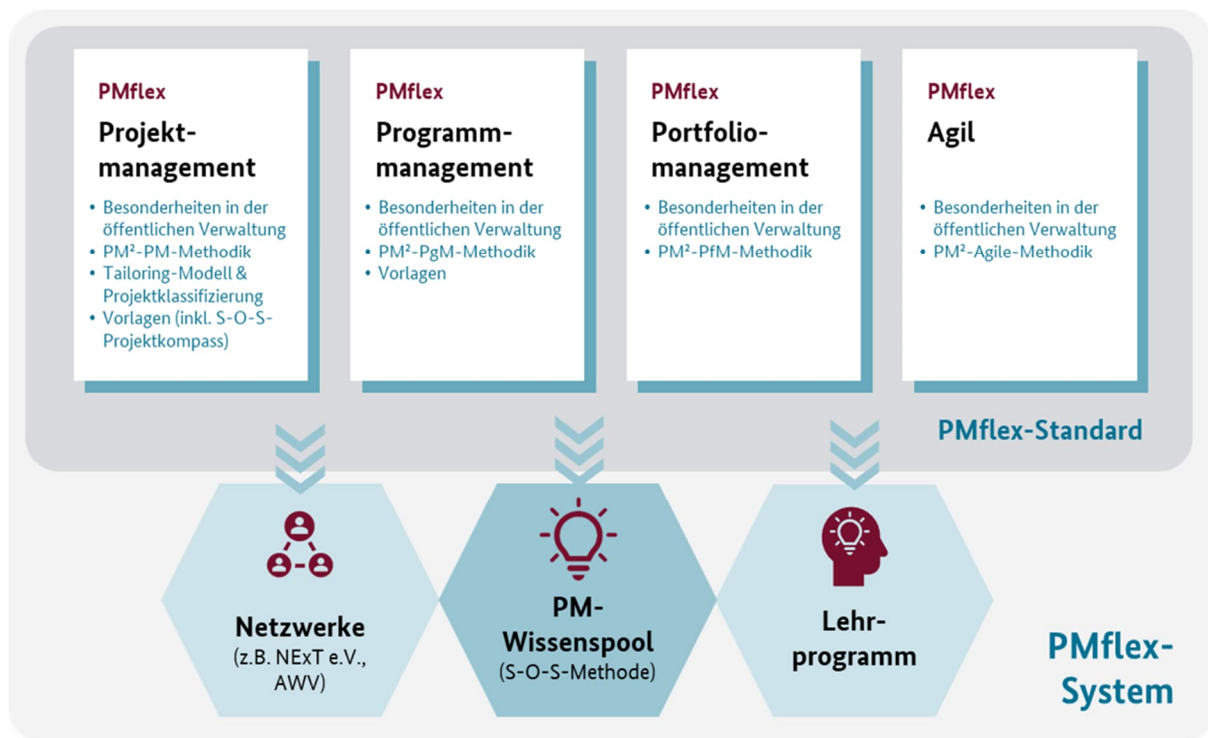


Abbildung 1: PMflex – Das ganzheitliche Projektmanagementsystem

Der Leitfaden PMflex-Projektmanagement liefert grundlegende Projektmanagementkonzepte und -methoden, die Voraussetzung für effektives Projektmanagement sind. Dies umfasst eine Projektgovernancestruktur, Prozessrichtlinien, Vorlagen (Artefakte), Richtlinien für die Benutzung dieser Vorlagen und eine Reihe von Mindsets. Der Leitfaden PMflex-Projektmanagement bildet das Fundament für das gesamte PMflex-Projektmanagementsystem auf dem die weiteren Elemente aufbauen.

Der Leitfaden PMflex-Programmmanagement ergänzt PMflex-Projektmanagement um Handlungsempfehlungen und Tools für das Programmmanagement, die es Programmleitungen ermöglicht, Programme besser zu steuern und zu überwachen. Außerdem enthält er Techniken für eine wirksame Berichterstattung über den Programmnutzen und dessen Nachverfolgung sowie für eine angemessene Kommunikation an die oberste Führungsebene und maßgebliche Stakeholderinnen und Stakeholder.

Der Leitfaden PMflex-Portfoliomanagement enthält Methoden und Techniken zur übergreifenden Strukturierung und Steuerung von Projekt- und Programmportfolios, um den Beitrag zur Strategie der Organisation zu maximieren. PMflex-Projektmanagement wird somit um eine strategische und über-

geordnete Komponente zur Bewältigung von Herausforderungen bei der Ausrichtung und Abstimmung komplexer Portfolios ergänzt. Im deutschsprachigen Raum ist ebenfalls der Begriff Multiprojektmanagement (MPM) verbreitet.²

Der Leitfaden PMflex-Agil baut auf PMflex-Projektmanagement auf und ergänzt den Leitfaden um agile Methoden für die Projektarbeit. Agile Methoden gewinnen zunehmend an Bedeutung und sind aus einem ganzheitlich angelegten Projektmanagementsystem nicht mehr wegzudenken. Gerade in komplexen Projekten bedarf es eines zusätzlichen Repertoires an agilen Methoden, um deren Komplexität zu bewältigen.

Innerhalb des PMflex-Systems werden die Leitfäden ergänzt durch:

- Netzwerkangebote zum fachlichen Austausch zwischen Projektmitarbeiterinnen und -mitarbeitern, Projektleitungen (NExT) sowie Führungsebenen (AWV),
- einen zusätzlichen PM-Wissenspool, welcher vertiefende Informationen zu verschiedenen Projektmanagementthemen bietet, sowie
- ein Lehrprogramm, welches sich maßgeschneidert an Projektmitarbeiterinnen und -mitarbeiter, Projektleitungen sowie die Führungsebene richtet.

PMflex übernimmt die PM²-Projektmanagementmethodik der Europäischen Kommission. PM² wurde von der Europäischen Kommission frei zugänglich gemacht, sodass die Methodik und das gesamte PM²-Angebot und die damit verbundenen Vorteile einem breiteren Kreis von Stakeholderinnen und Stakeholdern sowie Nutzenden zur Verfügung stehen. Die erhöhte Wirksamkeit des Managements und der Kommunikation von Projektarbeit leistet einen Beitrag zu den Zielsetzungen der Europäischen Union und greift die Anforderungen der Mitgliedstaaten und der Bürgerinnen und Bürger der EU auf.

Das Ziel von PM² ist es, in konvergierende Ansätze zu investieren, auf der Grundlage von Ähnlichkeiten und gemeinsamen Interessen der breiteren öffentlichen Verwaltung in der EU und darüber hinaus.

² Wenn im weiteren Verlauf der Begriff Multiprojektmanagement verwendet wird, ist dieser mit Portfoliomanagement gleichzusetzen. Die Vision von PMflex ist, eine einheitliche Projektmanagementsprache in der öffentlichen Verwaltung zu schaffen.

„Eine gemeinsame PM-Methodik, die für alle EU-Institutionen, Mitgliedstaaten, Dienstleistende und Bürgerinnen und Bürger der EU frei zugänglich ist.“

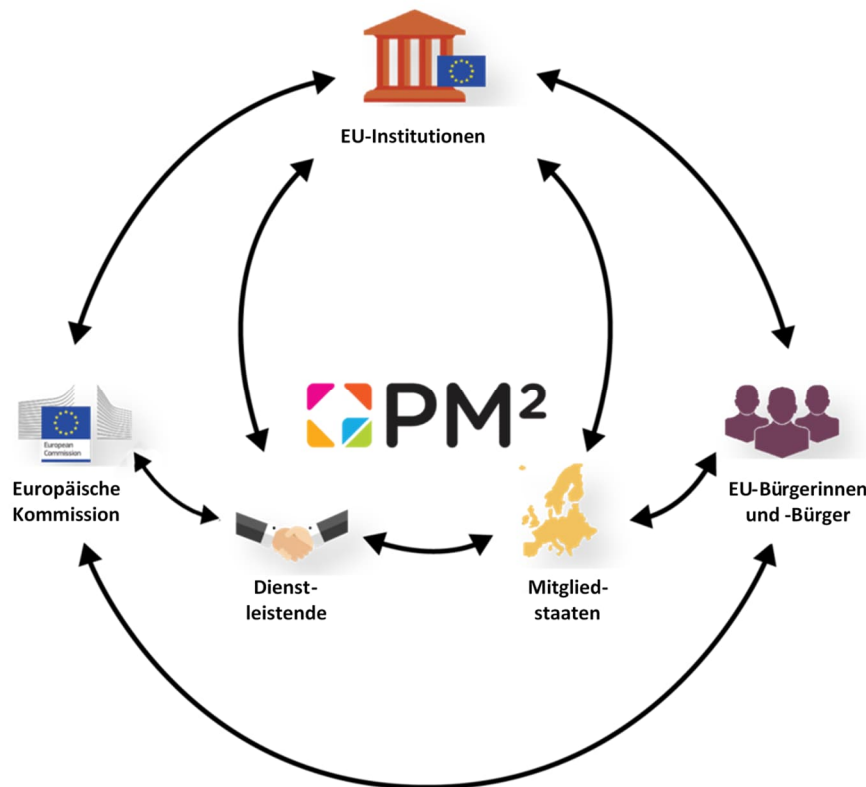


Abbildung 2: Vision und Synergien von PM²

Das CC GroßPM hat sich hierfür aus den nachfolgenden Gründen entschieden:

- Es handelt sich um eine Methode, in die eine Vielzahl weltweit anerkannter Projektmanagementpraktiken eingeflossen sind. Ihre Entwicklung wurde sowohl von praktischen Erfahrungen in verschiedenen Projekten innerhalb von europäischen Institutionen als auch in externen Einrichtungen beeinflusst.
- PM² bietet ein vollständiges Projektmanagementökosystem an, welches auch Programm- und Portfoliomanagement sowie agile Methoden beinhaltet. Zudem beinhaltet es alle Elemente, die Voraussetzung für die erfolgreiche Durchführung von Projekten sind, wie eine Projektsteuerungsstruktur, Prozessrichtlinien, Vorlagen (Artefakte³), Hinweise zu deren Verwendung und einer Reihe effektiver Denkweisen.
- Die Methode ist speziell auf die Bedürfnisse der öffentlichen Verwaltung zugeschnitten und hat sich bereits in einer Vielzahl von Projekten der Europäischen Kommission und anderer europäischer Organisationen bewährt. PM² wurde zwar im Hinblick auf die Anforderungen von EU-Institutionen und deren Projekten entwickelt, ist jedoch auf Projekte in jeder Organisation übertragbar.
- Es handelt sich um eine leichte, einfach zu implementierende und vor allem anpassbare Methodik, die durch die Europäische Kommission frei zur Verfügung gestellt wird (Open Access). Das ermöglicht eine uneingeschränkte Anpassung und Anwendung der Methode.

³ Der Begriff „Projektartefakte“ wurde aus der PM²-Methode übernommen. Es handelt sich hierbei um Dokumente, die im Rahmen des Projektmanagements zu erstellen sind.

- Die PM²-Methode wird durch das Centre of Excellence in Project Management (CoEPM²) und eine aktive Community-of-Practice kontinuierlich weiterentwickelt und verbreitet. Durch die Übernahme des Standards können in signifikantem Maße Aufwände für die Entwicklung und/oder Anpassung eigener Standards eingespart werden und es ergeben sich vielfältige Synergien und Austauschmöglichkeiten nicht nur innerhalb der deutschen öffentlichen Verwaltung, sondern auch innerhalb Europas.
- Die im BVA entwickelte S-O-S-Methode[©] wurde speziell für Großprojekte entwickelt und setzte auf bereits vorhandenen Projektmanagementstandards auf. Mit der Übernahme der PM²-Methodik wird zukünftig ein ganzheitlicher Projektmanagementstandard für alle Projektgrößen gewährleistet.

Das CC GroßPM hat im Rahmen der Erstellung von PMflex-Agil die PM²-Methodik für die öffentliche Verwaltung in Deutschland aus dem englischen PM²-Original übersetzt und in diesen Leitfaden übertragen. Darüber hinaus wird PMflex-Agil durch die folgenden Elemente ergänzt:

- Hinweise zu den Besonderheiten und Herausforderungen von Agilität in der öffentlichen Verwaltung
- Hinweise zu Erfolgsfaktoren von Agilität
- Praxisbeispiele, die den Einsatz agiler Methoden in der öffentlichen Verwaltung zeigen

Alle PMflex-Leitfäden und Projektartefakte sowie weitere Arbeitshilfen stellt das CC GroßPM auf seiner Webseite zum Download zur Verfügung.

2. Herausforderungen

Die im PMflex-Projektmanagement-Leitfaden dargestellten und in der nachfolgenden Abbildung zusammengefassten Herausforderungen von Projektmanagement in der öffentlichen Verwaltung gelten grundsätzlich auch für Agilität, welche den Einsatz agiler Methoden in verschiedenen Bereichen der Organisation fokussiert. Im Hinblick auf Agilität sind insbesondere die Einflüsse von Verwaltung, Anspruchsgruppen und Politik hervorzuheben, die in diesem Abschnitt erläutert werden.



Abbildung 3: Herausforderungen von Agilität in der öffentlichen Verwaltung

Verwaltungsstrukturen und -richtlinien

Die öffentliche Verwaltung ist gegenüber privatwirtschaftlichen Unternehmen häufig durch hierarchische und bürokratischere Strukturen gekennzeichnet. Entscheidungen werden üblicherweise durch die Vorgesetzten getroffen, Abstimmungen müssen unter Beachtung des Dienstweges erfolgen. Diese Strukturen können aber im Widerspruch zu agilen Methoden stehen, die eine flexible Organisation erfordern. Dies kann eine Herausforderung in der öffentlichen Verwaltung darstellen. Daher liegt besondere Priorität darauf, die Beschäftigten der öffentlichen Verwaltung auf einen kulturellen und organisatorischen Wandel vorzubereiten und sie weitestgehend zu unterstützen.

Agile Methoden zeichnen sich zudem durch die Zusammenarbeit von Fachleuten aus unterschiedlichen Bereichen aus (sogenannte cross-funktionale Teams). In den Projektteams werden Spezialisten zusammengezogen, die gemeinsam an Problemen arbeiten. Mit einer solchen zuständigkeits- und abteilungsübergreifenden Zusammenarbeit sind Einrichtungen der öffentlichen Hand nicht immer vertraut. Üblicherweise werden Entscheidungen innerhalb der Organisationsbereiche, z. B. Abteilungen, relativ autonom getroffen. Insofern bedarf es auch hier eines kulturellen und organisatorischen Wandels, der die Zusammenarbeit in solchen Teams unterstützt. Von entscheidender Bedeutung ist hierbei eine Unterstützung der Leitungsebene, um die Teams in die Lage zu versetzen, notwendige Entscheidungen treffen zu können.

Einflussnahme durch Politik

Eine weitere Herausforderung für Agilität liegt darin, dass agile Methoden mit dem Anspruch der Politik und der medialen Öffentlichkeit kollidieren, indem gefordert wird, dass gelieferte Ergebnisse unmittelbar alle Anforderungen erfüllen. Projekte in der öffentlichen Verwaltung zeichnen sich durch eine hohe Komplexität und viele Stakeholderinnen und Stakeholder aus, die an der Erstellung der Projektergebnisse beteiligt und deren unterschiedliche Interessen und Bedürfnisse zu berücksichtigen sind. Die sofortige Berücksichtigung aller Interessen ist mit einer agilen Vorgehensweise teilweise

schwer vereinbar, weil diese durch eine inkrementelle Vorgehensweise geprägt ist und somit Anforderungen erst nach und nach umgesetzt werden können.

3. Erfolgsfaktoren

Agilität ist mehr als ein Projektmanagementansatz. Es ist vielmehr eine Haltung. Agilität erfordert daher von den Projektverantwortlichen und allen Projektbeteiligten ein spezielles Mindset sowie die Verinnerlichung bestimmter Werte, die als übergreifender Erfolgsfaktor für die Einführung von Agilität angesehen werden kann. Das agile Manifest⁴ formuliert vier Werte und 12 handlungsleitende Prinzipien agilen Vorgehens, die in der nachfolgenden Abbildung dargestellt sind. Die vier Werte lauten:

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

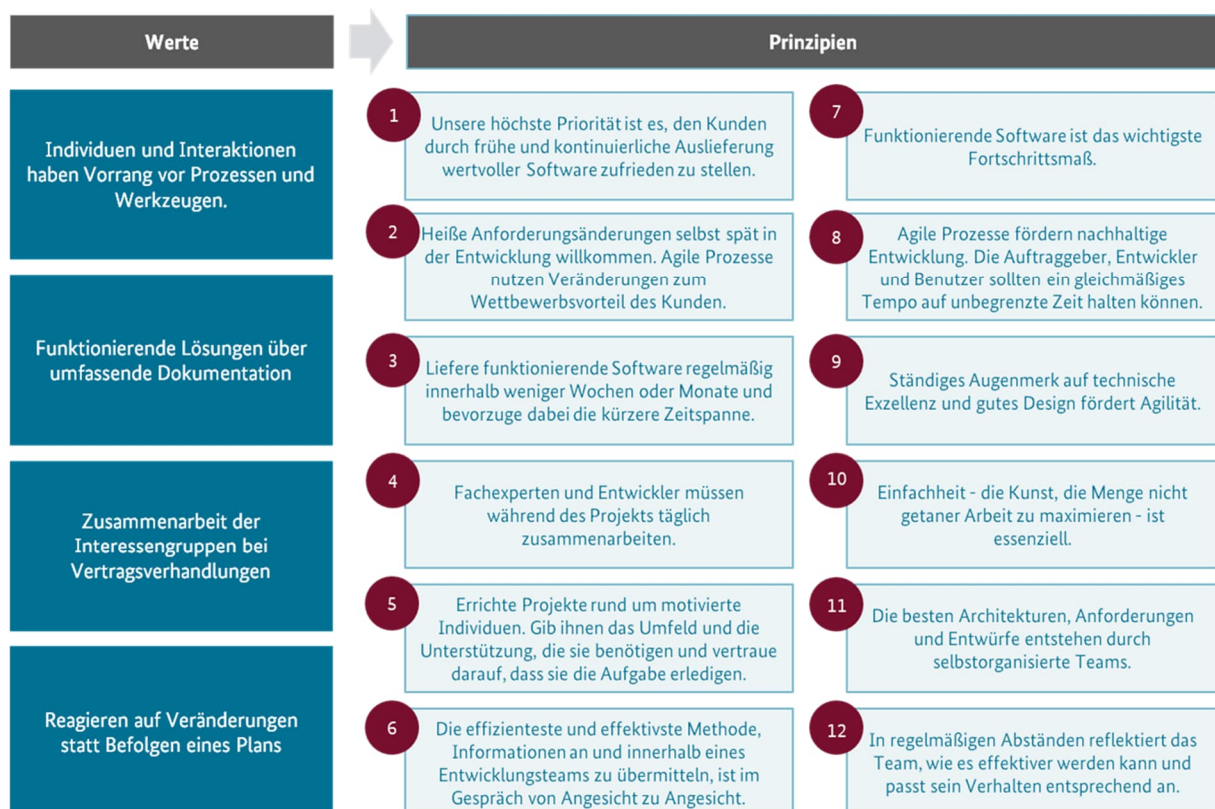


Abbildung 4: Werte und Prinzipien des agilen Manifests

Die Übernahme der oben genannten Werte und die Berücksichtigung der 12 Prinzipien werden in diesem Leitfaden als Erfolgsfaktoren für agiles Arbeiten in der öffentlichen Verwaltung herangezogen. Im Folgenden werden diese Prinzipien näher erläutert.

1. **Ausgeprägte Kundenzufriedenheit:** Agile Projekte sollen so durchgeführt werden, dass bereits sehr früh im Projekt und von da an in regelmäßigen Abständen Ergebnisse geliefert werden, um die Kundinnen und Kunden zufriedenzustellen. Dies kann in Form einzelner Teilprodukte erfolgen oder durch die Entwicklung von Prototypen oder Versionen, die kontinuierlich unter Berücksichtigung der Kundinnen- und Kundenwünsche verbessert werden, bis das Gesamtziel des Projekts erreicht ist.

⁴ Vgl. Agile Manifesto (2023). Manifest für Agile Softwareentwicklung ([Link](#))

2. Positive Einstellung zu Anforderungsänderungen: Agile Projekte stellen die Anforderungen der Kundinnen und Kunden in den Mittelpunkt. Daher sind Veränderungen der Anforderungen im Projektverlauf als positiv und Chance zur Verbesserung des Projektergebnisses anzusehen. Die Berücksichtigung von Anforderungsänderungen sollte daher als wesentliches Element agilen Vorgehens betrachtet werden.
3. Konzentration auf Wertschöpfung in kurzer Zeit: In agilen Projekten ist darauf zu achten, dass Ergebnisse in kurzen Zyklen geliefert werden und dass bei der Erstellung von Ergebnissen besondere Aufmerksamkeit auf dessen Nutzen und ihre Anwendbarkeit zu legen ist. Kurze Zyklen reduzieren in agilen Projektumgebungen die Komplexität der Aufgabenstellung und der Planung, um schnell Ergebnisse zu liefern.
4. Enge Zusammenarbeit der Projektbeteiligten: Es ist in agilen Projekten auf eine enge Zusammenarbeit der Projektbeteiligten zu achten. Hierbei sollten auch die Stakeholderinnen und Stakeholder aktiv in das Projekt einbezogen werden. Fachexpertinnen und Fachexperten sowie Entwicklerinnen und Entwickler müssen eng zusammenarbeiten, um Lösungen zu generieren, welche die Kundenbedürfnisse optimal erfüllen. Nur in der Zusammenarbeit, die alle Perspektiven berücksichtigt, wird ein optimales Ergebnis erzielt.
5. Unterstützendes Projektumfeld: Agile Projektteams müssen aus motivierten Beschäftigten bestehen, die selbstorganisiert an der Lösung von Problemen arbeiten. Hierfür muss ihnen der nötige Freiraum gegeben werden. Mikromanagement ist zu vermeiden, damit das Projektteam sein ganzes Potential entfalten kann. Ferner müssen Führungskräfte für ein Umfeld sorgen, in dem das Projektteam von äußeren Störungen geschützt arbeiten kann.
6. Informationsübermittlung im persönlichen Gespräch: Für agile Projekte ist eine transparente und vor allem direkte Kommunikation zwischen den Teammitgliedern, Entwicklerinnen und Entwicklern sowie Stakeholderinnen und Stakeholdern wichtig. Die für die Erfüllung der Aufgaben notwendigen Informationen sollen zwischen den Personen, die sie besitzen und denen, die sie brauchen, persönlich ausgetauscht werden. Hierdurch wird eine optimale Abstimmung der Aktivitäten der Projektteammitglieder erreicht.
7. Funktionsfähigkeit: Der Fortschritt agiler Projekte ist an den erstellten Ergebnissen zu messen. Häufig werden zur Bemessung des Projektfortschritts Kennzahlen erhoben, wie z. B. geleistete Arbeitsstunden oder Budgetausschöpfung, die aber nur geringe Aussagekraft in Hinblick auf die Zielerreichung haben.
8. Kontinuierliche Projektarbeit: Agile Projektarbeit erfordert eine Haltung, bei der das Projektteam das Ziel verfolgt, kontinuierlich in jeder Iteration ein Ergebnis zu liefern. Das erfordert einige Übung, insbesondere im Hinblick auf die Erreichbarkeit solcher Ziele. Es ist darauf zu achten, dass nach dem Sprint vor dem Sprint ist und die Projektteammitglieder in der Lage sein müssen, die Aufgaben über einen längeren Zeitraum hinweg zu bewältigen.
9. Exzellenz und gutes Design: Eine Voraussetzung für Agilität ist, dass Projektteams in jeder Hinsicht den Anspruch haben, Ergebnisse höchster Qualität zu liefern. Dies ist die Voraussetzung für Kundenzufriedenheit und vermeidet aufwändige Nacharbeiten. Hohe Qualität zeigt

sich dabei regelmäßig in gutem Design, welches dadurch gekennzeichnet ist, dass es die Kundenbedürfnisse optimal erfüllt. Dies kann eine exzellente Usability einer Fachanwendung ebenso sein, wie ein exzellent geschriebenes Gesetz.

10. Einfachheit: In agilen Projekten sollte die Aufmerksamkeit darauf gelenkt werden, Prozesse möglichst einfach zu gestalten. Unproduktive Tätigkeiten sollten vermieden und alle Kapazitäten in die Durchführung wertschöpfender Aktivitäten gelenkt werden.
11. Hohes Maß an Selbstorganisation: Agilität in Projekten kann nur durch motivierte und kompetente Projektteams erreicht werden, die unter einem hohen Maß an Selbstorganisation eigenverantwortlich, funktionsübergreifend, kooperativ arbeiten und Entscheidungen treffen können.
12. Regelmäßige Selbstreflektion: Ein Kerngedanke agilen Vorgehens ist es, sich um eine kontinuierliche Verbesserung zu bemühen, um das Projektergebnis und den Entwicklungsprozess zu optimieren. In agilen Projekten ist es wichtig, dass das Projektteam bereit ist, aus Fehlern zu lernen und sich selbst bzw. bestehende Prozesse ständig zu verbessern. Durch regelmäßige Retrospektiven können Erkenntnisse gewonnen werden, die anschließend im nächsten Zyklus direkt berücksichtigt und eingeplant werden.

4. Praxisbeispiele

4.1 Agile Softwareentwicklung mit Scrum im PNR-Projekt im BVA

Das Bundesverwaltungsamt (BVA) setzt bereits seit 2017 in einzelnen Projekten agile Methoden erfolgreich um. Mittels eines agilen Mindsets und entsprechender Methoden will das BVA schnell und flexibel auf Herausforderungen reagieren und erfolgreich IT-Projekte meistern. Ein Beispiel hierfür bietet die agile Softwareentwicklung im Bereich des Passenger Name Records (PNR).

Um in der Projektarbeit auf neue Erkenntnisse und Anforderungen schnell und flexibel reagieren zu können, wurde bei der Entwicklung des PNR-Verfahrens ein agiler Projektansatz verfolgt, welcher auf Scrum basiert. Scrum ist eine agile Methodik, welche auf drei Säulen beruht: Transparenz, Überprüfung und Anpassung. Er wird insbesondere in der Softwareentwicklung verstärkt eingesetzt. Das Besondere an dieser Vorgehensweise ist die Flexibilität und Geschwindigkeit in der Entwicklungsteams auf neue Erkenntnisse und Kundenwünsche reagieren können. Fortschritt und Hindernisse bei der Softwareentwicklung werden regelmäßig und sichtbar festgehalten. Ergebnisse und Funktionalitäten werden regelmäßig ausgeliefert und durch die Kundinnen und Kunden bewertet.

Agiler Iterationszyklus

Die nachfolgende Abbildung zeigt den idealtypischen Verlauf eines agilen Iterationszyklus. Im Rahmen des Projekts wurden die Anforderungen des BKA, der BPOL, des BVA und weiteren beteiligten Stakeholderinnen und Stakeholdern in das Product Backlog des teilprojektübergreifenden Anforderungsmanagements aufgenommen. Nach Abstimmung und Klärung der Anforderungen wurden die Inhalte der Anforderungen in kleinere Einheiten (Epics und User Stories) unterteilt und in das Product Backlog der Teilprojekte aufgenommen.

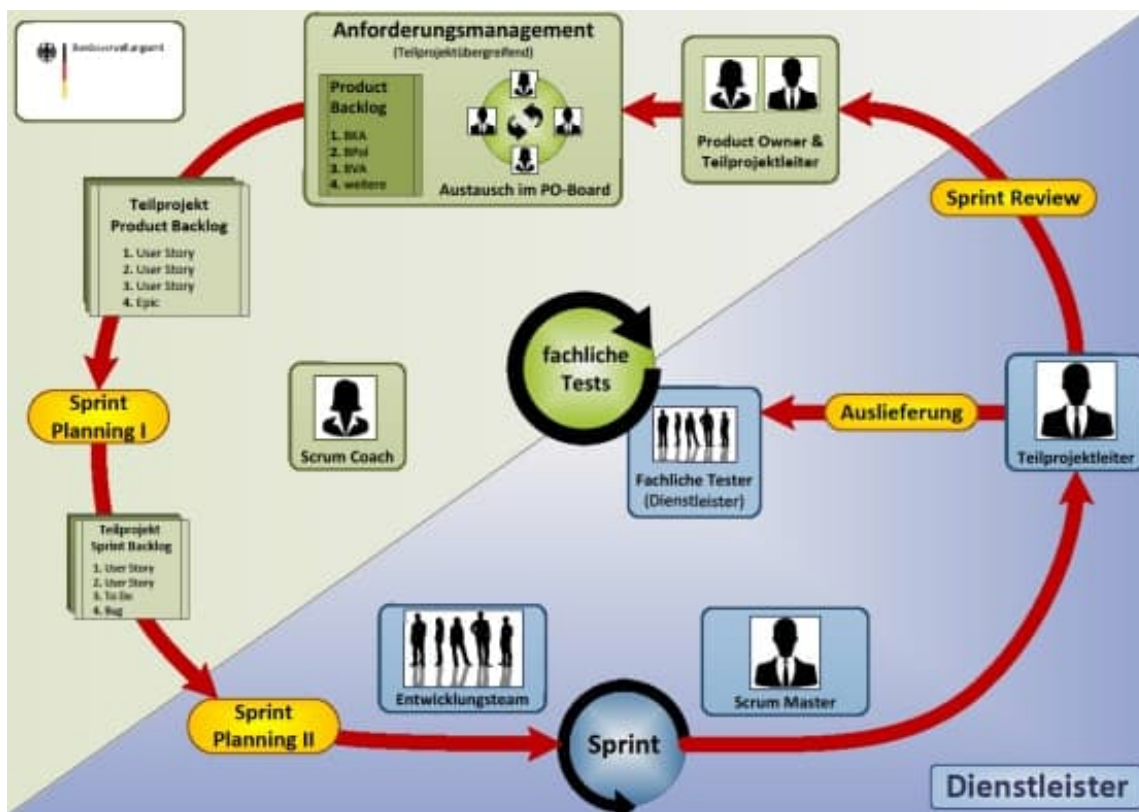


Abbildung 5: Agiler Iterationszyklus im PNR-Projekt im BVA

Anschließend wurden im Sprint Planning I (einer Form der Iterationsplanung) die am höchsten priorisierten User Stories in das Sprint Backlog des nächsten Sprints überführt, spezifiziert und mit dem Product Owner abgestimmt. Das darauffolgende Sprint Planning II fand innerhalb der Entwicklungsteams der Dienstleister statt, die die User Stories in vierwöchigen Sprintzyklen umsetzten. Anschließend wurden die User Stories als Funktionalitäten über die Teilprojektleitungen des Dienstleisters an die PNR-Teilprojekte ausgeliefert. Im nächsten Schritt wurden durch die Testteams die fachlichen Sprint-Tests gemeinsam mit dem Fachbereich durchgeführt.

Im Sprint Review, bei der es sich um eine Form der Iterationsretrospektive handelt, stellte der Dienstleister den Product Ownern, Teilprojektleitungen und weiteren Interessierten (z. B. der Fachlichkeit des BKA) die Ergebnisse vor und diese wurden gemeinsam bewertet. Wurde eine User Story nicht korrekt umgesetzt oder wurden Änderungen gewünscht, konnten diese dann bei der nächsten Sprintplanung berücksichtigt und neu umgesetzt werden.

Projektgovernance & Rollen

Das Projekt wies eine Projektgovernance auf, wie sie für agile Projekte typisch ist und die auch in der agilen PM²-Methodik berücksichtigt wird. Der Product Owner vertritt die Kundenseite im Projekt und ist für die Pflege des Product Backlogs zuständig. Er priorisiert die Elemente des Product Backlogs und plant Releases. Im PNR-Projekt gab es je Teilprojekt einen hauptamtlichen Product Owner aus dem Fachbereich, der im Tandem mit der jeweiligen technischen Teilprojektleitung des BVA die User Stories priorisiert und Sprints plant. Die Definition und Planung der Releases erfolgte durch den fachlichen Release Manager.

Der Scrum Master war für den Scrum-Prozess verantwortlich. Er sorgte dafür, dass der Prozess gelebt wird, unterstützte das Team, moderierte und räumte Hindernisse aus dem Weg. Diese Rolle wurde im PNR-Projekt nur bei den Dienstleistern aktiv ausgeübt. Auf BVA-Seite gab es die Rolle des Scrum Coaches oder des Agile Masters. Dieser stellte sicher, dass die agilen Prozesse eingehalten werden. Diese Rolle entspricht nicht der Teamleitung, da agile Teams selbstorganisiert arbeiten. Vielmehr kommt ihr eine moderierende, coachende Rolle zu.

Das Projektteam war im PNR-Projekt eine gemischte, interdisziplinäre Gruppe aus Entwicklerinnen und Entwicklern, Testerinnen und Testern sowie Redakteurinnen und Redakteuren für die Dokumentation. Im PNR-Projekt waren die Entwicklungsteams bei den Dienstleistern angesiedelt.

4.2 Implementierung von SAFe im Programm „Digitale Verwaltung Niedersachsen“

Die niedersächsische Landesregierung stand vor der Herausforderung, mehrere große Projekte im Rahmen des Programms Digitale Verwaltung Niedersachsen (DVN) umzusetzen. Bürgerinnen und Bürger sowie Unternehmen sollen künftig Dienstleistungen online in Anspruch nehmen können. Dazu gehört die Anmeldung eines Gewerbes ebenso wie die Beantragung von Wohngeld oder eines Schwerbehindertenausweises. Wie im Artikel „Agilität in der Praxis“ des Behördenspiegels vom Februar 2022 nachzulesen ist, waren in diesem Programm mehr als zehn komplexe Projekte mit über 100 Mitarbeiterinnen und Mitarbeitern zu managen, die mit einer Vielzahl von Beteiligten und sich ständig ändernden Anforderungen konfrontiert waren. Die niedersächsische Landesregierung entschied sich, dieser komplexen Ausgangslage durch den Einsatz agiler Methoden zu begegnen.

Zunächst musste eine geeignete Methodik gefunden werden, die eine agile Transformation ermöglicht. Das DVN-Programm entschied sich für das Scaled Agile Framework (SAFe), weil es klare Rollen und Verantwortlichkeiten definiert, Skalierungsmöglichkeiten für große Programme bietet und eine öffentlich zugängliche Dokumentation zur Verfügung stellt.

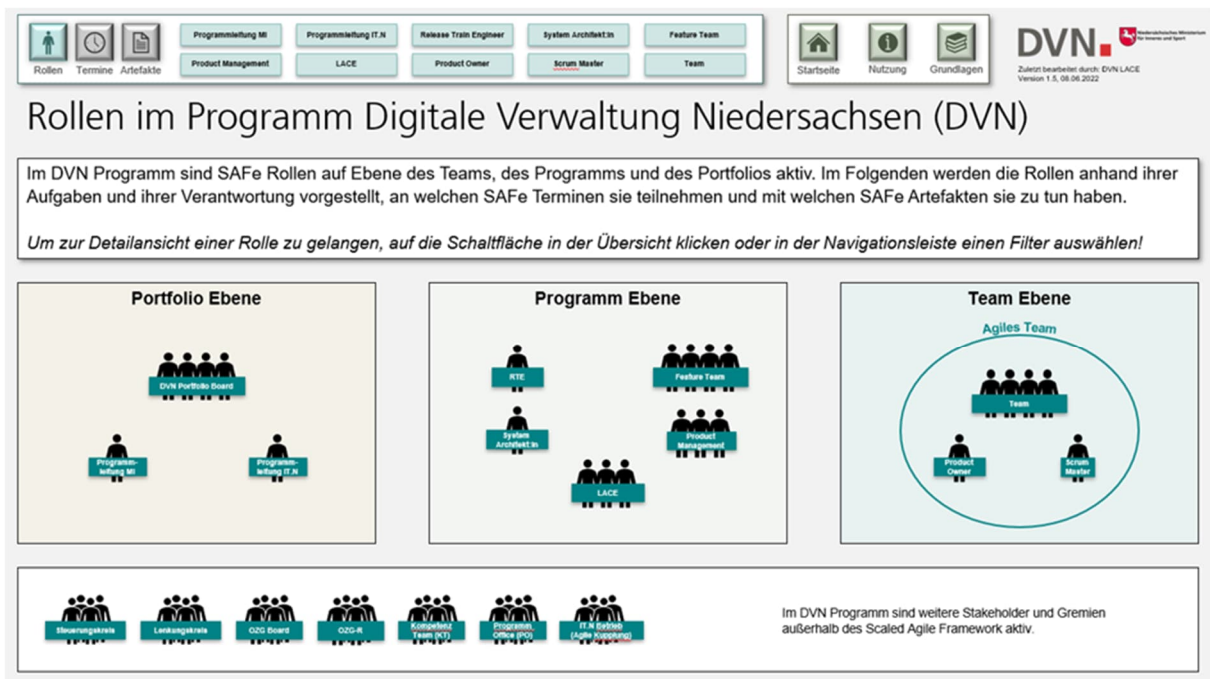


Abbildung 6: Rollen des Programms Digitale Verwaltung Niedersachsen

SAFe ist das international am weitesten verbreitete Framework für skalierte Agilität in der Wirtschaft und im öffentlichen Sektor. Rollen und Verantwortlichkeiten können auf das Management übertragen werden, ohne dass etablierte Strukturen aufgegeben werden müssen. Mit SAFe werden Werte und Prinzipien auf Organisationsebene implementiert und Kulturveränderungen durch geeignete Prozesse, Methoden und Artefakte unterstützt.

Etablieren eines Lean Agile Center of Excellence (LACE)

Ein zentrales und erfolgskritisches Element bei der Einführung von SAFe ist die Etablierung eines Lean Agile Center of Excellence (LACE). Hier wurden frühzeitig die relevanten Entscheidungsträgerinnen und Entscheidungsträger eingebunden, deren Zustimmung für die Umsetzung von SAFe notwendig war.

Das LACE wurde aus einem kleinen agilen Kompetenz-Team gebildet, das aus vier Personen mit methodischer Expertise und mit Verwaltungserfahrung besteht. Dieses Expertenteam verfügt über die notwendige langjährige Erfahrung in großen agilen Transformationen und hat die notwendige Entscheidungs- und Umsetzungscompetenz, ohne die es zu Verzögerungen oder gar zum Abbruch des Transformationsprozesses kommt. Eine agile Transformation erfordert über einen längeren Zeitraum eine hohe Disziplin aller beteiligten Akteure und greift in die Organisation, in bestehende Prozesse und in die Kultur ein. Zentrale Aufgaben des LACE waren daher die Erstellung eines Transformationsplans, die schrittweise Einführung agiler Regelwerke und Termine sowie die Implementierung und das Coaching der neuen agilen Rollen. Bei der Einführung des agilen Anforderungsmanagements wurden die SAFe-typischen agilen Planungsartefakte eingeführt.⁵

⁵ Vgl. Henke, A. (2022). Agilität in der Praxis Teil 2, Behördenspiegel. S. 42.

PI-Planning

Ein wesentliches Schlüsselement und einer der ersten großen sichtbaren Umsetzungsschritte des DVN-Programms unter Federführung des LACE war die Implementierung der Programm-Inkremental-Planung (PI-Planung). Die PI-Planung ist das Herzstück der SAFe-Methodik. Sie findet in regelmäßigen Abständen am Ende eines Inkrements oder Entwicklungszyklus statt. Im DVN-Programm findet die PI-Planung alle drei Monate statt. In diesem Meeting werden die Ergebnisse/Aufgaben der nächsten drei Monate in einem bewährten agilen Prozess geplant und abgestimmt. Dabei werden ausnahmslos alle Programmbeteiligten einbezogen: von der strategischen Ebene (CIO und Geschäftsführung des IT-Dienstleisters des Landes) über die Ansprechpartner der Ressorts bis hin zur operativen Ebene in den einzelnen Projektteams des Programms.

Vor allem im skalierten agilen Vorgehen ist es essenziell, dass alle Teams untereinander abgestimmte Ziele und Liefergegenstände haben, um sicherzustellen, dass übergreifend das Programm im Zusammenspiel produktive Ergebnisse liefern kann. „In der Regel findet dieses Event physisch mit erfahrungsgemäß bis zu 150 Teilnehmerinnen und Teilnehmern statt.“

Zusätzlich zur PI-Planung finden in kürzeren Abständen standardisierte und kürzere Termine zur Abstimmung fachlicher und methodischer Themen auf allen Planungsebenen statt. Dazu gehören Regeltermine der Product Owner und Scrum Master sowie System Demos. Durch diese Termine wird sichergestellt, dass die Planungen des PI-Plannings auch umgesetzt werden, auf Änderungen reagiert werden kann und eine größtmögliche Transparenz während der Umsetzung erreicht wird.⁶

Inspect & Adapt

Am Ende eines vierteljährlichen Inkrements müssen produktive Ergebnisse fertiggestellt und der Mehrwert sichtbar gemacht werden. Hierzu dient der gesonderte Regeltermin „Inspect & Adapt“-Workshop (Workshop zur Inspektion und Anpassung).⁷ In diesem Workshop werden die produktiven Ergebnisse des agilen Vorhabens allen relevanten Beteiligten vorgestellt. Die Planungsgenauigkeit und der entstandene Mehrwert wird ebenfalls festgestellt. So werden mögliche Verbesserungen am Vorgehen identifiziert. Ein „Inspect & Adapt“-Workshop besteht aus drei Teilen:

PI-System-Demo

Während einer PI-System-Demo stellen die agilen Teams ihre produktiven Ergebnisse allen relevanten Beteiligten vor. Dieser Termin ermöglicht eine maximale, ebenenübergreifende Transparenz bezüglich der erzielten Ergebnisse und bezüglich des Programmfortschrittes. In diesem Teil des Termins können auch Fragen gestellt und Feedback geben werden.

Quantitatives and qualitatives Measurement

Unmittelbar im Anschluss an die PI-System-Demo erfolgte ein quantitatives und qualitatives Measurement in einem kleineren Kreis. In diesem werden der Mehrwert der gelieferten Ergebnisse am Ende des Inkrements bewertet und mit den Schätzungen, die im Rahmen des letzten PI-Plannings erfolgten, verglichen. Im Laufe mehrerer Inkremente sollte sich hier ein positiver Trend zeigen, der darauf hindeutet, dass sich die Schätz- und Planungsgenauigkeit der Teams kontinuierlich verbessert. Dieser Teil des Workshops gibt Aufschluss darüber, ob die Planung tatsächlich umgesetzt und auch der geforderte Mehrwert geliefert werden kann.

⁶ Vgl. Henke, A. (2022). Agilität in der Praxis Teil 2, Behördenspiegel, 2022 (2). S. 31.

⁷ Vgl. Henke, A. (2022). Agilität in der Praxis Teil 2, Behördenspiegel, 2022 (2). S. 35.

Problem-Solving-Workshops

Im Anschluss werden Problemlösungsworkshops durchgeführt. In diesem Teil werden Hindernisse analysiert und diskutiert, die im Rahmen früherer Inkremente aufgetreten sind. Dazu werden meist kleinere Gruppen gebildet, die z.B. klassische „Fishbone-Analysen oder als Root-Cause-Analyse durchgeführt, um die Ursachen hinter den jeweiligen Problemen zu identifizieren und Lösungsvorschläge zu erarbeiten.

Die „Inspect & Adapt“-Workshops ist eine mögliche Umsetzung einer Retrospektiven zu einem Programminkrement. Retrospektiven kommt in agilen Methoden eine besondere Bedeutung zu. Sie sind Teil eines kontinuierlichen Lernprozesses, in dem Teams regelmäßig das zurückliegende Inkrement oder Iteration auf mögliche Verbesserungspotenziale analysieren, um ihr Verhalten anzupassen und immer besser zu werden.

Handlungsempfehlungen zur Implementierung agiler Methoden

Zusammenfassend lassen sich aus den Praxiserfahrungen des DVN-Programms folgende Empfehlungen für die Implementierung agiler Methoden in der Projektarbeit ableiten⁸:

- Beziehen Sie Finanzverantwortliche frühzeitig mit ein und Etablierung iterative Lernzyklen in der Finanzplanung.
- Bleiben Sie flexibel und übertragen Sie das Framework in ihren Verwaltungskontext, ohne die agilen Werte und Prinzipien zu entwerfen.
- Nutzen Sie die neuesten Technologien bei Ihrer agilen Transformation, um neue Wege zur Verbesserung der Zusammenarbeit in virtuellen Umgebungen zu finden.
- Befähigen Sie „nicht-agile“ Beteiligte sowie Befürworterinnen und Befürworter dazu, Treiberinnen und Treiber des Wandels in ihrer Behörde zu werden.
- Eine schnelle Umsetzung und Festigung der Strukturen in der Linienorganisation nach der getroffenen Entscheidung für eine Agile Transformation sind empfehlenswert.
- Schnelligkeit am Anfang ist entscheidend. Schaffen Sie Erfolge, bevor potenzielle Kritiker die Veränderung anzweifeln.

⁸ Vgl. Henke, A. (2022). Agilität in der Praxis Teil 2, Behördenspiegel, 2022 (2). S. 32.

II. PM²-AGILE - DIE METHODE DER EUROPÄISCHEN KOMMISSION

1. Einführung

Für Teil II dieses Leitfadens wurde der Leitfaden „PM²-Agile“ (Guide 3.0.1) vollständig aus dem Englischen übersetzt und dient als Einführung in PM²-Agile, die agile Erweiterung der PM²-Methodik.

Dieser Leitfaden richtet sich an:

- Praktizierende agiler Ansätze, die nach einem Rahmenwerk suchen, das ihnen helfen könnte, ihren Ansatz in bestehende Strukturen und geschäftsbezogene Regelungen zu integrieren.
- PM² nutzende Organisationen, die mit der Anwendung agiler Werte und Prinzipien auf ihre Arbeitsweise beginnen wollen.
- Führungskräfte von Organisationen, die mehr über agile Ansätze wissen müssen, entweder zur Evaluation zur eigenen Verwendung oder zur Verbesserung der Interaktion innerhalb von Teams, die derzeit auf diese Weise arbeiten.
- jede(n), die/der an der Übernahme agiler Praktiken interessiert ist oder mehr über diese erfahren möchte.

„Agil“ ist ein Sammelbegriff für eine Reihe von Praktiken, die auf einer Reihe von Grundwerten und Prinzipien beruhen, insbesondere bei der Softwareentwicklung, bei denen Anforderungen und Lösungen durch Zusammenarbeit zwischen selbst organisierten, cross-funktionalen (funktionsübergreifenden) Teams entstehen. Sie fördern adaptive Planung, evolutionäre Entwicklung, frühzeitige Bereitstellung und kontinuierliche Verbesserung sowie schnelle und flexible Reaktion auf Veränderungen. Zur Erweiterung dieser Aspekte wurden zusätzliche Modelle und Prozesse wie das Lean Start-up-Modell und der Lean UX (user experience) Prozess zu wesentlichen Tools, die Menschen, Teams und Organisationen dabei helfen, agile Methoden erfolgreich anzuwenden.

Jedoch sind Anwendende agiler Ansätze mit Herausforderungen im Zusammenhang mit Folgendem konfrontiert:

- agile Vorgehensweise bei gleichzeitiger Einhaltung von Unternehmensprozessen, -strukturen und -regelungen, einschließlich IT-Governance- und -Budgetregelungen, Programmstrukturen, Architektur- und Interoperabilitätsanforderungen.
- die einheitliche Anwendung agiler Methoden innerhalb der Organisation. Agiles Arbeiten kann unter Umständen nicht so effektiv wie sonst sein, wenn einige Teams einen agilen Ansatz übernehmen, andere wiederum nicht.

PM²-Agile ist darauf ausgerichtet, diese Herausforderungen anzugehen. Darüber hinaus schafft die Einbindung von Agilität in das PM²-Gesamtrahmenwerk sowie die PM²-Gemeinschaft und -Kultur die Grundlagen für ein besseres Lernen und Verbesserungen auf Unternehmens- und persönlicher Ebene.

Die PM²-Agile-Methodik bietet:

- gemeinsame Begrifflichkeiten, um die Kommunikation zwischen und innerhalb von Projektteams zu fördern
- das PM²-Agile-Modell

- die agilen Werte und Prinzipien
- die Grundlagen agiler Praktiken
- zusammengefasste Informationen über einige agile Tools und Techniken
- eine Beschreibung einer Reihe empfohlener Artefakte zur Dokumentation des Fortschritts bei der Steuerung von Aktivitäten unter Einsatz agiler Prinzipien und zur zugehörigen Berichterstattung
- Verweise und Anregungen für weiterführende Literatur zu Themen im Zusammenhang mit Agilität

Der PM²-Agile-Leitfaden richtet sich in erster Linie an die Bedarfe von Softwareentwicklungsprojekten. Auch werden agile Ansätze häufig außerhalb der Softwareengineering-Umgebung erfolgreich angewandt. Daher kann der Großteil der in diesem Leitfaden beschriebenen Praktiken, Tools und Techniken auch bei Nicht-IT-Projekten eingesetzt werden. Darüber hinaus haben die (angepassten) agilen Werte und Prinzipien die PM²-Agile-Mindsets in maßgeblichem Umfang beeinflusst, die eine Orientierung für Projektteams bieten sollen, die mit dem Projektmanagement verbundenen Komplexitäten in ihren Organisationen zu bewältigen.

PM²-Agile ersetzt keine Komponenten von PM². Dieser Leitfaden und der übrige Teil des PM²-Agile-Angebots erweitert PM² durch die Bereitstellung spezifischer Methoden, Tools und Techniken, die Teams dabei helfen und sie befähigen, agile und Lean-Prinzipien im Rahmen ihrer PM²-Projekte zu übernehmen.

Diese Methoden, Tools und Techniken sind darauf zugeschnitten, den reibungslosen Informationsfluss, kohärente Strukturen, Zuständigkeiten und harmonisierte Artefakte sicherzustellen. PM²-Agile erläutert, wie PM² auf Projekte anzuwenden ist, bei denen mindestens ein Team, das Teil des Projektkernteams (PCT) ist, agile Praktiken anwendet.

Die Kapitel können systematisch durchgearbeitet werden, um die Methodik kennenzulernen oder als Nachschlagewerk verwendet werden.

Bei der Anwendung von PM² können auch die vom Centre of Excellence in PM² (CoEPM²) zur Verfügung gestellten Ressourcen genutzt werden. Das CoEPM² wurde eingerichtet, um der Europäischen Kommission und den EU-Institutionen eine qualitativ hochwertige Projektmanagementmethodik sowie Unterstützungs- und Beratungsleistungen im Bereich Projektmanagement bereitzustellen. Das Project Support Network (PSN) unterstützt die PM²-Methodik, koordiniert ein interinstitutionelles Projektunterstützungsnetzwerk und fördert durch die Open-PM²-Initiative eine breite Anwendung der PM²-Methodik.

Weiterführende Informationen sind über die folgenden Websites zu finden:

https://pm2.europa.eu/index_de

<https://joinup.ec.europa.eu/collection/pm2-project-management-methodology>

Das CC GroßPM erhofft sich hiervon, dass Beschäftigte mit Projektaufgaben auf einen größeren Wissenspool zurückgreifen können, um ihre Aufgaben zu erfüllen.

Hinweis: Das CC GroßPM hat die in PM² verwendeten Begriffe gemäß der im deutschsprachigen Raum verwendeten Terminologien für Projektmanagement übersetzt, soweit diese nicht bereits in den allgemeinen Sprachgebrauch übernommen wurden. So wurde beispielsweise das „Agile Project Core Team“ als agiles Projektkernteam übersetzt. In der folgenden Übersetzung werden aber die von PM² verwendeten englischen Abkürzungen der Rollen nicht übersetzt (z. B. A-PCT für agiles Projektkernteam). Das CC GroßPM hat sich hierfür bewusst entschieden, um in länderübergreifenden Projekten auf Ebene der Abkürzungen und Kurzbezeichnungen eine einfachere Vergleichbarkeit zu gewährleisten. Als Vorbild diente dafür die französische PM²-Übersetzung, in der dies ebenfalls so gehandhabt wurde.

2. Was ist Agile?

„Agile“ ist ein Ansatz, der auf einer Reihe von Grundwerten und Prinzipien beruht und dessen Schwerpunkt auf der Bereitstellung vollwertiger Lösungen liegt. Er fördert adaptive Planung, evolutionäre Entwicklung, frühzeitige Bereitstellung und kontinuierliche Verbesserung sowie schnelle und flexible Reaktion auf Veränderungen.

Ein agiler Ansatz greift die inhärente Unsicherheit der Projektumgebung auf und schafft eine Organisation, die hochgradig adaptiv ist, indem sie kurzfristige Feedback-Schleifen einsetzt, um auf Veränderungen bei Produktanforderungen schnell reagieren zu können. Der Ansatz dient dazu, alle Projektprozesse fortlaufend zu verbessern und zu straffen.

Hauptmerkmale von Agilität:

- Der Schwerpunkt liegt auf der frühzeitigen und häufigen Bereitstellung von Mehrwert.
- Entscheidungen beruhen auf dem, was bekannt ist.
- Sie beinhaltet eine enge Zusammenarbeit zwischen allen beteiligten Parteien.
- Stakeholderinnen und Stakeholder werden auf allen Ebenen kontinuierlich eingebunden.
- Die Erstellung von Plänen erfolgt unter der Beteiligung der Teammitglieder.
- Inkrementelle Entwicklung mit kurzen Zyklen.
- Umfangsmanagement durch fortlaufende (Re-)Priorisierung der Arbeitselemente.
- Aufgreifen von Veränderungen, kontinuierliches Lernen und fortlaufende Verbesserung.
- Dokumentation und Kontrolle nur im erforderlichen Umfang.

Agilität hat seine eigenen Wurzeln, wurde jedoch maßgeblich durch die Lean-Philosophie beeinflusst. Die Lean-Kernprinzipien sind die Transparenz des Wertstroms und die Minimierung von Verschwendung. Beim Lean Management wird zwischen den folgenden sieben Arten der Verschwendung unterschieden:

1. Bestände – Halberzeugnisse (auch als „unfertige Erzeugnisse“ bezeichnet).
2. Überproduktion – es wird mehr produziert als nachgefragt.
3. Zusätzliche Verarbeitung – zusätzliche Schritte im Prozess, die nicht wirklich erforderlich sind.
4. Transport – Beförderung von Gütern von einem Ort zum anderen.
5. Wartezeiten – Lücken zwischen Verarbeitungsschritten.
6. Bewegung – Bewegungen im Rahmen des Prozesses.
7. Mängel – Fehler bei den Liefergegenständen, die sich auf ihre Features/Funktionalität auswirken.

Agile Ansätze greifen diese Formen von Verschwendung auf unterschiedliche Weise auf. Beispielsweise werden unfertige Erzeugnisse (Verschwendungsart 1) durch die richtige Granularität von Arbeitselementen und die Verringerung von Abhängigkeiten reduziert. Dies erfolgt Hand in Hand, und gemeinsam stellen die Ansätze sicher, dass zur Fertigstellung einer Arbeitseinheit keine parallelen Arbeiten erforderlich sind.

Überproduktion (Verschwendungsart 2) kann durch zusätzliche Features bedingt sein. Diese Quelle von Verschwendung kann durch eine optimal priorisierte Arbeitselementeliste (WILL), die Interaktion mit den Auftraggeberinnen und Auftraggebern im gesamten Projektlebenszyklus und die Einbindung aller Teammitglieder bei der Planung kontrolliert werden. Eine weitere Form von Überproduktion

zeigt sich, wenn es am Ende eines Zyklus mehr lieferfertige Produkte gibt, als wirksam getestet werden können. Dies kann bei einem agilen Ansatz durch eine kurze Feedback-Schleife im Planungsprozess sowie durch funktionsübergreifende Teams und einen transparenten Wertstrom minimiert werden. Überproduktion ist nur eine Ursache von Prozessengpässen, und es gibt agile Techniken, die dabei helfen, solche Engpässe zu minimieren.

Nicht wertschöpfende Aktivitäten (Verschwendungsart 3) und Übergaben (Verschwendungsart 4) werden durch die Einführung selbst organisierter, funktionsübergreifender Teams in Kollokation verringert. Durch diese Struktur werden effektive, effiziente und stabile Arbeitsmethoden geschaffen.

Wartezeiten (Verschwendungsart 5) und das häufige Wechseln zwischen Aufgaben (Verschwendungsart 6) werden durch die Verringerung unfertiger Erzeugnisse, eine frühzeitige und häufige Bereitstellung sowie durch die Sicherstellung klarer Prioritäten und ein Verständnis jedes Arbeitselements vermindert.

Mängel (Verschwendungsart 7) werden vermindert, indem die Qualität im Entwicklungsprozess sichergestellt wird, statt in Erkennungs- und Korrekturmaßnahmen zu investieren. Neben der agilen Gesamtorganisation, die dazu beiträgt, gibt es konkrete Techniken wie die testgetriebene Entwicklung und das Pair Programming, die dabei helfen, die Fehlerquote zu verringern.

Die Minimierung von Verschwendung ist lediglich das vorrangige Prinzip beim Lean-Management. Alle anderen Prinzipien werden bei agilen Ansätzen ebenfalls angewandt. Obgleich agile Ansätze vom Lean-Management wesentlich beeinflusst sind, sind sie durch eigenständige Merkmale gekennzeichnet. Beispielsweise sind evolutionäre Entwicklung, funktionsübergreifende Teams und der Ansatz zur internen Zusammenarbeit und Zusammenarbeit mit den Auftraggeberinnen und Auftraggebern für agile Praktiken spezifisch. Diese Praktiken beruhen auf den in Abschnitt 2.1 beschriebenen agilen Werten und Prinzipien.

Agile Ansätze versprechen höhere Chancen für Projekt- und Produkterfolg sowie eine stärkere Einbeziehung von Mitarbeitenden und Dienstleistern, was zu Softwaresystemen führt, die dank einer stärkeren Einbindung von Stakeholderinnen und Stakeholdern und eines intensiveren Aufgreifens ihrer sich entwickelnden Bedarfe ein besseres Nutzererlebnis bieten.

Derzeit gibt es rund ein Dutzend agiler Methoden und viele Hybridformen. Davon wurden einige in größerem Umfang aufgegriffen und erfreuen sich in den Medien großer Popularität. Dies entfachte einen Hype, der auch zur Verbreitung vieler Missverständnisse geführt hat. Daher ist es wichtig, die agilen Werte und Prinzipien klarstellend zu erläutern. Diese Klarstellung dient als Orientierung bei der Abkehr von derzeit gängigen Praktiken und Techniken zugunsten von PM²-Agile.

2.1 Agile Werte

Im Jahr 2001 führten siebzehn Personen ein Brainstorming durch, um eine bessere Art zu Arbeiten zu finden. Daraus entstand das sogenannte „Manifest für agile Softwareentwicklung“ (*Manifesto for Agile Software Development*).

Das Ergebnis der Brainstorming-Sitzung wurde auf <https://agilemanifesto.org/> veröffentlicht. Dieser Inhalt wird in der Regel als das „Agile Manifest“ bezeichnet, das vier Werte und zwölf Prinzipien definiert.

Das Agile Manifest bildet eine Grundlage für das PM²-Agile-Rahmenwerk. Diese Grundlage wird angepasst, um deren Werte und Erfahrungswerte aufgrund vieler Jahre der agilen Erfahrung wie folgt anzupassen:

- Der Fokus des ursprünglichen Manifests lag auf der Softwareentwicklung – PM²-Agile erweitert den Umfang, um den Fokus auf die Bereitstellung der Lösung zu legen.
- Der Fokus des ursprünglichen Manifests lag auf den Kundinnen und Kunden – bei PM²-Agile sind alle Stakeholderinnen und Stakeholder gleichermaßen wichtig.
- Der Fokus des ursprünglichen Manifests lag auf Entwicklungsteams – PM²-Agile betrachtet das gesamte Ökosystem der Organisation und wie es verbessert werden kann.

Jede der vier Wertaussagen im Agilen Manifest wird im Format „X mehr als Y“ präsentiert. Das Wichtigste bei den Aussagen ist es, zu verstehen, dass man die Konzepte auf der rechten Seite der Gleichung sehr wohl wertschätzen sollte, die auf der linken Seite genannten Aspekte jedoch noch mehr. Das Manifest sollte am besten so ausgelegt werden, dass es Präferenzen und nicht Alternativen definiert sowie einen Fokus auf bestimmte Bereiche legt, ohne andere außer Acht zu lassen.

Die vier Werte des Agilen Manifests sind im Einzelnen:

1. Individuen und Interaktionen mehr als Prozesse und Werkzeuge.
2. Funktionierende Lösungen mehr als umfassende Dokumentation.
3. Zusammenarbeit mit den Stakeholderinnen und Stakeholdern mehr als Vertragsverhandlung.
4. Reagieren auf Veränderung mehr als das Befolgen eines Plans.

Erster agiler Wert – Individuen und Interaktionen mehr als Prozesse und Werkzeuge

Prozesse und Werkzeuge sind sehr hilfreich, jedoch werden sie zu einem bestimmten Zeitpunkt konzipiert und beruhen auf dem zu diesem Zeitpunkt verfügbaren Wissen. Da sie jedoch zu einem späteren Zeitpunkt implementiert und genutzt werden, können sie häufig nicht an die internen und externen Veränderungen in Projekten angepasst werden. Noch wichtiger ist, dass Arbeiten wie Softwareentwicklung auf dem Wissen und der Kreativität von Menschen aufbauen.

Stoßen Prozesse und Werkzeuge an ihre Grenzen, können die Intelligenz und Fähigkeit von Menschen zur effektiven Zusammenarbeit die erforderliche Reaktions- und Problemlösungskapazität bereitstellen, indem neue Ideen und Einblicke aus zahlreichen verschiedenen Perspektiven generiert werden. Da bei Agilität die inhärente Unsicherheit bei Projekten anerkannt wird, werden Individuen und Interaktionen mehr als Prozesse und Werkzeuge geschätzt. Dies ist einer der wesentlichen Befähiger des Lean UX-Prozesses, einer wichtigen Säule von PM²-Agile.

Zweiter agiler Wert – Funktionierende Lösungen mehr als umfassende Dokumentation

Die für die Dokumentation aufgewendete Zeit geht von der Zeit für Entwicklung und Lernen ab, indem Prototypen der Projektliefergegenstände frühzeitig im Projekt präsentiert werden. Selbst wenn ein großer Teil eines Iterationsliefergegenstands verworfen wird, lohnt sich diese Anstrengung im Austausch für das gewonnene Feedback. Da durch dieses Feedback häufig die Spezifikationen geändert werden, ist es umso wichtiger, der schlanken Dokumentation (oder sogar noch besser der schlanken Modellierung) den Vorzug zu geben, da – falls der Großteil der Spezifikationen verändert wird – die Anstrengungen zur Spezifizierung umsonst waren. Darüber hinaus kann man häufig die echten Anforderungen nur dann erkennen, wenn die anfänglichen Ideen der Auftraggeberinnen und Auftraggeber mit der Realisierung auf Grundlage des sich entwickelnden Verständnisses der Entwickler zusammengeführt werden, um etwas Konkretes statt etwas Imaginäres zu schaffen.

Des Weiteren wird es Stakeholderinnen und Stakeholdern viel leichter fallen, eine funktionierende Lösung zu verstehen, statt ein komplexes technisches Diagramm mit einer Beschreibung ihrer internen Funktionsweise oder abstrahierenden Beschreibung ihrer Nutzung. Dokumentation hat ihren Stellenwert. Der Liefergegenstand Dokumentation ist genau wie Benutzer- und Betriebshandbücher in der Tat Teil der Gesamtlösung.

Das primäre Ziel jedes agilen Projektkernteams (A-PCT) ist es, Lösungen zu schaffen, keine Dokumente. Daher ist es wichtig, sich auf die Hauptliefergegenstände zu konzentrieren.

Bei diesem agilen Wert sind drei wichtige Punkte im organisatorischen Kontext zu beachten.

- Im ursprünglichen Agilen Manifest wird in dieser Wertaussage der Begriff „Software“ statt „Lösungen“ verwendet. Innerhalb einer Organisation wird der Begriff „Lösungen“ vorgezogen. Damit soll
 - hervorgehoben werden, dass die Software nur ein Teil eines IT-Projekts ist.
 - die Anwendung von PM²-Agile auf Nicht-IT-Projekte ermöglicht werden.
- Der agile Ansatz sollte auf ein Verständnis des/der Hauptliefergegenstands/-gegenstände angelegt werden. In Fällen, in denen der agile Ansatz auf Nicht-IT-Projekte angewandt wird, könnte es sich beim Hauptliefergegenstand um ein Dokument handeln. In diesem Fall sollte „Dokumentation“ als die Art von Dokumentation verstanden werden, die die Erstellung der Hauptliefergegenstände unterstützt.
- Es besteht das Risiko, dass „funktionierende Lösungen“ in einer Weise interpretiert werden, die zur Entwicklung lokal optimaler Lösungen oder „lokaler Optima“ (anstelle von global optimalen Lösungen) führt. Daher tragen als erfolgreich geltende Projekte gelegentlich zu einer Doppelung von IT-Investitionen bei (Entwicklung von Funktionen, die über eine vorhandene Anwendung zur Verfügung stehen) oder sind in einer so starren, geschlossenen und nicht interoperablen Weise konzipiert, dass umfangreiche zusätzliche Investitionen erforderlich sind, um die Anwendung mit anderen Anwendungen zu integrieren.

Dritter agiler Wert – Zusammenarbeit mit Stakeholderinnen und Stakeholdern mehr als Vertragsverhandlung

Verträge werden häufig zu einem frühen Zeitpunkt unterzeichnet, was zu aufwendigen Änderungsmanagementverfahren führt. Da Änderungen unvermeidlich sind, ist es besser, sie durch eine effiziente Zusammenarbeit mit Stakeholderinnen und Stakeholdern zu bewältigen. Darüber hinaus täuschen Verträge eine falsche Sicherheit vor. Es ist wichtig zu erkennen, dass das Wissen um das, was erforderlich und was zu tun ist, auf der Zusammenarbeit zur Lösung der Probleme und Nutzung von Möglichkeiten beruht, die nicht im Vertrag vorgesehen waren.

Durch die Zusammenarbeit mit den Stakeholderinnen und Stakeholdern wird eine auf Konsens beruhende Entscheidungskultur geschaffen, durch die ein gemeinsames und geteiltes Verständnis des zu lösenden Problems und der möglichen Lösungen erlangt werden kann.

(Im ursprünglichen Agilen Manifest wird in dieser Wertaussage der Begriff „Kundinnen“ bzw. „Kunde“ statt „Stakeholderinnen“ bzw. „Stakeholder“ verwendet.)

Vierter agiler Wert – Reagieren auf Veränderung mehr als das Befolgen eines Plans.

Es ist sehr wichtig, einen Plan zu haben. Jedoch sollte dieser nur für die aktuelle Iteration ausgearbeitet sein. Dadurch wird ermöglicht, Anpassungen auf Basis des während der aktuellen Iteration eingehenden Feedbacks vorzunehmen und die nächste Iteration zu planen. Agile Praktiken ermöglichen eine

langfristige Planung, jedoch sollten diese Pläne übergeordneter Natur sein und auf dem Verständnis fußen, dass sie gegebenenfalls häufig zu aktualisieren sind. Dieser vierte Wert ist im Kern der wesentliche Befähiger der Feedback-Schleife „Entwickeln, Messen, Lernen“ (Build-Measure-Learn) im Lean Start-up-Modell, einer weiteren wichtigen Säule von PM²-Agile.

2.2 Agile Teams

Der wichtigste Produktivitätsfaktor in einem Team sind die Menschen und die Art, wie sie miteinander interagieren. Agile Teams organisieren sich selbst, sind funktionsübergreifend, veränderlich und durch eine hohe Zusammenarbeit gekennzeichnet.

Selbstorganisation bedeutet, dass jede(r) im Team zusammenarbeitet, um die beste Weise zu ermitteln, die Arbeit durchzuführen. Das Team ist funktionsübergreifend und besteht aus Mitgliedern mit der benötigten kombinierten Expertise zur Entwicklung und Bereitstellung einer Lösung. Es besteht aus Personen mit verschiedenen technischen Kenntnissen. Das Team schließt auch Stakeholderinnen und Stakeholder mit dem erforderlichen Fachwissen ein. Veränderlichkeit bedeutet, dass sich die Zusammensetzung des Teams im Verlauf der Zeit verändern kann.

Obwohl die PM²-Agile-Rollen bestimmte spezifische Zuständigkeiten aufweisen, haben alle Mitglieder des Projektteams in einem Projekt, bei dem PM²-Agile zur Anwendung kommt, einige gemeinsame Zuständigkeiten.

- Bereitstellung der Outputs, die die Erwartungen der Stakeholderinnen und Stakeholder unter Berücksichtigung der Qualitätsstandards der Organisation am besten erfüllen.
- Einhaltung der Zusagen.
- Einhaltung des Projektbudgets und Streben nach Effizienz bei gleichzeitiger Bereitstellung der Outputs.
- Förderung eines kollaborativen und kooperativen Arbeitsumfelds.
- Austausch von Informationen, Wissen und Expertise mit Kolleginnen und Kollegen.
- Hilfestellung für andere Teammitglieder, um ihre Fähigkeiten und Kompetenzen zu entwickeln.
- Proaktiv und „Grow-“Mindset, bei dem der Ansatz des kontinuierlichen Lernens und der kontinuierlichen Entwicklung sowohl bei weichen als auch bei harten Skills verinnerlicht ist.
- Leistung eines Beitrags zur kontinuierlichen Verbesserung von PM²-Agile.

Agile Teams arbeiten auf eine sehr kollaborative Weise zusammen, indem sie die wirksamsten Kommunikationstechniken für ihre Situation übernehmen und bestrebt sind, so eng wie möglich zusammenzuarbeiten.

Ziel ist es sicherzustellen, dass

- jede(r) das Gefühl hat, zum Team zu gehören und gemeinsam im Team zu sein. Es sollte keine „Außenseiter“, keine „die anderen“, sondern nur ein „uns“ geben.
- das Team beinhaltet alle Personen, die zur Entwicklung des Systems erforderlich sind. Ein autarkes Team sollte über die Fähigkeiten und das Wissen verfügen, um den Auftrag zu erledigen.

Das ist nicht immer möglich. Manchmal besteht das Erfordernis, externe Experten für kürzere Zeiträume für bestimmte Aufgaben hinzuziehen.

- jede(r) im Team nach ihren/seinen Möglichkeiten ihren/seinen Beitrag leistet. Es findet eine Verlagerung weg von Spezialisten, die sich auf eine spezifische Kategorie von Arbeit konzentrieren, wie Analysen oder Datenbank-Administration, hin zu Teams mit multidisziplinären Mitgliedern statt.
- das Team selbst organisierend ist. Die Personen, die am besten für das Planen und Organisieren der Arbeit geeignet sind, sind diejenigen, die die Arbeit machen. Dies führt zu besseren Schätzungen, realistischeren Zeitplänen und einem erhöhten Bekenntnis des Teams zum Plan.
- das Team in einem gleichmäßigen Tempo auf unbegrenzte Zeit agiert. Keine heroischen Beiträge, keine Burn-outs.
- jede(r) eng zusammenarbeitet.

2.3 Kontinuierliche Verbesserung und validiertes Lernen

Kontinuierliche Verbesserung und stetiges Lernen sind eine Kernphilosophie agiler Arbeitsweisen. Dies erfolgt durch häufige Reviews und Retrospektiven, d. h. Routinen, mit denen Teams aus Erfahrungen und Planänderungen lernen und das Gelernte in den nächsten Zyklus einbringen. Reviews sind ein gutes Instrument, um Feedback von den Stakeholderinnen und Stakeholdern einzuholen und Erkenntnisse über die Lösung zu gewinnen. Retrospektiven sind ein gutes Instrument, um Feedback und Erkenntnisse über den Prozess und das agile Team selbst zu gewinnen. Beide tragen maßgeblich zum Erfolg von Projekten und Lösungen bei, da sie einen roten Faden für den gesamten Verbesserungsprozess bilden. Ohne wirksame und sinnvolle Lernprozesse wäre dies jedoch nicht möglich. PM²-Agile betrachtet Lernen als einen wesentlichen Baustein und wendet im Rahmen des Lean Start-up-Modells, das in PM²-Agile ebenfalls eine wichtige Rolle spielt, das Konzept des validierten Lernens an.

Auch wenn die am Ende eines Projekts zusammengetragenen Erkenntnisse weiterhin wertvoll sind, müssen häufige Reviews und Retrospektiven im gesamten Projektverlauf verinnerlicht und zur Gewohnheit gemacht werden, denn dort finden kontinuierliche Verbesserung und validiertes Lernen statt.

Entwickeln, Messen, Lernen

Die Teams folgen einer Feedback-Schleife, um fortlaufend aus ihren Hypothesen zu lernen und sich weiter zu verbessern. Hierzu nutzen sie die Ideen, das Produkt und die Daten als Anstoß.

In der Entwickeln-Messen-Lernen-Schleife ist Geschwindigkeit ein wichtiges Element des kontinuierlichen, validierten Lernens, das später die richtige Entwicklung ermöglicht. Die Wirksamkeit eines Teams bemisst sich nach seiner Fähigkeit, Ideen zu generieren, daraus schnell (mittels Rapid Prototyping) ein Minimum Viable Product zu erschaffen, die Marktresonanz darauf zu messen und durch Analyse der generierten Daten aus diesem Experiment zu lernen. Mit anderen Worten: Es handelt sich um einen Lernzyklus, in dem Ideen in Produkte umgesetzt, die Reaktionen und das Verhalten der Nutzenenden in Bezug auf die entwickelten Produkte gemessen werden und anschließend darüber entschieden

wird, ob die Idee unverändert weiterverfolgt oder „umgeschwenkt“ (geändert) werden soll. Dieser Prozess wird beliebig oft nach Bedarf wiederholt.⁹

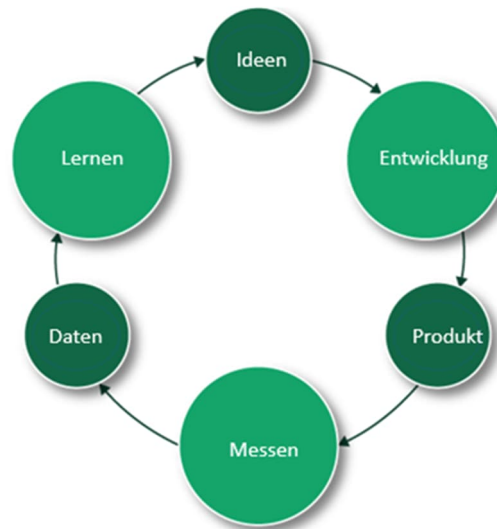


Abbildung 7: Minimierung der Gesamtzeit durch die Schleife

Minimum Viable Product (MVP)

Der Aufbau von Start-ups ist mit zahlreichen Risiken verbunden und erfordert das richtige Timing. Für Start-ups ist es sicherer, mit ihrer Idee zu experimentieren, indem sie frühzeitig Prototypen erstellen und sie bei potenziellen Kundinnen und Kunden möglichst früh und fortlaufend testen. So entstand die Idee eines MVP, weil sie nicht viel Zeit hatten, um günstig und schnell ein Produkt zu entwickeln, das ihnen Erkenntnisse liefern und ermöglichen würde, Marktführer zu werden.

Die Erstellung eines MVP bedeutet nicht unbedingt, dass ein voll funktionsfähiger Softwarecode entwickelt werden muss. Am wichtigsten ist, dass die Nutzung des MVP gemessen werden kann und die erhobenen Daten in den Lernprozess einfließen und neue Ideen anstoßen. Dies hält den Kreislauf in Gange und sorgt dafür, dass sich die Lösung zur richtigen Lösung weiterentwickelt. Der Lean UX-Prozess (ein weiterer wichtiger Baustein von PM²-Agile) geht Hand in Hand mit dem Lean Start-up-Modell, da es die Erschaffung des MVP mit Rapid Prototyping und der Erzielung schneller Lernergebnisse unterstützt.

Das Lean Start-up-Modell und der Lean UX-Prozess haben auch weltweit Eingang in den öffentlichen Sektor gefunden, um die Kostenunsicherheit und Risiken in späteren Phasen der Produktentwicklung, an denen viele Stakeholderinnen und Stakeholder und öffentliche Haushalte beteiligt sind, zu verringern.

Das Minimum Viable Product (MVP) ist das Ergebnis eines fortlaufenden validierten Lernprozesses, das grünes Licht für die inkrementelle Entwicklung oder auch einen Hinweis darauf gibt, dass eine Idee nicht weiterverfolgt werden sollte.

⁹ Adaptiert aus: The Lean Startup: Eric Ries, Currency International Edition 2017



Abbildung 8: Was ein MVP ist und was es nicht ist

Zweck eines MVP

- Möglichkeit, eine Produkthypothese mit minimalen Ressourcen zu testen
- Beschleunigtes Lernen
- Weniger verschwendete Entwicklungszeit
- Möglichst schnelle Bereitstellung des Produkts an frühere Anwenderinnen und Anwender, um aus deren Nutzung zu lernen
- Grundlage für andere Produkte

In der nachfolgenden Abbildung sind alle wesentlichen Elemente dargestellt, die bei der Entwicklung des richtigen Produkts mit geringeren Risiken bis zur Marktreife zu berücksichtigen sind. Es ist sehr zu empfehlen, alle diese Elemente einzubeziehen, anstatt Abkürzungen zu nehmen und sich nur auf ein oder zwei Aspekte zu konzentrieren.

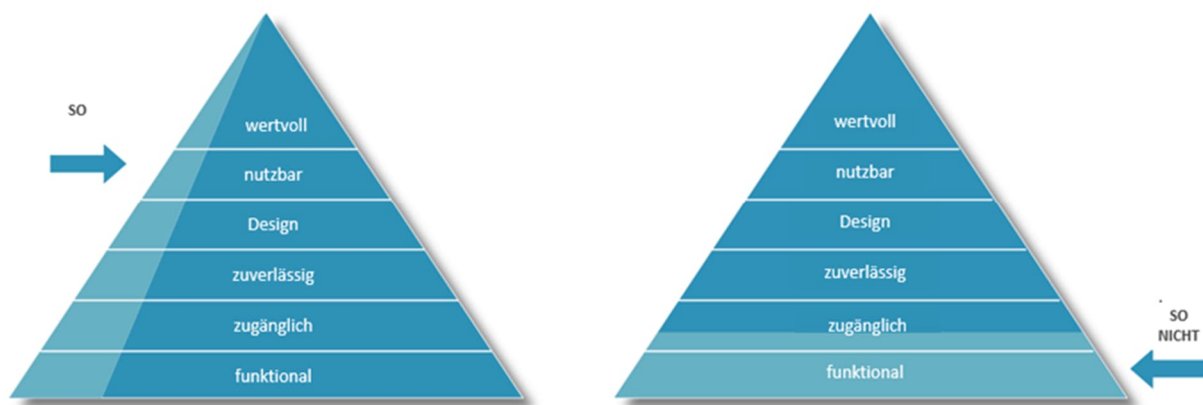


Abbildung 9: Das richtige MVP erfüllt alle Elemente

Ein MVP muss die folgenden wesentlichen Elemente enthalten und sich auf diese Elemente konzentrieren:

- Funktionalität: Die Funktionen müssen einen klaren Wert für die Nutzenden erbringen/nachweisen.
- Zugänglichkeit: Das MVP muss die Standards für Zugänglichkeit erfüllen.
- Verlässlichkeit: Angemessene Nachweise dafür, dass das ursprüngliche Problem oder Ziel in einer schlüssigen und relevanten Weise, die eine Fortführung rechtfertigt, gelöst bzw. erreicht wird.

- Design: Das Design des MVP muss höchsten Branchenstandards entsprechen.
- Nutzbarkeit: Das MVP muss einfach und intuitiv zu verwenden sein.
- Wert: Das MVP liefert einen Wert für die Organisation und erfüllt die Nutzerbedürfnisse.

Erfolgskriterien

- Alle wesentlichen Elemente werden eingehalten.
- Key Performance Indicators (KPIs) und Metriken werden gleich zu Beginn definiert.
- Häufige Nutzungsuntersuchungen in der Nutzerschaft und Validierung durch Stakeholderinnen und Stakeholder. Die Untersuchungen werden nach Möglichkeit auf höchstens fünf Nutzende beschränkt, um keine Zeit mit übermäßigen Analysen zu verschwenden.

2.4 Mythen über Agilität

Manche (sowohl in der IT als auch auf der Fachseite) gehen mit großer Begeisterung an die Anwendung agiler Ansätze bei ihrer Arbeit heran. Andere stehen agilen Methoden aus verschiedenen Gründen skeptisch gegenüber, was auch an bestimmten Mythen über Agilität liegen mag. Einige dieser Mythen sind nachfolgend aufgeführt:

Mythos: Agilität ist die Lösung für alles

- Agile Projekte können wie jedes andere Projekt scheitern. Aber sie scheitern vielleicht früher und sparen so Ressourcen.

Mythos: Agile Teams erstellen keine Dokumentation

- Sie erstellen keine unnötige Dokumentation.

Mythos: Agilität heißt, dass nichts geplant wird

- In der agilen Entwicklung findet eine umfangreiche Planung auf mehreren Ebenen und mit hoher Transparenz statt.
- Es gibt einen Releaseplan, Iterationspläne und Stand-up-Meetings, in denen die tägliche Arbeit geplant wird.
- Es werden verschiedene Planungstools (z. B. Produkt- und Iterations-Backlogs) eingesetzt.

Mythos: Agilität ist undiszipliniert

- Tatsächlich ist agile Entwicklung äußerst diszipliniert. Praktiken wie kontinuierliche Integration, automatisiertes Testen, Iterationsreviews und Retrospektiven, Iteration und Release-Planning sind einige Beispiele, die zeigen, wie diszipliniert agile Entwicklung sein muss.
- Bei den meisten agilen Praktiken werden für alle Iterationen und Besprechungen zeitliche Begrenzungen (sog. „Timeboxes“) festgelegt.
- Es besteht die Verpflichtung, das Produkt regelmäßig zu „liefern“.
- Die Verbesserung der Praktik ist Teil der Praktik.

Mythos: Agilität ist nichts anderes als Scrum.

- Scrum ist nicht der einzige agile Ansatz, es gibt mindestens zehn weitere.
- Scrum konzentriert sich vor allem auf den Aspekt der „Konstruktion“ von Projekten.

- Scrum behandelt keine Aktivitäten für das Management des gesamten Projektlebenszyklus.
- Es stimmt, dass Scrum heute das bekannteste agile Rahmenkonzept ist.

Mythos: Bei Agilität gibt es keine Architektur

- Die Lösungsarchitektur wird auf das minimiert, was bekannt ist.
- Lösungsarchitektur ist wertvoll und wird von agilen Teams in der einen oder anderen Form verwendet.

3. Überblick über PM²-Agile

PM²-Agile ist ein Angebot, das die Verwendung agiler Praktiken bei Projekten und Arbeitsaufgaben jeder Art ermöglichen und unterstützen soll. Auch wenn derartige Praktiken vielleicht schon bestehen, kann es aufgrund der Größe der Organisation und der wechselseitigen Abhängigkeiten in ihrem Ökosystem schwierig sein, den Erfolg eines Projekts sicherzustellen.

Anders als bei der Umsetzung agiler Konzepte in kleinen Betrieben muss in großen Organisationen unter Umständen die Verbindung von Agilität mit anderen Projektansätzen ermöglicht werden. Gleichzeitig sind verschiedene Anforderungen in Bezug auf IT-Governance, Unternehmensarchitektur und Interoperabilität zu erfüllen.

PM²-Agile unterstützt von seiner Konzeption her sowohl diejenigen, die agile Methoden bereits praktizieren, als auch diejenigen, die bereit sind, in ihren Projekten agile Ansätze auszuprobieren.

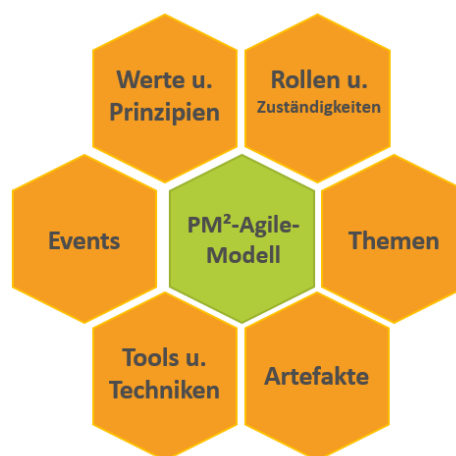


Abbildung 10: Elemente von PM²-Agile

3.1 Prinzipien von PM²-Agile

Um verständlicher zu machen, worum es bei agiler Softwareentwicklung geht, haben die 17 Verfasser des Manifests für agile Softwareentwicklung in zwölf Prinzipien dargelegt, was sie unter „Agilität“ verstehen. PM²-Agile hat diese Prinzipien – mit einigen Modifizierungen – übernommen.

Die zwölf Prinzipien von PM²-Agile:

- Höchste Priorität ist es, die Auftraggeberin oder den Auftraggeber durch frühe und kontinuierliche Lieferung wertvoller Lösungen zufriedenzustellen.
- Anforderungsänderungen sind erwünscht.
- Generiere regelmäßig Wert durch funktionierende Lösungen.
- Fachleute und das Projektkernteam (PCT) müssen während des gesamten Projekts zusammenarbeiten.
- Stelle Teams mit motivierten Menschen zusammen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen, um selbstorganisiert zu arbeiten, und vertraue darauf, dass sie die Aufgabe erledigen.
- Das effizienteste und wirksamste Kommunikationsmittel ist das persönliche Gespräch.

- Das wichtigste Maß für Fortschritt sind der Wert und die Brauchbarkeit dessen, was geliefert wurde.
- Ständiges Augenmerk auf Qualität.
- Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
- Das Team analysiert regelmäßig, wie es besser werden kann, und ändert und passt sein Verhalten entsprechend an.
- Agile Prozesse fördern nachhaltige Entwicklung. Das Projektteam sollte ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
- Agile Praktiken sollten unternehmensorientiert sein und die Anforderungen in Bezug auf IT-Governance, Unternehmensarchitektur und Interoperabilität berücksichtigen. Agile Teams sollten mit Teams und Stakeholdern, die alternative Ansätze verfolgen, wirksam zusammenarbeiten können.

3.2 Agilität und die PM²-Methodik

Die PM²-Methodik ist die offizielle Projektmanagement-Methodik der Europäischen Kommission. Sie soll Projektleitungen (PMs) befähigen, durch ein wirksames Management ihrer Projekte über den gesamten Lebenszyklus Lösungen und Nutzen für ihre Organisationen zu erbringen. Die PM²-Methodik wurde für die Anforderungen der Institute und Projekte der Europäischen Union entwickelt, ist aber auf Projekte jeder Organisation übertragbar. Es handelt sich um einen ausgewogenen Ansatz, der den Durchführungsteams ein autonomes Arbeiten ermöglicht und gleichzeitig die Konformität mit den Unternehmensstrukturen und betrieblichen Abläufen sicherstellt. Die PM²-Methodik ist in sich flexibel und schreibt keine spezifischen Prozesse auf technischer Ebene vor. Dieser Ansatz ermöglicht die schnelle Einführung agiler Praktiken. Zudem können agile Teams durch die Ausrichtung an der PM²-Methode den Einschränkungen der Governance auf höherer Ebene voll Rechnung tragen.

IT-Projekte müssen die Richtlinien, Entscheidungen und Kommunikationsstrukturen ihrer Organisation berücksichtigen. Weitere einzubeziehende Faktoren sind unter anderem:

- ein fester Haushaltszyklus,
- eine herkömmliche Managementkultur von Befehl und Kontrolle,
- ein herkömmlicher Wasserfall-Projektlebenszyklus,
- eine funktionale/schwache Matrix-Projektorganisation,
- ein klarer Meilenstein für die Projektgenehmigung,
- Planung und Architekturentscheidungen im Vorfeld.

Auf Grundlage dieser organisationsspezifischen Faktoren lässt sich ein belastbares Rahmenwerk für die Durchführung von IT-Projekten entwickeln, sofern sie von (lokal auf Organisations-/Abteilungsebene definierten) bewährten Verfahren und Aktivitäten begleitet werden. Klare Rollen und Zuständigkeiten ergänzen diese Faktoren ebenfalls und ermöglichen den Teammitgliedern, während des Lebenszyklus eines Projekts spezifische, genau definierte Aufgaben zu übernehmen.

Die PM²-Projektmanagement-Methodik umfasst und behandelt Lösungen für die vorgenannten Einschränkungen.

Die PM²-Methodik ist eine schlanke, einfach umzusetzende Methodik, die Projektteams an ihre spezifischen Anforderungen anpassen können.

Die PM²-Methodik bietet:

- eine Projekt-Governance-Struktur,
- Prozessrichtlinien,
- Vorlagen für Artefakte,
- Handlungsempfehlungen für die Verwendung der Artefakte,
- einen Katalog wirksamer Mindsets.

Bei der PM²-Methodik wird anerkannt, dass viele Arten von Projekten komplex und unsicher sind und eine agile Herangehensweise einen positiven Beitrag leisten kann, solche Projekte erfolgreich zu managen. Bestimmte Managementanforderungen lassen sich jedoch mit einem rein agilen Projektmanagementansatz nicht vollständig erfüllen.

Die PM²-Methodik ist eine umfassende Projektmethodik, die von einer Projekt-Governance-Struktur mit einem Lebenszyklus von aufeinanderfolgenden Phasen unterstützt wird. Sie deckt das gesamte Spektrum von Management und Projektdurchführung ab und ermöglicht gleichzeitig einen unternehmerorientierten agilen Ansatz. Die nahtlose Integration von der PM²-Methode und agilen Methoden hilft den Teams, richtlinien-, prozess- und auditkonform zu arbeiten. PM²-Agile vereint Auditanforderungen, die Zusammenarbeit mit Auftragnehmerinnen und Auftragnehmern, die Zusammenarbeit bei anderen Projekten und mit anderen Abteilungen und die Abstimmung mit den Gruppen auf Programm- und Portfolioebene unter einem Dach und befähigt so die Projektteams, die gewünschte „Agilität“ zu erreichen.

PM²-Agile bietet:

- Rollen und Zuständigkeiten (als Erweiterung der PM²-Governance),
- Einbindung in den gesamten PM²-Projektlebenszyklus,
- Kernthemen,
- eine Reihe wichtiger Events,
- eine Reihe empfohlener Agile-Artefakte (als Erweiterung der PM²-Artefakte),
- ein Glossar zu Agilität,
- agile Tools und Techniken.

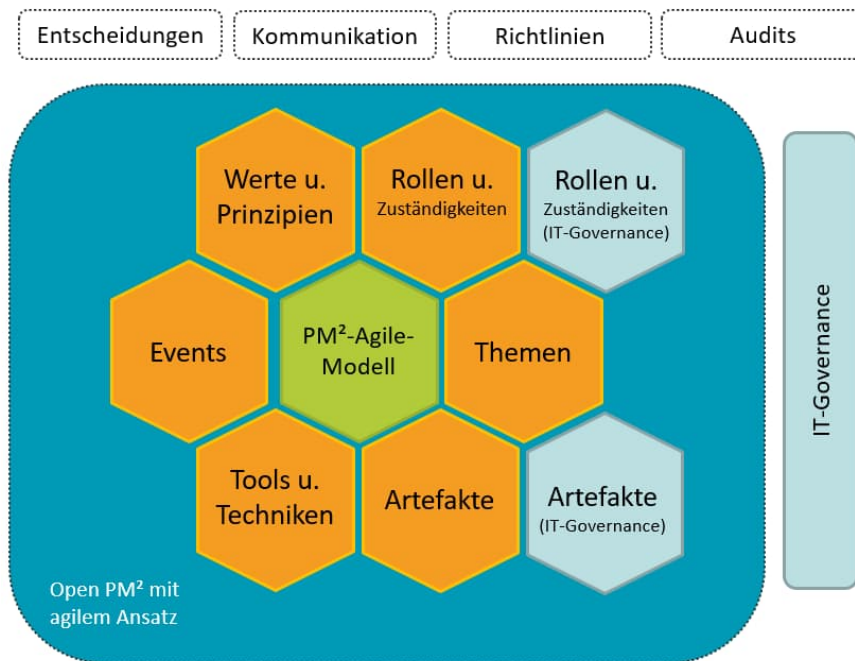


Abbildung 11: Elemente von PM²-Agile im Kontext von der PM²-Methodik

Bei Wahl eines agilen Projekts ermöglichen die inkrementellen und iterativen Entwicklungsansätze eine neue Durchführungsstrategie, die mithilfe der in der Abbildung dargestellten spezifischen Elemente von PM²-Agile umgesetzt wird.

Da es sich um ein PM²-Projekt handelt, sind alle diese spezifischen Elemente in den Rahmen der PM²-Methodik eingebettet, die das Ganze über das Governance-Modell (Rollen und Zuständigkeiten) und die Artefakte während des gesamten Projektlebenszyklus solide zusammenhält.

3.3 Das PM²-Agile-Modell

PM²-Agile ist keine neue agile Philosophie oder ein neuer agiler Ansatz. PM²-Agile passt einfach bewährte agile und „schlanke“ Praktiken an und verbindet sie mit der PM²-Methodik. PM²-Agile ermöglicht agilen Teams eine auf die PM²-Projektmanagementmethodik abgestimmte Arbeitsweise, bei der sie agile Werte und Prinzipien anwenden und Unternehmensorientierung und Compliance auf Ebene des Gesamtprojekts aufrechterhalten können. Die nachfolgende Abbildung zeigt, wie PM²-Agile mit Programmmanagement, Softwareentwicklung und den anderen Unternehmenspraktiken (z. B. Portfoliomanagement, Programmmanagement, IT-Betrieb usw.) verzahnt ist.

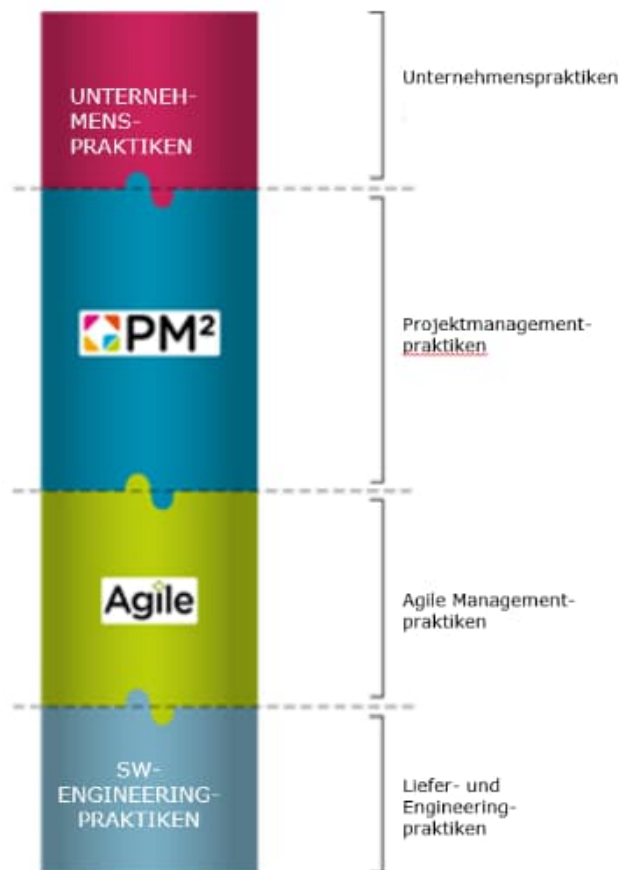


Abbildung 12: Einordnung von PM²-Agile in die Organisationsumgebung

Die PM²-Methode hat gegenüber anderen bekannten gängigen Projektmanagementmethoden den Vorteil, dass es „Agilität“ bewusst ermöglicht (d. h. die PM²-Prozesse und -Mindsets auf die agile Philosophie abgestimmt sind) und mit den übrigen Managementstrukturen und -prozessen einer Organisation verknüpft ist. PM²-Agile ist zwar keine Softwareentwicklungsbibliothek, enthält aber gleichwohl einige optionale Techniken wie testgetriebene Entwicklung, Pairing usw. Da PM²-Agile jedoch in erster Linie für die Softwareentwicklung eingesetzt wird, ist es in die Softwareentwicklungspraktiken gut integriert.

Agile fördert selbst organisierte Teams. Die PM²-Methode stellt den geeigneten Governance-Rahmen zur Verfügung, der die Governance-Anforderungen der gesamten Organisation widerspiegelt, den Projektkernteams (PCTs) aber auch ermöglicht, die Vorteile der Selbstorganisation zu nutzen. Zu diesem Zweck sieht PM²-Agile ein Projektkernteam (PCT) vor, dessen Mitglieder agil arbeiten.

Die nachfolgende Abbildung zeigt, wie sich PM²-Agile-Teams in die gesamte Projektorganisation einfügen. Für das agile Team wird eine Gruppe von Rollen und Zuständigkeiten festgelegt (siehe Abschnitt Rollen und Zuständigkeiten).

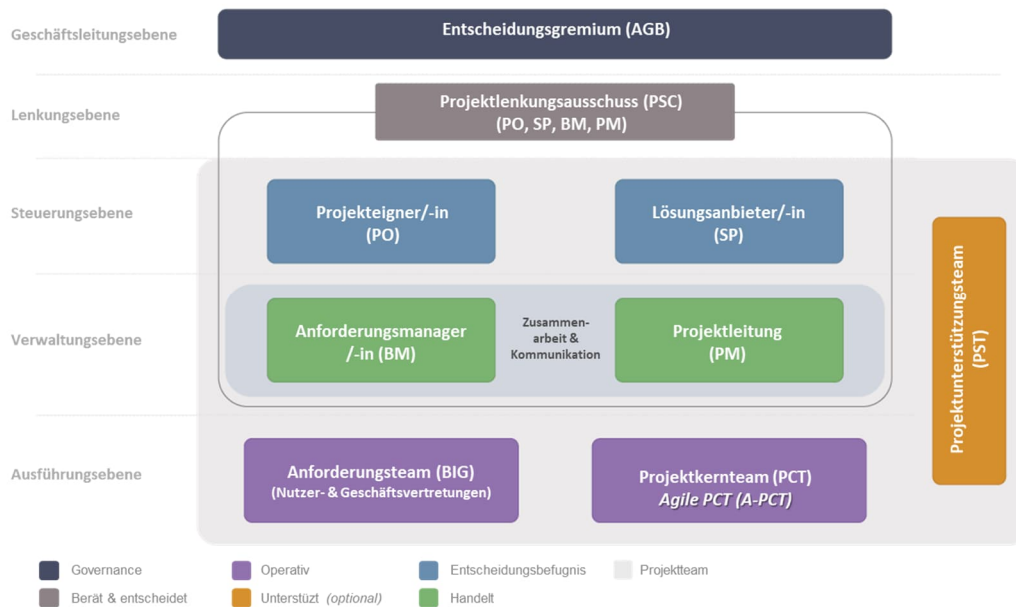


Abbildung 13: Agile Teams im Kontext der PM²-Projektorganisation

3.3.1 Lebenszyklus

Jedes Projekt hat einen Anfang und ein Ende. Der Lebenszyklus eines Projekts beginnt und endet zu identifizierbaren Zeitpunkten auf einer Zeitskala. Jede Projektphase entspricht einem Zeitraum während der Lebensdauer eines Projekts, in dem ähnliche Arten von Aktivitäten durchgeführt werden (z. B. haben planungsbezogene Aktivitäten gemessen am Aufwand ihr Hoch in der Planungsphase etc.). In der PM²-Methode durchlaufen Projekte vier unterschiedliche Phasen.



Abbildung 14: Projektphasen von PM²

Der in der Abbildung dargestellte Projektlebenszyklus umfasst alle Ereignisse und Aktivitäten ab dem Zeitpunkt, an dem die Idee/der Bedarf entstanden ist, bis zum endgültigen Abschluss des Projekts. Es ist zu beachten, dass die einzelnen Phasen selten klar voneinander abgegrenzt sind, sondern die Aktivitäten einer bestimmten Phase (z. B. Planungsaktivitäten) in der/den nachfolgenden Phase(n) (z. B. in der Durchführungsphase) noch weitergehen.

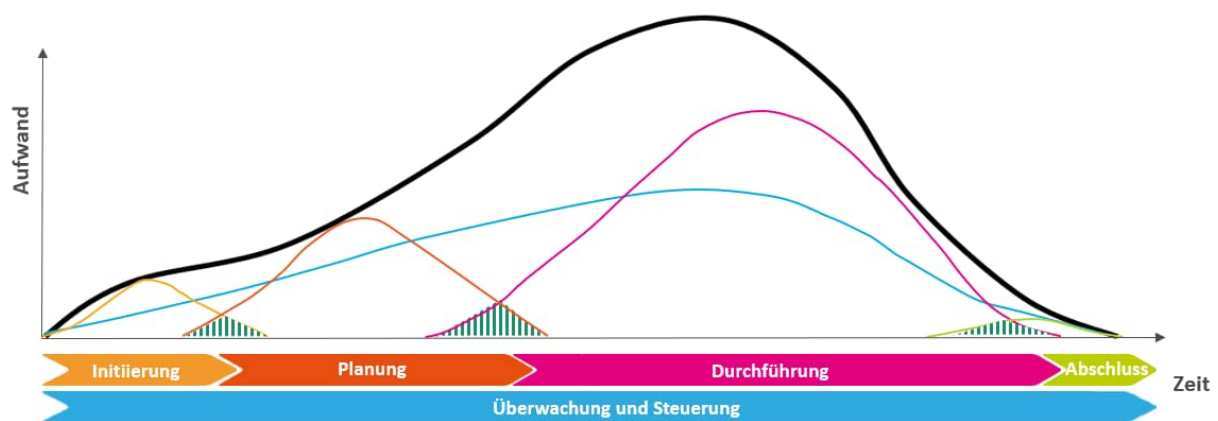


Abbildung 15: PM²-Projektlebenszyklus: Überschneidung phasenspezifischer Aktivitäten

Die Merkmale einiger Phasen (z. B. Planung) sind auf Ebene des agilen Teams möglicherweise nicht so greifbar wie auf der Ebene des Gesamtprojekts. Die Projektphasen existieren dennoch, und es sollte ein gemeinsames Verständnis sowohl zwischen den Mitgliedern des Projektteams als auch mit anderen Projektstakeholdern gefördert werden. Die sequenzielle Abfolge ergibt sich aus dem Konzept, dass das Projekt ein festgelegtes Anfangs- und Enddatum hat. Die Projektaktivitäten müssen daher mit der Initiierung und Planung beginnen, die zur Umsetzung führen und schließlich mit der Abnahme, der Einführung und dem Abschluss enden. Auf Teamebene werden die Aufgaben innerhalb von Iterationen jedoch nicht nacheinander ausgeführt: Planung, Analyse, Codeerstellung, Test und Review erfolgen in jeder Iteration. Einige agile Praktiken können auch ein eigenes Zeitfenster für Planung und Review vorsehen.

Bei PM²-Agile gibt es iterative Zyklen auf drei Ebenen: tägliche Zyklen, Iterationen und Releases. Unabhängig von ihrer Dauer folgen diese Zyklen dem sogenannten „CIR“-Rhythmus (Coordinate, Implement, Review (Koordinieren, Umsetzen, Überprüfen)) von PM²-Agile.

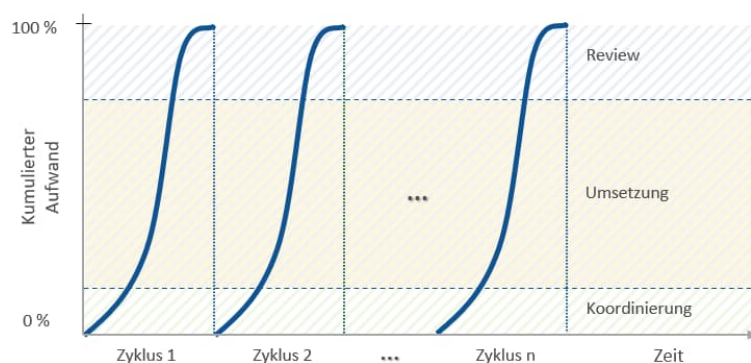


Abbildung 16: CIR-Rhythmus von PM²-Agile

Die Kombination der Phasen auf hoher Ebene mit den Iterationszyklen auf niedriger Ebene führt zu einer Aufwand-Zeit-Kurve mit im Zeitverlauf größeren Überschneidungen zwischen den Aktivitäten, die sich über mehrere Projektphasen erstrecken.

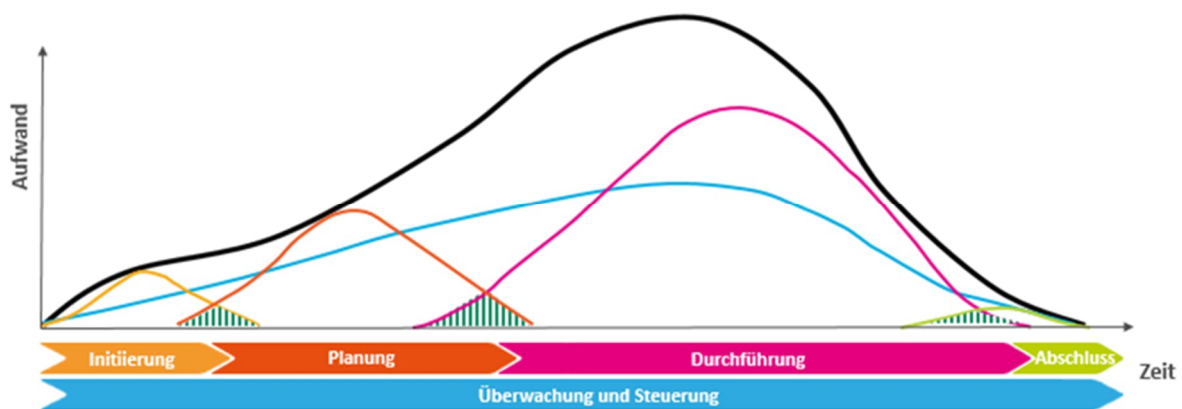


Abbildung 17: PM²-Projektlebenszyklus mit Iterationen der Aktivitäten: starke Überschneidung phasenspezifischer Aktivitäten

Am Anfang des Projekts geht es vor allem darum, zu verstehen, was entwickelt werden soll, indem eine Gesamtvision festgelegt wird und die Stakeholderinnen und Stakeholder und ihre Erfolgskriterien ermittelt werden. Da in dieser Phase noch vieles unsicher ist, erkennt PM²-Agile an, dass alle Annahmen, auf denen die Identifizierung der möglichen Systemfunktionen beruhen wird, ermittelt und formuliert werden müssen. Dadurch, dass Annahmen formuliert und nicht als Fakten behandelt werden, haben die Stakeholderinnen und Stakeholder und das Team die Möglichkeit, Fragen zu stellen, die Probleme zu erörtern und die beste Lösung zu finden. Ziel ist es, mindestens eine Lösung zu ermitteln und anschließend zu bewerten, ob sie auf Basis der Annahmen technisch durchführbar ist. Dies erfolgt durch Aufstellen mehrerer Hypothesen, die getestet werden müssen, um die formulierten Annahmen so schnell wie möglich zu validieren und die mit Unsicherheit verbundenen Risiken zu mindern.

Wichtig ist auch, die groben Schätzungen der Kosten, Zeitpläne und Risiken im Zusammenhang mit dem Entwicklungsaspekt des Projekts zu verstehen. Durch Einbeziehung dieser Schätzungen in die Schätzungen und Planungsartefakte auf Projektebene wird sichergestellt, dass sie berücksichtigt werden.

Mit Fortschreiten des Projekts verlagert sich der Schwerpunkt der fachspezifischen Aktivitäten zum inkrementellen Aufbau der Lösung. Während sich das Team daran macht, die Hypothesen zur Validierung der formulierten Annahmen zu prüfen, wächst das Verständnis, welche Aufgaben in welcher Weise zu erledigen sind. PM²-Agile unterstützt dies durch die Integration einer Nutzenanalyse für jedes Merkmal, die auf die Validierung einer Annahme abzielt.

Da sich die Entwicklung in verschiedenen Iterationen vollzieht, werden die Anforderungen fortlaufend überprüft und klargestellt. Sowohl technische als auch nicht technische Risiken werden ebenfalls regelmäßig behandelt. Dieser iterative und inkrementelle Ansatz von PM²-Agile führt dazu, dass nachhaltig und zügig Wert für die Organisation geschaffen wird. Dieser Wert wird mittels eines Bewertungs- und Lernrahmens genau gemessen. Das Rahmenwerk erfasst geeignete, relevante Daten aus einem MVP (oder nachfolgenden Versionen) und verwendet diese als (validierte) Lernmechanismen für die Entscheidungsfindung.

Zum Ende des Projekts hin verlagert sich der Schwerpunkt darauf, sicherzustellen, dass die Lösung gut in die Liefergegenstände des Gesamtprojekts integriert ist. Außerdem muss sie zur Realisierung des von der Organisation gewünschten Nutzens beitragen. Ein weiterer Schwerpunkt liegt darauf, sicherzustellen, dass die Organisation das Projekt „in Besitz“ nimmt und nach Projektabschluss für die Weiterentwicklung und Pflege des Informationssystems verantwortlich ist.

Die nachfolgende Abbildung enthält eine Übersicht über die wichtigsten Aktivitäten und Artefakte in den einzelnen Projektphasen. Eine ausführliche Darstellung der Agile-Artefakte findet sich in Abschnitt 7, „Artefakte“. Es ist zu beachten, dass Artefakte inkrementell und iterativ nach agilen Prinzipien geliefert werden.

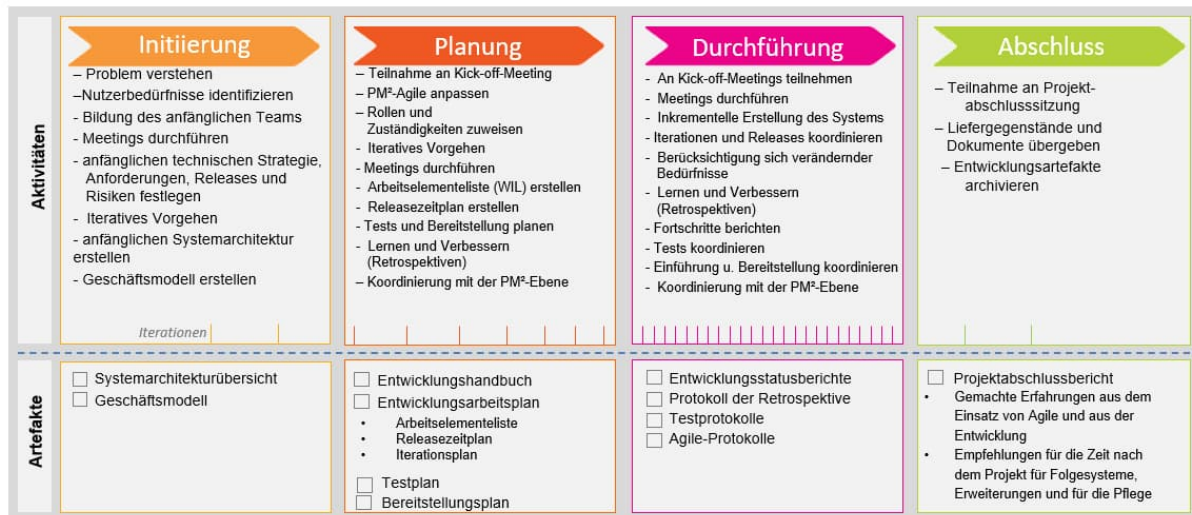


Abbildung 18: Übersicht über die Aktivitäten und Artefakte von PM²-Agile in den einzelnen Phasen

Unter dem Gesichtspunkt der Agilität können Projekte mehrere Releases haben, die sich aus dem inkrementellen Fortschritt in einer oder mehreren Iterationen ergeben. Zudem ist der Output jeder Iteration das Ergebnis des täglich erzielten inkrementellen Fortschritts. Dieser Ansatz mit mehreren Ebenen (siehe nachfolgende Abbildung) erstreckt sich über die verschiedenen in der PM²-Methode vorgesehenen Phasen, wobei auf allen drei Ebenen Wert geschaffen und Feedback gesammelt wird.

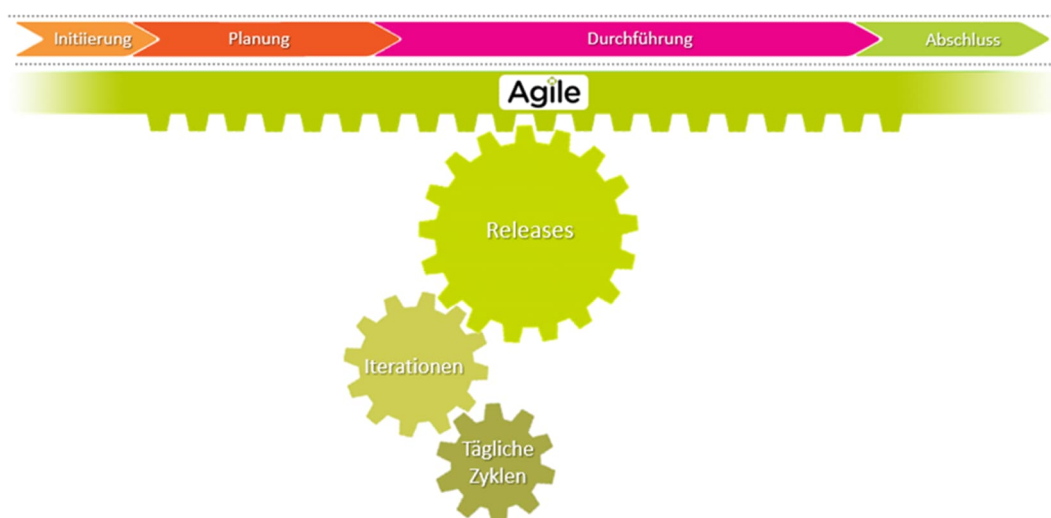


Abbildung 19: Von den Projektphasen zu den täglichen Zyklen

3.3.2 Iterationen

Eine agile Iteration ist eine Zeitspanne innerhalb eines Projekts, in der das agile Projektkernteam (A-PCT) einen stabilen, potenziell lieferbaren Teil der Lösung zusammen mit anderen notwendigen Begleitdokumenten oder Liefergegenständen erstellt. Jede Iteration hat eine Timebox, d. h. eine festgelegte Dauer. Der inhaltliche Umfang der Iteration wird aktiv gemanagt, damit er in diese Timebox passt.

Am Ende einer Iteration steht ein inkrementelles Produkt, mit dem das System dem Endprodukt einen Schritt näherkommt. Innerhalb jeder Iteration werden Prototypen oder bestimmte Teile des Systems entwickelt oder aktualisiert. Es ähnelt ein bisschen einer „wachsenden“ Software. Anstatt das System Stück für Stück nacheinander aufzubauen, wird das gesamte System in jedem Zyklus im CIR-Rhythmus weiterentwickelt.

Jede in eine Timebox gepackte Iteration wird sorgfältig geplant. Die Iterationsziele werden genau definiert, die Abnahmekriterien festgelegt und die Aufgaben und Zuständigkeiten der Beteiligten klargestellt. Außerdem wird durch vereinbarte Metriken und Bewertungsmethoden die Transparenz des Fortschritts sichergestellt.

Bei jeder Iteration muss Zeit eingeplant werden, um die Basis für die in der nachfolgenden Iteration durchzuführenden Arbeiten festzulegen. In diesem Zusammenhang spielen Entwicklungsblöcke (im Rahmen des Release Planning) eine wichtige Rolle, da sie es den Teams ermöglichen, die vom Product Owner (PrOw) und anderen geschäftlichen Stakeholderinnen und Stakeholdern erhaltenen Anforderungen kritisch zu hinterfragen. Innerhalb der Entwicklungsblöcke moderieren agile Teammitglieder (ATeM) eine Reihe von Workshops, die sich auf die Skizzierung und Entwicklung (einfacher) Low-Fidelity-Prototypen für künftige Merkmale konzentrieren, bis ein gemeinsames Verständnis erreicht ist. Anschließend werden (detaillierte) High-Fidelity-Prototypen erstellt und geliefert, die der Product Owner (PrOw) den geschäftlichen Stakeholderinnen und Stakeholdern, wie dem Anforderungsmanager oder der Anforderungsmanagerin (BM) oder dem Anforderungsteam (BIG), zur Validierung vorlegen kann. Wenn das agile Projektkernteam (A-PCT) überzeugt ist und sich darauf geeinigt hat, welche Elemente zuerst zu liefern sind, kann die Entwicklung beginnen.

Bei jeder Iteration sollten die Arbeitselemente mit der höchsten Priorität implementiert und die kritischsten Risiken behandelt werden. Dies gewährleistet, dass jede Iteration den maximalen Mehrwert erzeugt und gleichzeitig Unsicherheiten verringert. Inkrementelle und iterative Entwicklung werden üblicherweise mit kontinuierlicher Integration verbunden. Komponenten, die einen Komponententest durchlaufen haben, werden in einen Build integriert, der anschließend zum Integrationstest übergeben wird.

Die Fähigkeiten der integrierten Software wachsen also mit jeder weiteren Iteration in Richtung Integrationsziele. Durch regelmäßige Builds (beispielsweise täglich oder häufiger) ist es möglich, bei Integrationen und Tests aufgetretene Probleme zu zerlegen und über den Entwicklungszyklus zu verteilen. Große Projekte scheitern oft deshalb, weil alle Probleme bei einer einzigen konzentrierten Integration, die sehr spät im Projektzyklus stattfindet, gleichzeitig aufgedeckt werden. Je später man sich im Projektzyklus befindet, desto höher ist das Risiko, dass ein einziges Problem den Fortschritt des gesamten Teams, das keinen agilen Ansatz verfolgt, zunichtemacht.

Die beste und wirksamste Methode, mit der ein agiles Projektkernteam (A-PCT) greifbare Fortschritte nachweisen kann, besteht darin, den Output jedes Zyklus verfügbar zu machen. So können die Teams frühzeitig Feedback erhalten, auf dessen Basis sie immer besser verstehen, welche Aufgaben in welcher Weise zu erledigen sind.

PM²-Agile sieht drei Möglichkeiten vor, wie Arbeitsergebnisse verfügbar gemacht werden können. Die Methoden haben ähnliche Ziele (die Möglichkeit, Fortschritte nachzuweisen, Mehrwert zu liefern und Feedback zur Verringerung der Unsicherheit einzuholen), unterscheiden sich jedoch leicht in ihrer Abfolge und im notwendigen Grad der Formalisierung.

- **Verfügbarmachung von Arbeitselementen** – Dies erfolgt, wenn das Team ein oder mehrere Arbeitselemente aus der Arbeitselementenliste (WIL) fertigstellt.
- **Verfügbarmachung am Ende einer Iteration** – Dies erfolgt am Ende der Iteration und unterstützt den Iterationsreview. Umfasst alle Arbeitselemente, die bei der betreffenden Iteration als bereit für den Review erachtet werden.
- **Release** – Hierbei handelt es sich um ein vereinbartes Lösungsincrement, das an die Auftraggeberin oder den Auftraggeber und die Nutzenden geliefert wird, damit sie es als reales Produkt verwenden können. Es umfasst alle abgeschlossenen Arbeitselemente aus mehreren vorherigen Iterationen.

In den folgenden Abschnitten werden diese einzelnen Ansätze näher ausgeführt.

Verfügbarmachen von Arbeitselementen

Die wichtigste treibende Kraft für ein agiles Projektteam (A-PCT) ist die Arbeitselementenliste, die von dem Product Owner (PrOw) nach Priorität geordnet wird. Wenn das Team ein Element der Liste fertigstellt, sind an die Auftraggeberin oder den Auftraggeber bereits greifbare Fortschritte zu liefern und ist wertvolles Feedback einzuholen. Diese Abfolge sollte während der Iteration beibehalten werden, bis das Team die kumulierten Arbeitsergebnisse beim Iterationsreview präsentiert (siehe nachfolgende Abbildung).

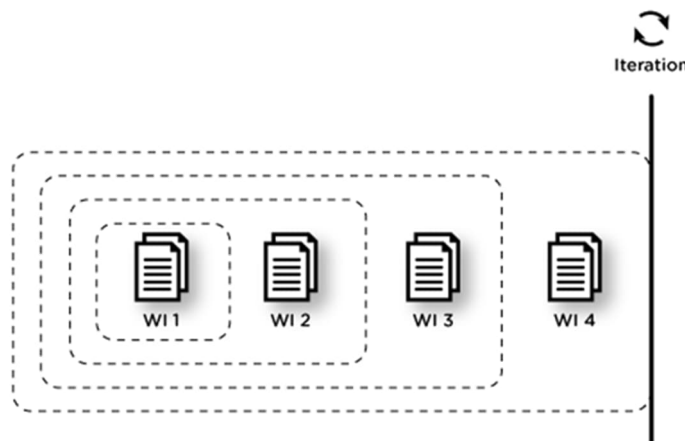


Abbildung 20: Verfügbarmachen von Arbeitselementen

Die Freigabe eines Arbeitselements sollte ein recht einfacher und unkomplizierter Vorgang sein. Eine einfache Bereitstellung in einer Standard-Testumgebung ist oft ausreichend.

Nachdem das Arbeitselement verfügbar gemacht wurde, benachrichtigen die an der Implementierung beteiligten agilen Teammitglieder (ATeM) den Product Owner (PrOw) und bitten um Feedback.

Da dies ein wiederkehrender Prozess ist, sollte er möglichst schlank und mit minimalem formalem Aufwand gestaltet werden, da das Ziel darin besteht, schnell Feedback als Vorbereitung auf den Iterationsreview einzuholen. Am Ende ist es die „Summe“ aller verfügbar gemachten Arbeitselemente, die über den Erfolg eines Iterationsreviews entscheidet.

Nachweis der Iterationsergebnisse

Zu Beginn jeder Iteration verpflichtet sich das agile Projektkernteam (A-PCT) zu einem mit dem Product Owner (PrOw) vereinbarten Ziel. An diesem Ziel orientiert sich das Team bei der Implementierung einer bestimmten Menge von Arbeitselementen aus der Arbeitselementeliste (WIL). Bis zum Ende der Iteration weist das Team alle implementierten Arbeitselemente nach, um zu validieren, ob das Iterationsziel erreicht wurde.

Wie im vorherigen Abschnitt dargelegt, trägt jedes verfügbar gemachte Arbeitselement dazu bei, dass ein Artefakt „wächst“ und sich weiterentwickelt. Dieser Prozess setzt sich in mehreren Iterationen fort, bis eine Gruppe von Funktionen gemäß der von dem Product Owner (PrOw) festgelegten Strategie freigegeben wird (siehe nachfolgende Abbildung).

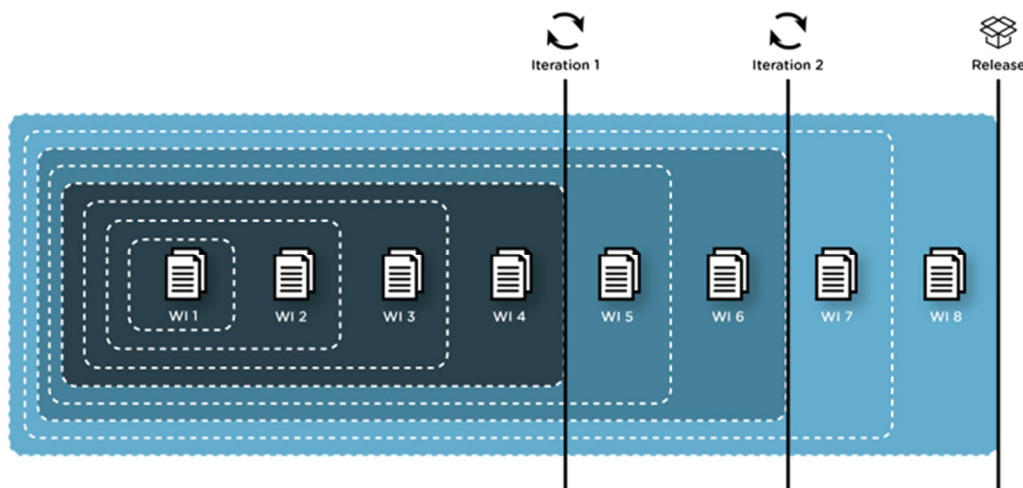


Abbildung 21: Lösung „wächst“ zu einem Release

Durch die Verfügbarmachung von Arbeitselementen erhält das Team schnelles Feedback und bleibt auf Kurs; jedoch erst am Ende der Iteration wird beurteilt und verifiziert, ob das Team alle Ziele, zu denen es sich verpflichtet hat, erreichen konnte. Diese Beurteilung erfolgt im Iterationsreview.

Der Iterationsreview ist ein vorgeschriebenes Ereignis, das dazu dient, dass ein Team Feedback von der Auftraggeberin oder dem Auftraggeber einholt, um sicherzustellen, dass die Arbeitselemente erledigt (im Sinne ihrer „Definition of Done“) sind. Dieses Ereignis erfordert ein bestimmtes Maß an Formalismus. Auch wenn ein großer Teil des Iterationsinhalts bereits verfügbar gemacht und mit einer gewissen Zuverlässigkeit validiert wurde, ist jedes Arbeitselement einer formalen Prüfung zu unterziehen.

Im Laufe des Iterationsreviews wird der Nutzen aus der fortlaufenden Verfügbarmachung von Arbeitselementen erkennbar. Dies liegt daran, dass die agilen Teammitglieder (ATeM) die entwickelten Arbeitselemente in vollem Umfang nachweisen und gleichzeitig den übrigen Stakeholderinnen und Stakeholdern helfen können, sie zu validieren. Wenn den Stakeholderinnen und Stakeholdern die Möglichkeit zu einer strengen, klaren Validierung gegeben wird, wirkt sich dies positiv auf die Gesamtwahrnehmung der Lösung aus.

Der Product Owner (PrOw) gibt das primäre Feedback zu den während der Iteration verfügbar gemachten Arbeitselementen. Beim Iterationsreview sollten jedoch auch die Anforderungsmanagerin oder der Anforderungsmanager (BM), die Mitglieder des Anforderungsteams (BIG) und andere relevante Stakeholderinnen und Stakeholder in die Beurteilung einbezogen werden.

3.3.3 Releases

Ein Release ist das Ergebnis eines inkrementellen Prozesses über mehrere Iterationen, bei dem das Team die Lösung als marktfähiges Produkt (oder als Minimum Viable Product), das von den Stakeholderinnen und Stakeholdern verwendet werden kann, freigibt.

Je nach entwickelter Lösung sind unter Umständen weitere formale Ebenen und Validierungen erforderlich. Normalerweise ermöglicht der inkrementelle Prozess auf Grundlage einer fortlaufenden Verfügbarmachung von Arbeitselementen und nachgewiesener, validierter Arbeitsergebnisse einen soliden Release. Dennoch können spezifische Anforderungen zusätzliche Validierungsschritte notwendig machen, um sicherzustellen, dass das Release die vereinbarte „Definition of Done“ wirklich vollständig erfüllt.

Bei der Entwicklung komplexerer Lösungen erfordert der Release-Prozess möglicherweise zusätzliche Vorbereitung und einige zusätzliche Aktivitäten (Nutzerschulungen, Behebung kritischer Fehler etc.). Dieser zusätzliche Aufwand kann als Teil einer sogenannten Einführungsiteration geplant werden. Der Hauptzweck besteht darin, eine erfolgreiche Bereitstellung der Lösung in der Produktionsumgebung sicherzustellen und so die künftige Verfügbarmachung für die Nutzergemeinschaft zu ermöglichen.

Bei einer Einführungsiteration konzentriert sich das Team im Wesentlichen darauf, für eine reibungslose Einführung der Lösung zu sorgen. Das Team kann beispielsweise entscheiden, dass keine neuen Funktionen weiterentwickelt und stattdessen Aktivitäten durchgeführt werden, die dem Produkt ebenfalls einen Mehrwert hinzufügen. Es ist auch möglich, ein agiles Team in zwei parallel arbeitende Unterteams aufzuteilen. Das eine Team kann beispielsweise an den Release-Aktivitäten arbeiten, während das andere neue Funktionen für ein Nachfolge-Release entwickelt.

Typische Aktivitäten im Rahmen einer Einführungsiteration sind unter anderem:

- Behebung kritischer oder schwerwiegender Fehler
- Dokumentation verschiedener Aspekte des Systems (z. B. Support- und Benutzerdokumentation)
- Durchführung zusätzlicher Funktions-, Leistungs-, Belastungs-, System- und Integrationstests
- Schulung von Supportpersonal und Endnutzenden
- Aktualisierung des Entwicklungsplans und Bereitstellungsplans

Eine einzige Einführungstransition reicht möglicherweise nicht aus, um alle notwendigen Aktivitäten durchzuführen. Eine Einführungsiteration umfasst einige andere Aktivitäten, wird aber nicht anders geplant. Sie ist und bleibt eine Iteration und ist daher wie jede „normale“ Iteration (mit geeigneten Events, Iterationsplanung mit einer Arbeitselementeliste etc.) durchzuführen.

3.4 Mindsets von PM²-Agile

Die PM²-Methode fördert eine Reihe von Mindsets, deren Ziel es ist, die vier Säulen (Governance, Lebenszyklus, Prozesse und Artefakte) zu integrieren und zu ergänzen. Diese Mindsets sind hilfreiche Gedächtnisstützen für wirksame Einstellungen und Verhaltensweisen. Sie helfen den Projektteams, zu definieren, was für den Projekterfolg (wirklich) wichtig ist, und sich darauf zu konzentrieren, und unterstützen sie bei der (Neu-)Einordnung der Projektziele in den breiteren Kontext der Organisation.

Die Mindsets sollen Projektteams helfen, die mit dem Projektmanagement verbundenen Komplexitäten zu bewältigen. Sie stellen einen Katalog gemeinsamer Überzeugungen und Werte für alle Projektteams zur Verfügung.

Die elf PM²-Mindsets sind nachfolgend in einer Liste mit elf Aussagen dargestellt. Sie gelten für Projektleitungen und Projektteams, die PM² in der Praxis anwenden:

1. Wende PM²-Best Practices beim Management von Projekten an.
2. Denke daran, dass die Methoden dem Projekt dienen sollen, nicht umgekehrt.
3. Handle bei allen Projekt- und Projektmanagementaktivitäten stets ergebnisorientiert.
4. Sei bestrebt, Projektergebnisse mit maximalem Mehrwert zu erzielen, anstatt einfach nur Pläne zu befolgen.
5. Fördere eine Projektkultur der klaren Kommunikation und wirksamen Zusammenarbeit.
6. Weise die Projektrollen den Personen zu, die für das Projekt am besten geeignet sind.
7. Stelle möglichst produktiv ein Gleichgewicht zwischen den „P“s des Projektmanagements her: Produkt (product), Prozess (process), Plan (plan), Personen (persons), Pleasure/Pain (Freude und Frust) and Perception (Wahrnehmung).
8. Investiere in die Entwicklung Deiner fachlichen und verhaltensbezogenen Kompetenzen, um besser zum Projekt beitragen zu können.
9. Beziehe Stakeholderinnen und Stakeholder in die notwendigen organisatorischen Änderungen mit ein, um den Projektnutzen zu maximieren.
10. Gib Wissen weiter, manage aktiv die gemachten Erfahrungen und trage zur Verbesserung des Projektmanagements innerhalb der Organisation bei.
11. Orientiere dich an den PM²-Handlungsempfehlungen für Ethik und Berufsethik.

Um die PM²-Mindsets zu verinnerlichen, sollten sich Projektleitungen (PMs) und Projektteams, die die PM²-Methodik in der Praxis anwenden, die folgenden wichtigen selten gestellten Fragen (IAQs) stellen:

- Wissen wir, was wir tun?
- Wissen wir, warum wir es tun? Interessiert das überhaupt jemanden?
- Sind die richtigen Personen beteiligt?
- Wissen wir, wer was tut?
- Lieferung um jeden Preis und mit jedem Risiko?
- Ist dies wichtig?
- Ist dies eine Aufgabe für „sie“ oder für „uns“?
- Sollte ich einbezogen werden?
- Haben wir uns verbessert?
- Gibt es ein Leben nach dem Projekt?

Die PM²-Mindsets entsprechen übrigens ganz den Prinzipien der Agilität, wie sie in PM²-Agile und in diesem Leitfaden dargelegt sind. Darüber hinaus werden die PM²-Mindsets durch die nachstehenden PM²-Agile-Mindsets ergänzt.

Mindsets von PM²-Agile

Die wichtigsten Mindsets eines agilen Durchführungsansatzes ergeben sich aus einer Interpretation des Manifests für agile Softwareentwicklung und können mit drei Punkten zusammengefasst werden:

- Fokus auf die Bereitstellung der Lösung (während im ursprünglichen Manifest der Begriff „Softwareentwicklung“ verwendet wurde, der von sehr vielen als reine Konzentration auf Softwareentwicklung oder -konstruktion verstanden wird).
- Fokus auf alle Stakeholderinnen und Stakeholder (während im ursprünglichen Manifest der Begriff „Kundinnen“ und „Kunden“ verwendet wurde, den sehr viele nur mit „zahlenden“ geschäftlichen Stakeholderinnen und Stakeholdern zu verbinden scheinen).
- Fokus auf das gesamte Ökosystem der Organisation und seine Verbesserung (während der Fokus im ursprünglichen Manifest auf den Entwicklungsteams lag).

Treiber

- Es gibt kein agiles Projekt. Es gibt nur PM²-Agile Projekte.
- Agile Teammitglieder sind Teil des PM²-Projektkernteams (PCT).
- Agile Rollen sind nicht gleichbedeutend mit Personen. Die Rollen werden von Personen erfüllt.
- Es gibt Auftraggeberinnen und Auftraggeber und Nutzende.
- Nimm eine nutzerzentrierte Haltung ein.
- Das Tun kommt vor der Analyse.
- Gehe seriell im Großen und iterativ im Kleinen vor.
- Praktiziere Agilität, aber berücksichtige auch die „Organisation“.
- Organisiere die Arbeit rund um Timeboxen, nicht anderes herum.
- Löse Dich von der „Bereitstellung von Liefergegenständen“ und konzentriere Dich auf die „Gestaltung der Nutzererfahrung“.
- Sei im Rahmen der Einschränkungen der Organisation so agil wie nötig und so agil wie möglich.
- „Feiere Misserfolge“ als Möglichkeit, offen über Fehler zu sprechen und den Lernprozess zu beschleunigen.

Laufende Ziele

- Erfüllung des Projektauftrags
- Weiterentwicklung der Fähigkeiten der Teammitglieder
- Verbesserung der bestehenden Infrastruktur
- Verbesserung der Teamprozesse und -umgebung
- Nutzung der bestehenden Infrastruktur
- Bewältigung von Risiken

- Konzentration auf den Wertestrom, in dem die Auftraggeberin oder der Auftraggeber den Wert definiert
- Konzentration auf fortlaufende Minimierung von Verschwendung, die den Fluss behindert
- Transparenz, die es den Teammitgliedern ermöglicht, die beiden vorherigen Ziele kontinuierlich zu verbessern

Rechte jedes Einzelnen

- Recht auf respektvolle Behandlung
- Recht auf geeignete Arbeitsbedingungen/-umgebung zur Ausführung der Aufgabe:
 - Unterrichtung über den geschäftlichen Kontext
 - zeitnahe Entscheidungen
 - Erhalt der identifizierten notwendigen Ressourcen
 - Einhaltung von Zusagen sowohl innerhalb des Teams als auch auf Ebene des Gesamtprojekts und der Organisation
- Recht auf aktive Beteiligung an den Planungs- und Schätzungsaktivitäten, Verpflichtung zur Lieferung gemäß den Qualitäts- und Organisationsstandards
- Recht auf eine Umgebung des kontinuierlichen Lernens, in der die Entwicklung von Fähigkeiten als nutzbringend für die Person, das Team und die gesamte Organisation angesehen wird. Fehler werden als Chancen für Lernen und Verbesserung betrachtet.

3.5 Artefakte und Dokumentation bei PM²-Agile

Agilität bedeutet nicht, dass es „keine Dokumentation“ gibt, sondern dass die Dokumentation auf das Nötigste beschränkt wird. Das „Nötigste“ bemisst sich sowohl nach den Anforderungen des Projekts als auch den Anforderungen des Unternehmens in Bezug auf Transparenz, Rückverfolgbarkeit und Kontrolle.

Die Dokumentation, der in der agilen Ebene geplanten und ausgeführten Arbeiten trägt entscheidend dazu bei, die Transparenz und Koordination zwischen den verschiedenen Ebenen der PM²-Projektorganisation (z. B. zwischen den Steuerungs-, Management- und Ausführungsebenen) zu erhöhen.

Relevanz und Nutzen der Projektdokumentation (als Informations- und Wissensquelle) nehmen außerhalb der Projektgrenzen für Entscheidungen auf projektübergreifender Ebene und auf der Multiprojekt- (bzw. „Programm- und Portfolio“-)ebene immer mehr zu. Agile Teams müssen daher eine Dokumentation erstellen, die unternehmensorientiert (im Hinblick auf Inhalt, Struktur und Sprache) und gleichzeitig innerhalb der Organisation standardisiert ist und den PM²-Dokumentationsanforderungen entspricht.

Eine Reihe von Artefakten unterstützt die Anwendung von PM²-Agile. Diese Artefakte dienen zur Erfassung und Dokumentation von Informationen über den Managementansatz, spezifische (Entwicklungs-) Aktivitäten, Meilensteine, Probleme und Fortschritte.

Die Artefakte gliedern sich in drei Kategorien: IT-Governance-Artefakte (die unabhängig vom gewählten Managementansatz allen IT-Projekten gemeinsam sind), (entwicklungsspezifische) Agile-Artefakte und Koordinierungs- und Berichtsartefakte.

Art des Artefakts	Beschreibung
IT-Governance-Artefakte	Enthalten die von der IT-Governance der Organisation geforderten Angaben. Sie dokumentieren den Business Case, den Projektauftrag, die Systemarchitekturübersicht und das Geschäftsmodell.
(Entwicklungsspezifische) Agile-Artefakte	Erfassen Informationen in Bezug auf die Planung (entwicklungs-) spezifischer Prozesse, Aktivitäten, Releases, Iterationen und anderer Meilensteine.
Koordinierungs- und Berichtsartefakte	Erfassen die notwendigen Informationen, um die Aktivitäten des Gesamtprojekts mit denen des agilen Projektkernteams (A-PCT) zu koordinieren und sicherzustellen, dass die Projektleitung einen Überblick über die bereichsspezifischen Aktivitäten, Probleme, Meilensteine und Fortschritte hat.

Die folgenden Artefakte unterstützen die Projektarbeit auf Grundlage von PM²-Agile:

- Business Case und Projektauftrag
- Systemarchitekturübersicht
- Geschäftsmodell
- Entwicklungshandbuch (wird Bestandteil des gesamten Projekthandbuchs)
- Entwicklungsarbeitsplan (wird Bestandteil des gesamten Projekthandbuchs)
- PM²-Agile-Protokolle
- Bereitstellungsplan
- Testpläne
- Entwicklungsstatusbericht (wird Bestandteil der Projektberichte)

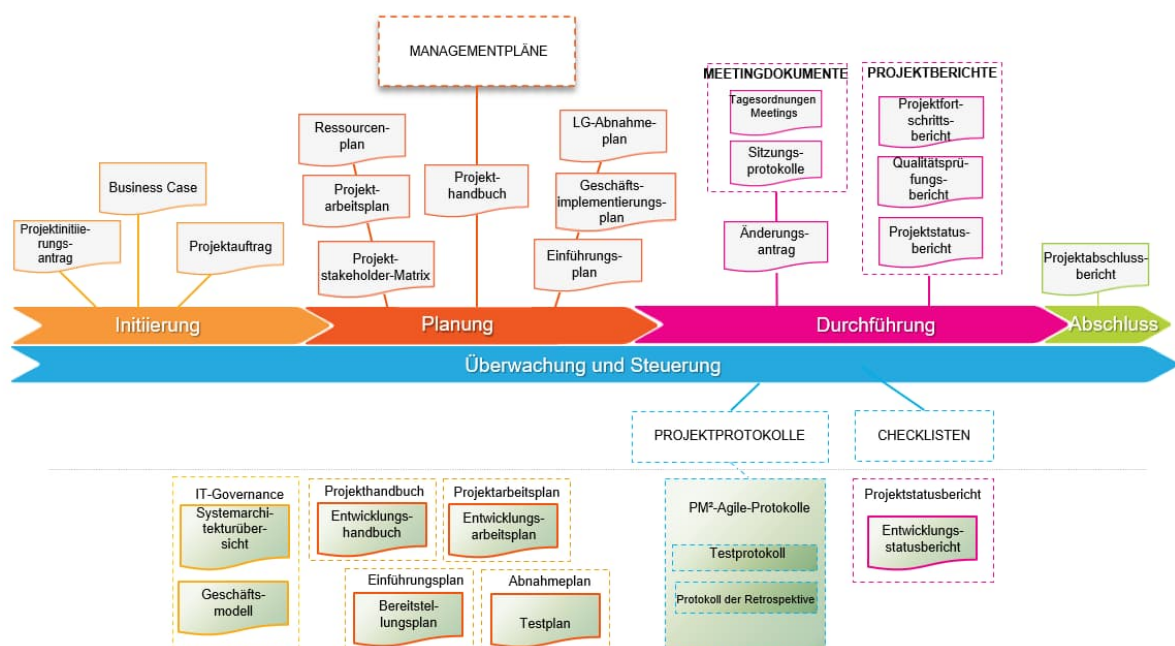


Abbildung 22: (Kombinierte) Artefaktlandschaft von PM²-Agile

Es ist zu beachten, dass laut IT-Governance (siehe Abschnitt Artefakte) für Projekte, die Informationssysteme als einen Teil der Lösung beinhalten, die folgenden Dokumente obligatorisch sind:

- Business Case
- Projektauftrag
- Systemarchitekturübersicht
- Geschäftsmodell

Je nach Organisationsstruktur kann die IT-Governance zusätzlich zu den vorgenannten Artefakten noch weitere Artefakte verlangen.

3.6 Tailoring und Anpassen

PM²-Agile sollte nicht als eine für alle geeignete Einheitslösung gesehen werden. Vielmehr sollte der Ansatz individuell angepasst oder „maßgeschneidert“ werden, um die jeweiligen geschäftlichen und projektbezogenen Bedingungen und Einschränkungen zu berücksichtigen, und sich weiterentwickeln können, wenn die Projektteams bei jeder Iteration, jedem Release oder Projekt dazulernen und neue Erkenntnisse (über die Umgebung und die Lösung) gewinnen.

„Tailoring“ bedeutet, bestimmte Teile der Methodik, wie Prozessschritte, den Inhalt von Artefakten oder die Aufteilung der Zuständigkeiten zwischen den verschiedenen Rollen, zu ändern. Organisationen passen die Methodik so an die besonderen Anforderungen ihrer Struktur und Kultur an und stimmen sie auf ihre Prozesse, Richtlinien etc. ab. Ein solches Tailoring kann auch auf anderen Ebenen, beispielsweise in einem Bereich, einer Abteilung oder sogar in einem kleineren Sektor, erfolgen und auf alle zugehörigen Projekte angewandt werden.

Im Entwicklungshandbuch (siehe Abschnitt Artefakte) wird dokumentiert und dargelegt, wie und warum PM²-Agile für ein bestimmtes Projekt angepasst wurde, damit es den Anforderungen und Zielen des Projekts besser gerecht wird. Dies umfasst normalerweise folgende Angaben:

- Begründung für die Anpassungen und Abweichungen,
- Informationen über Abweichungen von den Standardverfahren und die Gründe hierfür (normalerweise durch Angabe der hinzugefügten oder weggelassenen Artefakte oder Tools und Techniken),
- Anpassung der Rollen und Zuständigkeiten,
- verwendetes Tool oder Medium und Format für die einzelnen Artefakte,
- Verweise auf andere Handlungsempfehlungen und Informationen, die das Projekt gegebenenfalls zusätzlich zu PM²-Agile und der PM²-Methode heranzieht,
- welche (entwicklungsspezifischen) Reviews mit welchem Grad an Formalität durchgeführt werden,
- Verfahren und Artefakte für die Berichterstattung über Fortschritte, die Durchführung von Messungen/Bewertungen, das Management von Anforderungen und Änderungsanträgen etc.,
- Konfiguration spezifischer Assets für die Strategieentwicklung,
- Zuständigkeiten für Review und Genehmigung (mittels einer RASCI-Tabelle).

Die folgende Liste enthält typische Aktivitäten bzw. wichtige Faktoren, die beim Tailoring zu berücksichtigen sind:

- Einbeziehung der richtigen Stakeholderinnen und Stakeholder,
- Identifizierung der Richtlinien und Verfahren, die innerhalb der Organisation zwingend einzuhalten sind, und Sensibilisierung aller agilen Teammitglieder hierfür,
- Entscheidung, welche projektspezifischen Inhalte notwendig sind,
- Erfassung/Aktualisierung des spezifischen Begriffsverzeichnisses (Definitionen, Akronyme und Abkürzungen) für die jeweilige Aufgabe (projektweites Glossar oder bereichsspezifischer Prozess),
- Erfassung von Referenzen (wichtigen Dokumenten für die jeweilige Aufgabe),
- Entscheidung, wie und in welchem Umfang die Arbeit dokumentiert wird,
- Sicherstellung, dass die Projektorganisation, die Zuständigkeiten und Schnittstellen auf Projekt- und Agile-Ebene dargelegt werden und allen klar sind,
- Entscheidung, mit welchen Tools die Arbeitsergebnisse erstellt und gepflegt werden,
- Sicherstellung, dass die richtigen Tools und die richtige Infrastruktur zur Verfügung stehen und das Team weiß, wie sie zu nutzen sind,
- Kenntnis der verfügbaren Metriken und Berichte und Entscheidung, welche wie häufig verwendet werden,
- Feststellung, dass das Team über die benötigten Schulungen und Ressourcen, einschließlich Personal, verfügt,
- Überprüfung und Vereinbarung des Tailorings.

4. Themen von PM²-Agile

Mit der Einführung von PM²-Agile müssen einige Projektmanagementkonzepte und -praktiken angepasst werden, weil ihnen jetzt ein anderer Katalog von Werten und Prinzipien zugrunde liegt. PM²-Agile trägt dem Rechnung und unterteilt die Konzepte und Praktiken in mehrere Wissensbereiche, sogenannte „Themen“. In diesem Abschnitt sollen diese Themen vorgestellt werden, und es wird erläutert, wie sich die Konzepte und Praktiken weiterentwickelt haben, um diese Realität zu unterstützen.

4.1 Lean UX

Das in Abschnitt 2.3, „Kontinuierliche Verbesserung und validiertes Lernen“, beschriebene Lean Start-up-Modell ist ein wesentlicher Befähiger für die Aktivitäten von kontinuierlicher Verbesserung und validiertem Lernen. Lean UX, ein weiterer wesentlicher Baustein von PM²-Agile, ist davon stark beeinflusst.

Das Lean Start-up-Modell unterstützt den Lean UX-Prozess in PM²-Agile durch:

- Beseitigung von Verschwendung bei der Entwicklung durch User Experience (UX) Design,
- Harmonisierung der Beziehungen zwischen dem agilen Projektkernteam (A-PCT) und den übrigen Stakeholderinnen und Stakeholdern,
- Mindset-Wechsel von der Annahme zum Experimentieren.

Lean UX hilft den beteiligten Stakeholderinnen und Stakeholdern, den Prozess nachzuvollziehen und zu verstehen, wie wichtig es ist, alle in jeder PM²-Phase einzubinden. Es basiert auf dem Grundsatz, dass Anforderungen Annahmen sind, die das agile Projektkernteam (A-PCT) vor Beginn der Entwicklung validieren muss. Entwicklung ist mit Zeit und Aufwand verbunden, um zu einem ausreichend marktfähigen Produkt zu gelangen, aus dem man lernen kann und das einen geschäftlichen Mehrwert liefert. Schnelle Prototypen sichern daher geeignete Lernerfahrungen, die das Team weiterbringen. Der Grundgedanke lautet, dass Prototypen ohne Code ein gemeinsames Verständnis, kontinuierliche Verbesserung und validiertes Lernen unter den Teammitgliedern und Stakeholderinnen und Stakeholdern fördern.

Design Blocks bilden die Basis, um die Annahmen kritisch zu hinterfragen und zu validieren, und unterstützen die Entwicklung von MVPs.

Lean UX gewährleistet, dass alle in Abschnitt 2.3, „Kontinuierliche Verbesserung und validiertes Lernen“, beschriebenen wesentlichen Bausteine bei der Erstellung eines MVP berücksichtigt werden.

Die vier Schritte des Lean UX-Prozesses wurden angepasst und nahtlos in den PM²-Agile-Lebenszyklus integriert, wobei ein besonderer Schwerpunkt auf die Initiierungs- und Durchführungsphasen gelegt wurde. Diese Integration wird in den nachstehenden Abschnitten näher ausgeführt:

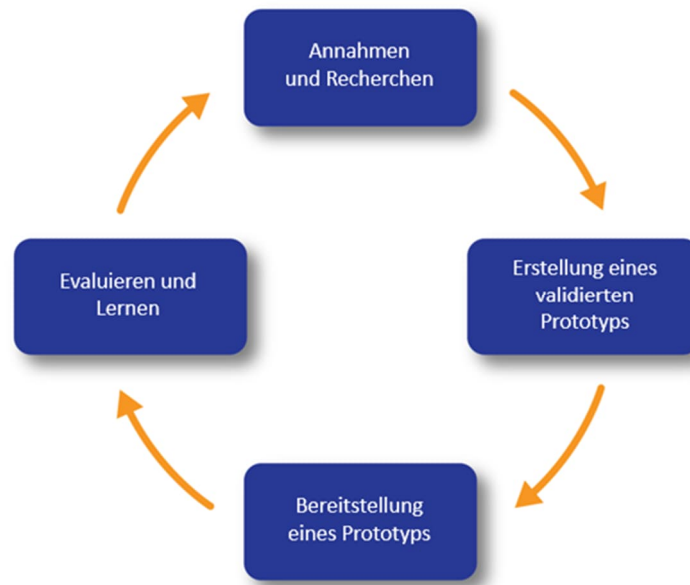


Abbildung 23: Lean UX-Prozess von PM²-Agile

Annahmen festlegen und Recherche

Dieser Schritt spielt sowohl in der Initiierungsphase von der PM²-Methode (als Grundlage für die Definition der Anforderungen) als auch in der Durchführungsphase (auf operativer Ebene in Form von Design Blocks) eine zentrale Rolle.

In der Initiierungsphase finden gemeinsame Besprechungen mit den Stakeholdern und Stakeholdern statt, um ihnen die Gelegenheit zu geben, Ideen einzubringen und ihre Erwartungen und organisationsspezifischen Ziele zu formulieren und ein gemeinsames Verständnis zu schaffen. Diese ersten Ideen und Annahmen bestimmen die künftigen Anforderungen. Dabei werden mehrere alternative Lösungen in Betracht gezogen, die mittels einer SWOT-Analyse auf ihre Stärken, Schwächen, Chancen und Bedrohungen (wie im Business Case dargelegt) untersucht werden. Das Team hat jetzt ein Bild vom Ökosystem der Lösung und kann, wenn das Projekt genehmigt wird, mit der Erstellung eines ersten MVP beginnen.

In der Durchführungsphase werden bei jeder Iteration auch Design Blocks verwendet, um die Hypothesen der jeweiligen Merkmale noch einmal zu überprüfen und das gemeinsame Verständnis (erneut) sicherzustellen. Dies ist sehr wichtig, damit die Erwartungen weiter übereinstimmen und ein gemeinsames Verständnis besteht, da sich der Kontext der Lösung möglicherweise ändert und Erkenntnisse aus früheren Iterationen gewonnen werden.

Validierten Prototyp erstellen

Lean UX legt nahe, dass schnelle Prototypen von essenzieller Bedeutung sind, weil kaum Entwicklungsaufwand anfällt, der den validierten Lernprozess verzögern und somit Verschwendung verursachen könnte. In der Durchführungsphase (die durch den iterativen Charakter von PM²-Agile unterstützt wird) überprüft das agile Projektkernteam (A-PCT) die Annahmen und Hypothesen, entwickelt anschließend mithilfe von Design Blocks Low-Fidelity-Prototypen und validiert diese mit den Stakeholderinnen und Stakeholdern.

Wenn das Team und die Stakeholderinnen und Stakeholder sich einig sind, entwickeln UI-Designer High-Fidelity-Prototypen und legen sie den Stakeholderinnen und Stakeholdern und Nutzenden zur Validierung vor. Danach kann der Prototyp bereitgestellt werden.

Prototyp bereitstellen

In diesem dritten Schritt wird, nachdem die geschäftlichen Stakeholderinnen und Stakeholder und das agile Projektkernteam (A-PCT) ein gemeinsames Verständnis hergestellt und sich geeinigt haben, der Prototyp der User Experience bereitgestellt. Es ist zu beachten, dass der Akt der Bereitstellung nicht unbedingt eine technische Bereitstellung beinhaltet. Ziel ist es, ein für den nächsten Schritt bereites Experiment einzufrieren und als Instrument für validiertes Lernen zu nutzen.

Evaluieren und Lernen

Nach der Bereitstellung des High-Fidelity-Prototyps werden reale, objektive Nachweise verfügbar, die dem gesamten Team bei der Validierung der zuvor formulierten Hypothesen helfen. Diese Nachweise ermöglichen einen objektiven Lernprozess, der die Entscheidungsfindung unterstützt, darunter die Entscheidung über Umschwenken (Pivot) oder Fortfahren (Persevere), d. h., ob ein Merkmal, eine Reihe von Merkmalen oder sogar die gesamte Lösung angepasst oder unverändert weiterverfolgt werden soll. Wenn die Entscheidung Fortfahren lautet, werden die agilen Teammitglieder (ATeM) durch die vorbereitenden Arbeiten in der Lage sein, in den folgenden Iterationen mit der Entwicklung der spezifischen Elemente der Lösung zu beginnen.

Die agilen Teammitglieder (ATeM) führen Usability-Tests und Interviews mit Nutzenden und Stakeholderinnen und Stakeholdern durch. Diese stehen allen offen, sodass das Team testen kann, ob die aufgestellten Hypothesen mit den Erwartungen der Stakeholderinnen und Stakeholder und Nutzer-schaft im Einklang sind. Hierbei kann es sich um informelle Feedback-Gespräche oder ein Hauptevent in PM²-Agile, wie einen Iterationsreview oder eine Besprechung im Rahmen des Release Planning, handeln.

Das Lean UX-Mindset konzentriert sich auf den gemeinsamen Erstellungsprozess und unterstützt die PM²-Agile-Themen. Dadurch, dass agile Teams und Stakeholderinnen und Stakeholder zusammengebracht werden, entsteht ein ständiger Austausch, bei dem die Lieferung digitaler Lösungen im Hinblick auf Planung, Koordinierung und Berichterstattung, Anforderungen etc. laufend validiert wird.

4.2 Planung

Eines der häufigsten Missverständnisse über Agilität betrifft die Planung und hat seinen Ursprung in einem der Grundwerte agiler Softwareentwicklung: „Reagieren auf Veränderung mehr als das Befolgen eines Plans“. Es ist zu beachten, dass dabei das Befolgen eines Plans und nicht die Planung gemeint ist. Der Planungsaufwand ist bei agilen Projekten größer als bei herkömmlich gemanagten Projekten. Vom Führen der Arbeitselementeliste (WIL) und Schätzen des Aufwands für die Arbeitselemente bis hin zu Stand-up-Meetings und Iterationsplanung: der Erfolg hängt explizit von einer wirksamen und effizienten Planung ab.

Die Planung von agilen im Vergleich zu herkömmlichen Projekten unterscheidet sich in mehrfacher Hinsicht. Ein Beispiel ist die Intensität der Planungsaktivitäten, die sich bei agilen Projekten über den gesamten Projektlebenszyklus verteilt.

Es gibt auch Unterschiede in Bezug auf Detaillierungsgrad, Granularität und Format der Pläne sowie die verschiedenen an den Planungsaktivitäten beteiligten Hauptakteure.

Planung spielt eine wichtige Rolle bei der Zusammenführung und Koordinierung der Arbeiten verschiedener Personen. Dabei ist der unterschiedliche Grad der Beteiligung am Projekt, zeitliche Einschränkungen, Planungshorizonte und Zuständigkeiten zu berücksichtigen. Die Planungsmethoden

sollten so flexibel sein, dass die Projektkoordinierungsrolle dem Ziel der Bereitstellung idealer digitaler Lösungen gerecht werden kann. Damit bestimmte Elemente des Plans (einschließlich Darstellung und Granularität) für alle Teammitglieder und Stakeholderinnen und Stakeholder zugänglich und transparent sind, sollten Vereinbarungen getroffen werden. Ohne solche Vereinbarungen ist es schwierig, Ressourcen verbindlich zuzuweisen oder verschiedene parallele Arbeitsbereiche zu synchronisieren. Fehlende Transparenz und Abstimmung kann zu Abkoppelungen und Ineffizienzen führen und in der Folge Spannungen und Reibungen in der gesamten Organisation verursachen.

Beispielsweise besteht eine Kontroverse über die Notwendigkeit (und den Zeitpunkt) der Planung von Output mit konkreten Zeitplänen, festen Budgetansätzen und die Aufschlüsselung der Aktivitäten nach vereinbarten, vollständigen Anforderungen. Dies liegt daran, dass sich agile Teams (zu Recht) nicht auf die langfristige Detailplanung konzentrieren. Sie erstellen ungern Detailpläne, herkömmliche Zeitplanungen und Kostenschätzungen. Ressourcenbedarf und Zeitrahmen werden von ihnen anders gemessen: anhand von Iterationen, Story Points und der Velocity ihrer jeweiligen Aufgabe. Releases basieren auf einem „Stichtag“ und nicht auf genau definierten Anforderungen; die Produktion baut auf Inkrementen auf, in die das Feedback der Auftraggeberin oder dem Auftraggeber und anderen Stakeholderinnen und Stakeholdern einfließt. Es ist jedoch notwendig, mit den Erwartungen der Stakeholderinnen und Stakeholder auf der Management- und Lenkungsebene umzugehen, die normalerweise vor Projektbeginn vollständige Pläne haben möchten.

Indem diese Stakeholderinnen und Stakeholder in transparente agile Planungs- und Fortschrittsüberwachungsprozesse einbezogen und darüber aufgeklärt werden, lässt sich die Wirksamkeit des agilen Prozesses nachweisen.

In PM²-Agile werden eine ausreichende Planung und Dokumentation im Vorfeld vorgeschlagen, die mit den IT-Governance- und Projektmanagementanforderungen konform sind. An diesem Prozess orientiert sich die anschließende iterative und inkrementelle Planung und Entwicklung.

PM²-Agile fördert die folgenden Planungsgrundsätze:

- Die Planungszyklen (und ihr Umfang) sollten nach dem Zyklus, zu dem sie gehören, und der Art des Projekts entschieden werden.
- Alle sind an der Planung beteiligt.
- Planung ist wichtiger als Pläne: „Mach die Planung, verwirf den Plan!“.

4.3 Abstimmung und Berichterstattung

Für agile Praktiken haben sich bestimmte Begrifflichkeiten, Rituale, Formate und Metriken für eine effiziente Kommunikation und Koordinierung der Arbeit im Team entwickelt. Ein agiles Projektkernteam (A-PCT) arbeitet jedoch nicht isoliert. Möglicherweise sind andere nicht agile Teams an dem Projekt beteiligt, und es gibt verschiedene Stakeholderinnen und Stakeholder-Gruppen mit spezifischen Informationsbedürfnissen und -präferenzen. PM²-Agile erfüllt diese Bedürfnisse durch entsprechende Anpassung der Informationsarten, Formate, Granularität, Sprache und Häufigkeit der Kommunikation.

Die Abstimmung zwischen dem A-PCT und anderen Stakeholderinnen und Stakeholder-Gruppen kann unterschiedliche Themen betreffen. Die folgenden Themen sind für die meisten Arten von IT-Projekten relevant:

- Planung – Das A-PCT muss einen Releaseplan vorlegen, der leicht verständlich ist und sich in den Arbeitsplan des Gesamtprojekts einfügt. Außerdem können einige Inhalte aus den Iterationsplänen auch außerhalb des A-PCT verwendet werden.
- Berichterstattung – Die jeweiligen Metriken zur Messung des Fortschritts und die Häufigkeit der Berichterstattung durch das A-PCT sollten an die akzeptierten Metriken für die anderen Teams und Stakeholderinnen und Stakeholder (z. B. auf den Managementebenen) angeglichen werden.
- Architektur – Die architekturbezogenen Artefakte sollten gemäß den Standards der Organisation erstellt werden, damit Folgenabschätzungen, Analysen der Wiederverwendbarkeit und der Interoperabilität und andere Arten von Analysen ohne zusätzliche Gemeinkosten durchgeführt werden können.
- Bereitstellung und Einführung – Das Projektkernteam (PCT) und das agile Projektkernteam (A-PCT) sollten mit dem für den IT-Betrieb zuständigen Team zusammenarbeiten. Transparenz ermöglicht eine Zusammenarbeit, die die Integration von Prozessen, beispielsweise des Software-Konfigurationsmanagements, unterstützt.
- Risiken – Die A-PCTs können sowohl technische als auch andere Projektrisiken mit dem PM²-Risikomanagementprozess und unter Verwendung der empfohlenen Artefakte (z. B. des Risikoprotokolls) erfassen.

Die externe Abstimmung zwischen dem A-PCT und anderen Projektebenen und Stakeholderinnen und Stakeholdern wird durch regelmäßige Projektbesprechungen und Berichte unterstützt. Externe Kommunikationsevents umfassen:

- Besprechungen zur Projektnachbereitung,
- Projektstatusberichte,
- Demos und Abnahme der Liefergegenstände.

Bei der externen Abstimmung über Art und Granularität der Planungs- und Berichtseinheit können Probleme auftreten. Berichterstattung und Planung sollten in drei Hauptgruppen unterteilt werden:

- Liefereinheiten (z. B. Arbeitspakete, Lieferungen),
- Funktionseinheiten (z. B. Nutzungsszenarien, Geschäftsprozesse, Funktionen, Features),
- Arbeitseinheiten (z. B. Arbeitselemente wie User Storys, Enabler Storys, Fehler).

Diese Einheiten sollten (soweit möglich und anwendbar) miteinander verknüpft werden. Beispielsweise sollten die kleineren Arbeitselemente (Storys) miteinander in Beziehung gebracht und zu einer Funktionseinheit (z. B. einem Feature) zusammengefasst werden. Ebenso sollte eine Gruppe von Features (Funktionseinheiten) mit einer spezifischen Liefereinheit (z. B. einer Lieferung) verknüpft werden. Diese Rückverfolgbarkeit ermöglicht eine wirksamere externe Abstimmung, da sie den Stakeholderinnen und Stakeholdern eine klare Vorstellung von der Durchführung und Lieferung des aktuellen Projekts gibt.

Die interne Abstimmung innerhalb eines A-PCT wird durch eine offene, häufige, effiziente und persönliche Kommunikation unterstützt. In der nachfolgenden Abbildung sind verschiedene interne Kommunikationsevents dargestellt. Dazu zählen:

- Daily Stand-up,
- Iterationsplanung,

- Iterationsreview,
- Iterationsretrospektive.



Abbildung 24: Kommunikation

4.4 Anforderungen

Anforderungen beschreiben im Allgemeinen Merkmale, die ein Produkt oder eine Dienstleistung haben soll, um die Bedürfnisse der Stakeholderinnen und Stakeholder zu erfüllen.

Könnten die Anforderungen früh im Projektlebenszyklus vereinbart und festgelegt werden, wäre es dem Team möglich, in wirksamer Weise

- genau zu dokumentieren, was geliefert wird.
- den Bedarf an Personalressourcen zu schätzen.
- das Projektbudget zu entwickeln.
- einen vollständigen Projektarbeitsplan zu erarbeiten, sodass Ressourcen verbindlich zugewiesen werden können.

Wenn sowohl diejenigen, die die Ressourcen beantragen, als auch diejenigen, die sie zur Verfügung stellen, sicher sind, dass die Lösung in der Anforderungsspezifikation umfassend beschrieben ist, kann der herkömmliche Vorab-Planungsansatz verwendet werden. Sind die Anforderungen jedoch nicht gänzlich bekannt oder ist die Änderungsrate hoch, liefern Agile- und Lean UX-Ansätze die besseren Ergebnisse. Eine gut gemanagte agile Projektorganisation schafft die Voraussetzungen dafür, dass sich antragstellende Seite und Ressourcengeber gemeinsam weiterentwickeln, indem sie sich regelmäßig mit der Auslegung ihrer Vereinbarungen befassen. Der nächste Satz der Anforderungsdefinitionen kann daher an die gemachten Erfahrungen angepasst werden.

Die Identifizierung der Anforderungen selbst ist mit einigen strategischen Risiken behaftet.

- Anforderungen können aus einer gewissen Kenntnis möglicher Lösungen abgeleitet sein. Sie stellen daher das „Wie“ und nicht das „Was“ dar und schränken somit die Möglichkeit ein, die am besten geeignete Lösung zu finden.
- Anforderungen sind Investitionsposten. Die Erfüllung und Aufrechterhaltung jeder neuen Anforderung ist immer mit einem zusätzlichen Aufwand verbunden. Außerdem könnten diese

IT-Investitionen einen doppelten Aufwand darstellen, wenn sich die Anforderungen auf die Entwicklung oder den Kauf von Funktionen beziehen, die es bereits gibt. Zur Minimierung dieses Risikos ist es wichtig, die verfügbaren Funktionen, Komponenten und Dienste auf ihre mögliche Wiederverwendbarkeit zu analysieren.

- Ein wichtiger erster Schritt besteht darin, die Anforderungen durch Klarstellung des zu lösenden Problems oder des zu erfüllenden Bedarfs zu ermitteln. Ohne diesen Schritt können Problemlösungsmöglichkeiten übersehen oder die Auswirkungen der Umsetzung auf die gesamte Unternehmensarchitektur unterschätzt werden. Deshalb ist es wichtig, vor der Erfassung der Anforderungen das Problem oder den Bedarf im Business Case vollständig zu dokumentieren. Sobald die Lösungsarchitektur klarer ist, sind anschließend die Auswirkungen ihrer Umsetzung zu beurteilen.

Damit die Anforderungen richtig gemanagt, umgesetzt und überwacht werden, berücksichtigt PM²-Agile zwei sehr wichtige Dimensionen zur Beschreibung des Verhaltens einer Lösung:

- Abstraktionsebene – Es gibt unterschiedliche Abstraktionsebenen, auf denen Anforderungen identifiziert werden können. In PM²-Agile wird die zentrale Rolle von Features und Storys bei der Beschreibung des Verhaltens einer Lösung anerkannt.
- Umsetzungsperspektive – Die Umsetzungsperspektive einer Lösung darf nicht auf eine geschäftliche Sichtweise beschränkt werden. Bei der Entwicklung einer Lösung ist es wichtig, alle zugrunde liegenden Arbeiten, die zur Umsetzung der geschäftlichen Features notwendig sind, zu kennen und zu verstehen. Dies ist die zentrale Rolle der geschäftsbezogenen und befähigenden Elemente von PM²-Agile.

Features und Storys¹⁰

Ein Feature ist eine vorgeschlagene Funktionalität im Kontext einer Lösung, mit der die Bedürfnisse des Unternehmens, der Auftraggeberin oder des Auftraggebers und der Nutzenden erfüllt werden. Der Projektauftrag sieht eine ausführlichere Beschreibung der vorgeschlagenen Lösung vor, bei der die Features und die damit zu erfüllenden Bedürfnisse identifiziert und dargelegt werden.

Features werden im Projektauftrag in einem hohen Abstraktionsgrad beschrieben, um sicherzustellen, dass der Umfang der Lösung auf allen Ebenen der Projektorganisation genau verstanden wird. Die Beschreibung der Features sollte Aufschluss über ihren Kontext und die Nutzenhypothese geben, die bei der Bereitstellung zu verifizieren ist.

Um die vorab festgelegte Releasestrategie möglichst wirksam durchzuführen, müssen die Features nach Priorität geordnet werden. In diese Aktivität sollten wichtige Stakeholderinnen und Stakeholder wie Anforderungsmanagerin oder Anforderungsmanager (BM), Projekteignerin bzw. Projekteigner (PO) und das Anforderungsteam (BIG) eingebunden werden.

Zur Unterstützung der Priorisierung empfiehlt PM²-Agile ausdrücklich, mehrere Elemente wie geschäftlichen Mehrwert und die „Größe“ eines Features miteinander zu kombinieren. Features, die den höchsten geschäftlichen Mehrwert mit der kürzesten Turnaround-Zeit erbringen, werden zuerst bereitgestellt. Bei Projekten mit höherem Risiko sollte zudem der Risikoreduzierungsfaktor jedes Features berücksichtigt werden. Features mit einem höheren Risikofaktor sollten höhere Priorität haben.

Ein Feature wird im Kontext eines Release in mehreren Iterationen entwickelt und bereitgestellt. Da der Arbeitsrhythmus des agilen Projektkernteam (A-PCT) auf Iterationen (kleineren Zyklen) beruht,

¹⁰ Weitere Informationen zu Features und Storys können dem Tool/der Technik „Features und Storys“ entnommen werden.

müssen die Features (Funktionseinheiten) in mehrere Storys (Arbeitseinheiten) zerlegt werden, die dann in jeder Iteration geplant und umgesetzt werden können. Durch eine Bewertung des Outputs jeder Iteration anhand der fertiggestellten Storys kann man den Fortschritt jedes Features oder Release einfach messen und bei Bedarf notwendige Anpassungen vornehmen.

Eine Story ist die Darstellung einer Funktionalität, die aus Sicht der Nutzenden verfasst wird. Die Story, die von dem agilen Projektteam (A-PCT) innerhalb einer Iteration umgesetzt wird, ist ein kleiner vertikaler Schnitt des Systemverhaltens, der die inkrementelle Entwicklung unterstützt.

Storys liefern der geschäftlichen und technischen Seite gerade so viel Informationen, dass sie die Absicht verstehen.

Details werden aufgeschoben, bis die Story bereit für die Umsetzung ist. Durch Abnahmekriterien und Annahmetests werden Storys spezifischer, was zur Sicherstellung der Systemqualität beiträgt.

Storys sind vor allem für das agile Projektteam (A-PCT) bestimmt, können aber auch von anderen Stakeholderinnen und Stakeholdern wie der Anforderungsmanagerin oder dem Anforderungsmanager (BM) oder den Nutzervertretungen (URs) im Anforderungsteam (BIG) verwendet werden.

Storys sind ein einfaches, unkompliziertes Instrument zur Identifizierung und Nachverfolgung von Anforderungen. Da sie kurz sind, fallen nur geringe Gemeinkosten für ihre Erstellung und Aktualisierung bei geänderten Anforderungen an. Storys konzentrieren sich auf die Lieferung eines Werts und bilden eine gute Grundlage, um Anforderungen nach Priorität zu ordnen und umzusetzen.

In PM²-Agile werden Storys als Arbeitseinheiten oder Arbeitselemente bezeichnet und in einer nach Priorität geordneten Arbeitselementeliste (WIL) gesammelt.

Geschäftselemente und befähigende Elemente

Die Umsetzung einer Lösung konzentriert sich darauf, den von den Stakeholderinnen und Stakeholdern identifizierten Bedarf zu erfüllen. Die Definition des Lösungsumfangs und die Ermittlung der gewünschten Features erfolgen fast immer rein aus der geschäftlichen Perspektive. Bei der Definition und Entwicklung einer Lösung ist es jedoch notwendig, sich von dieser geschäftlichen Perspektive zu lösen

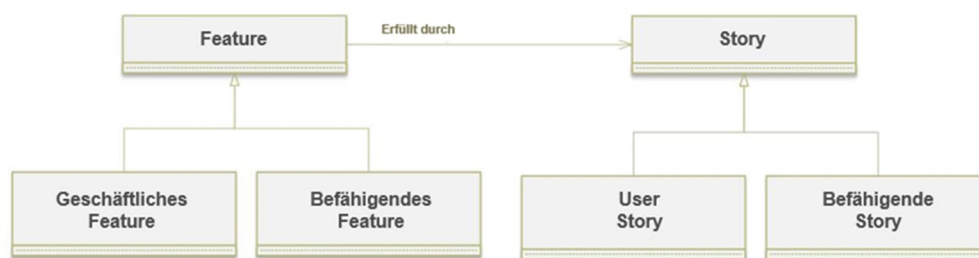


Abbildung 25: Beziehung zwischen Features und Storys

und auch den nicht geschäftsbezogenen Aufwand, der die Umsetzung der erforderlichen Features ermöglicht, in den Blick zu nehmen. In PM²-Agile wird dieser nicht geschäftsorientierte Aufwand als „Enablement-Arbeit“ (befähigende Arbeit) bezeichnet. Sie umfasst Architektur, Infrastruktur, die Bewertung von Optionen, rechtliche Compliance, kontinuierliche Verbesserung und andere nicht funktionale Anforderungen.

Ein Feature mit einer geschäftsbezogenen Perspektive ist ein „geschäftliches Feature“; ein Feature, das aus einer Perspektive der Befähigung zu sehen ist, wird als „befähigendes Feature“ bezeichnet. Da sich die Lösung, die geplant und entwickelt werden soll, aus befähigenden Elementen zusammensetzt, müssen diese stets sichtbar sein.

Die für Features dargelegten Geschäfts- und Enablement-Konzepte sind auf Storys übertragbar. Eine Story aus einer geschäftlichen Perspektive ist eine „User Story“, während eine Story aus einer Perspektive der Befähigung eine „befähigende Story“ ist.

Befähigende Elemente sind aufgrund ihrer Wichtigkeit bei der Priorisierung der Arbeitselementeliste (WIL) zu berücksichtigen. Den Systemarchitektinnen und Systemarchitekten (ArOw) fällt eine sehr wichtige Rolle zu, da sie dem Product Owner (PrOw) helfen, das richtige Gleichgewicht zwischen geschäftsbezogenen und befähigenden Arbeiten zu finden. Ein ausschließlicher Fokus auf die Entwicklung geschäftlicher Elemente kann gute Anfangsergebnisse hervorbringen.

Das agile Projektkernteam (A-PCT) wird jedoch Lieferprobleme bekommen, sobald das Fehlen einer soliden Architektur erkennbar und seine Velocity dadurch massiv beeinträchtigt wird. Bei einem ausschließlichen Fokus auf die Entwicklung befähigender Elemente erhält man eine Lösung mit einer robusten Architektur, die jedoch nicht die von den Nutzenden erwarteten Funktionen liefert.

Weitere Informationen über Features und Storys sind dem Leitfaden zu den Tools und Techniken von PM²-Agile zu entnehmen.

4.5 Schätzung und Priorisierung

Die agile Schätzung unterscheidet sich von „traditionellen“ ausführlichen Schätzungen im Voraus dadurch, dass sie das Konzept *gerade genug Details im Voraus* (*just enough upfront detail*) beinhaltet. Dabei wird der Aufwand, der in die Schätzung investiert werden sollte, mit der Menge an verfügbaren Informationen und den Erwartungen oder Anforderungen im Hinblick auf die Genauigkeit der Schätzungen in Einklang gebracht.

Die Schätzung im Rahmen von PM²-Agile basiert auf drei zentralen Konzepten:

- Die relative Schätzung ist eine grobe Schätzung des Umfangs eines Arbeitselements, der in der Regel in einer neutralen Einheit wie Punkten gemessen wird.
- Velocity misst die Anzahl der Punkte, die das agile Projektkernteam (A-PCT) innerhalb einer Iteration liefern kann.
- Bei der absoluten Schätzung wird der Umfang eines Arbeitselements (gemessen in Punkten) in eine objektive ausführliche Aufwandsschätzung übersetzt, bei der in der Regel als Einheit die tatsächlichen Stunden verwendet werden. Die Schätzung des Arbeitsaufwands zeigt an, wie viel Zeit das Team für die Fertigstellung eines bestimmten Arbeitselements benötigen wird.

Grundlage für relative Schätzungen ist der Umstand, dass Menschen besser darin sind, relative Schätzungen vorzunehmen¹¹. Anstelle von Personenstunden oder Personentagen zur Bestimmung des Umfangs eines Arbeitselements werden bei PM²-Agile Punkte zur Schätzung von Arbeitselementen in Relation zu anderen Arbeitselementen verwendet.

Relative Schätzungen werden tendenziell genauer, wenn das Projekt voranschreitet und dem Projektteam sowohl das Problem als auch die Lösung klarer werden.

Es ist wichtig, die Team-Velocity zur Planung regelmäßiger Iterationen und zur Messung der Teamleistung nachzuverfolgen. Eine ausführliche Anleitung für die punktbasierte agile Schätzungstechnik enthält das Dokument „Agile Tools and Techniques“ (Agile Tools und Techniken).

¹¹ Weitere Informationen zur relativen Schätzung können dem Tool/der Technik „Planning Poker“ entnommen werden.

Zusätzlich zur Schätzung werden Arbeitselemente entsprechend ihrem Bereitstellungswert und ihrem Beitrag zur Erreichung der Projektziele priorisiert. Die Aushandlung und Vereinbarung der Priorität der einzelnen Arbeitselemente unter Berücksichtigung der Schätzungen des Teams ermöglichen die Reaktion auf unerwartete Ereignisse bei den Projekten.

Die MoSCoW-Priorisierung beispielsweise („Must have“ (diesmal unbedingt erforderlich), „Should have“ (sollte diesmal umgesetzt werden) „Could have“ (kann diesmal umgesetzt werden) „Won’t have“ (wird diesmal nicht umgesetzt)) versetzt das Team in die Lage, gemeinsam und bewusst Entscheidungen darüber zu treffen, was das „Minimum Usable Subset“ oder „Minimum Viable Product“ darstellt.

Einige Methodiken empfehlen, 60 % des Gesamtaufwands den „Must have“-Anforderungen, 20 % den „Should have“-Anforderungen und 20 % den „Could have“-Anforderungen zuzuweisen. In Fällen, in denen die „Should have“- und die „Could have“-Arbeitselemente nicht in dem vorgegebenen Zeitrahmen bewältigt werden können, wird daher eine Umplanung vorgenommen, bei der die wichtigsten Funktionen des Informationssystems – die „Must have“-Anforderungen – weiterhin erfüllt werden.

Sowohl die Schätzung als auch die Priorisierung sind Arbeitselemente, die während des Projekts kontinuierlich durchgeführt werden sollten. So kann mit Voranschreiten des Projekts der Plan, wie die angestrebten Ziele erreicht werden sollen, besser angepasst werden.

4.6 Risiken

Agiles Risikomanagement umfasst die Beseitigung von Hindernissen, die den Arbeitsfortschritt der Mitglieder des Projektteams behindern. Solche Hindernisse, die auch als Impediments oder Roadblocks bezeichnet werden, sind alle Situationen oder Ereignisse, die den Arbeitsfortschritt während einer Iteration verhindern oder blockieren. Durch die Beseitigung dieser Hindernisse wird das Gesamtrisiko eines PM²-Agile-Projekts reduziert.

Es wird häufig gesagt, dass PM²-Agile-Projekte von Natur aus weniger risikoreich seien. Das ist wahr und geht auf die Grundlagen des Agilen Manifests zurück. Die Risikomanagementaktivitäten von PM²-Projekten werden auch bei PM²-Agile angewendet und in allen Iterationen durchgeführt.

Die folgenden Prinzipien des Agilen Manifests verdeutlichen, warum PM²-Agile von Natur aus weniger risikoreich ist.

- Gemeinsame Teams – Die Zusammenführung von Projektteam und Kundin bzw. Kunde, vorzugsweise am selben physischen Ort, führt zu einer besseren Kommunikation und weniger Zeitverschwendung.
- Einfachheit – Maximieren der Menge an nicht erledigter Arbeit: Das Team sollte sich nur auf das Minimum an Arbeit konzentrieren, das erledigt werden muss. Verschwenderische Prozesse sollten beseitigt und die Anwendung von Gold-Praktiken eingestellt werden. Die Beseitigung von Verschwendung ist eine Risikomanagementpraxis, die die Grundlage von PM²-Agile bildet.
- Regelmäßige Reflexion – Durch die kontinuierliche Verbesserung der Prozesse während des gesamten Projekts wird das Risiko exponentiell reduziert. Prozessverbesserungen sollten in den folgenden Iterationen sofort umgesetzt werden, um sicherzustellen, dass dieselben Risiken im Projekt nicht erneut auftreten.

Bei PM²-Agile gibt es zwei Arten von Risikomanagementansätzen, die dabei helfen, Agile perfekt auf den PM²-Risikomanagementprozess abzustimmen. Diese sind:

- Implizit – Ist der Ansatz, der dem agilen Prozess inhärent ist, sodass sich das Risikomanagement „natürlich“ aus dem/den iterativen Planungs- und Reviewprozess und -aktivitäten ergibt.
- Explizit – Bezieht sich auf die Umsetzung von Risikomanagementstrategien zur Identifizierung, Bewertung und Minderung von Projektrisiken durch den Einsatz spezifischer agiler Risikomanagement-Tools und -Techniken.

Um eine bessere Vorstellung von den Konzepten des impliziten und expliziten Risikomanagements in PM²-Agile-Projekten zu vermitteln, wird im Folgenden erläutert, wie diese Konzepte jeweils auf die einzelnen der vier bei der PM²-Metodik für das Risikomanagement vorgesehenen Schritte angewendet werden.

- Identifizierung von Risiken – Bei PM²-Agile identifiziert das gesamte Team während Events wie Iterationsplanung, Daily Stand-up, Iterationsretrospektive usw. implizit Risiken und hält die Ergebnisse auf Whiteboards und mithilfe anderer Tools fest. Wendet das Team eine explizite Strategie an, können auf die Tagesordnung eines Planungsmeetings zusätzliche Punkte zur expliziten Identifizierung und Priorisierung von Risiken aufgenommen werden. Das Ergebnis hat Einfluss auf die für die betreffende Iteration geplante Arbeit. Darüber hinaus können Risiken kontinuierlich während des Daily Stand-up-Meetings identifiziert werden, entweder als Hindernisse (implizite Strategie) oder als Risiken (explizite Strategie).
- Risikobewertung – Projekte auf Grundlage von PM²-Agile führen in der Regel nur „qualitative Analysen“ anstelle von „quantitativen Analysen“ durch, da die kurzen Entwicklungszyklen und die ständigen Reviews im Rahmen von PM²-Agile dies möglich und effektiv machen. Qualitative Elemente, wie die Risikowahrscheinlichkeit und die Risikoauswirkungen auf einer relativen Skala von eins bis fünf, machen die Risikobewertung zu einer Teamaufgabe und liefern konsolidierte Informationen für den Risiko-Information Radiator.
- Entwicklung einer Risikobewältigungsstrategie – Bei PM²-Agile ist das gesamte Team bei jeder Projektiteration implizit in die Planung der Risikobewältigung eingebunden, da alle Mitglieder an den Planungsmeetings, Daily Stand-ups, Retrospektiven usw. teilnehmen. Wenn für das Risikomanagement ein expliziter Ansatz erforderlich ist, helfen Tools und Techniken wie risikobasierte Spikes oder Risk Boards dabei, unmittelbare Strategien für die Bewältigung von Risiken und deren Umsetzung zu entwickeln.
- Kontrolle von Risikobewältigungsaktivitäten – Die Risikobewältigung ist ein in der PM²-Methode beschriebener kontinuierlicher und laufender Prozess. Aufgrund seiner iterativen Natur sind diese Überwachungs- und Kontrollprozesse in PM²-Agile implizit enthalten. Daily Stand-ups ermöglichen durch die Offenlegung potenzieller Bedrohungen und Hindernisse eine regelmäßige Überwachung von Risiken.
- Im Rahmen von PM²-Agile erfolgt eine Neubewertung von Risiken mindestens am Ende der Iteration, während der Iterationsretrospektive. In dieser werden festgestellte Risiken und Bedenken erneut betrachtet, um in zukünftigen Iterationen zu implementierende Änderungen festzulegen.

4.7 Qualität

PM²-Agile nimmt Bezug auf zwei unterschiedliche, aber komplementäre Arten von Qualität – die Qualität des Entwicklungsprozesses und die Qualität der Outputs des Prozesses.

Bei der PM²-Methodik soll durch das Projektqualitätsmanagement sichergestellt werden, dass das Projekt die erwarteten Ergebnisse auf möglichst effiziente Art und Weise erreicht und dass die Liefergegenstände von den relevanten Stakeholderinnen und Stakeholdern akzeptiert werden. Qualität entsteht dabei durch die Zusammenarbeit, das gemeinsame Verständnis und die Empathie zwischen allen Stakeholderinnen und Stakeholdern. Es ist ein Anzeichen von Erfolg, wenn der Endnutzende die Outputs des Prozesses als nützlich und intuitiv empfindet.

Bei PM²-Agile wird die Qualität des Entwicklungsprozesses gesteuert, indem der Ansatz, der am besten zu dem Projektumfeld passt, individuell angepasst wird und häufig das gesamte agile Projektkernteam (A-PCT), die Anforderungsmanagerin oder der Anforderungsmanager (BM) und die Vertreterinnen oder Vertreter der Nutzergemeinschaft in jedem Timebox-Arbeitszyklus einbezogen werden. In der Summe müssen sie sich ausschließlich darauf konzentrieren, Ergebnisse zu erzielen, die von der Organisation der Projekteignerin bzw. des Projekteigners (PO) als werthaltig wahrgenommen werden.

Um zweckdienliche Bereitstellungen zu erreichen, sind eine enge Zusammenarbeit zwischen der Projekteignerin bzw. dem Projekteigner (PO) und der Lösungsanbieterin bzw. dem Lösungsanbieter (SP) sowie der Einsatz agiler Tools und Techniken erforderlich. Timebox-Arbeitszyklen und häufige Nutzer-Demos bieten Zeit und Raum für gemeinsame Verbesserungen. Dies ermöglicht es dem agilen Projektkernteam (A-PCT) zu prüfen, ob die Lösung in die gewünschte Richtung geht, indem sie den geschäftlichen Stakeholderinnen und Stakeholdern maximalen Mehrwert bietet.

4.8 Weiterentwicklung und Veränderung

Die Dimension Weiterentwicklung und Veränderung eines Projekts betont das Erfordernis, die natürliche Weiterentwicklung des Projektumfangs anzuerkennen und zu begrüßen. Mit Voranschreiten des Projekts gewinnen sowohl die leistende Organisation als auch die anfordernde Organisation ein besseres Verständnis für das zu lösende Problem und die Art und Weise, wie es angegangen werden kann.

Da durch PM²-Agile die PM²-Methodik verbessert und erweitert wird, sollten Änderungen mit gravierenden Auswirkungen auf den Entwicklungsaspekt des Projekts innerhalb des übergeordneten Projekt-Änderungsmanagementprozesses gesteuert werden. Zudem sollten agile Teammitglieder (ATeM) einbezogen werden, um ein richtiges Verständnis der Änderungsanforderungen zu gewährleisten.

Aus der spezifischen Perspektive von PM²-Agile ist es wichtig sicherzustellen, dass Änderungsanforderungen in die entsprechenden Arbeitselemente, aus denen die Arbeitselementeliste (WIL) besteht, übersetzt werden. Diese Arbeitselemente werden priorisiert und anschließend im Rahmen einer oder mehrerer Iterationen implementiert.

PM²-Agile trägt der natürlichen und erwarteten Weiterentwicklung und Veränderung während des Projekts durch Release Planning und iterative und inkrementelle Entwicklungspraktiken Rechnung.

Die Praxis des Release Planning konzentriert sich auf die übergeordnete (Makro-)Planung des vollständigen bekannten Projektumfangs. Jedes geplante Release umfasst alle freizugebenden Arbeitselemente und beschreibt, wann diese über die verschiedenen Iterationen hinweg implementiert werden sollen. Sie basiert auf der Idee, einen grobgranularen Projektplan zu erstellen und diesen als Leitfaden für die Entwicklung feingranularer Pläne in den einzelnen Iterationen zu verwenden. Somit ist offensichtlich, dass eine richtig geschätzte und priorisierte Arbeitselementeliste (WIL) für die Sicherstellung eines ordnungsgemäßen Release Planning kritisch ist.

Release Planning ist ein kontinuierlicher Vorgang, in dem sich das zunehmende Wissen des Teams über die Lösung widerspiegeln sollte. Er verbessert die Genauigkeit der Projektplanung und ermöglicht aktualisierte Releasepläne, aus denen eindeutig hervorgeht, was das Team in den einzelnen geplanten Releases zu liefern beabsichtigt.

Durch die Übernahme des Release Planning kann sich das agile Projektkernteam (A-PCT) besser auf den zunehmenden Arbeitskräftebedarf einstellen. Durch das Release Planning wird zudem für die Projekteignerin bzw. den Projekteigner (PO) sichtbar, wie sich die Lösung entwickelt, und wird ihr bzw. ihm genug Zeit gegeben, um zu reagieren, wenn die Entwicklung nicht wie anfänglich geplant verläuft.

Der Releaseplan ist der wichtigste Output der Release Planning-Aktivität. Als Teil des Entwicklungsarbeitsplans ist er ein zentrales Element für die Steuerung der Weiterentwicklung und Veränderung innerhalb des Entwicklungsaspekts eines Projekts und für die Kommunikation von Fortschritten und Trends an die oberste Führungsebene. Weitere Informationen zum Releaseplan enthält Abschnitt 7.2.2.2, „Releaseplan“.

Die iterativen und inkrementellen Entwicklungspraktiken beschreiben, wie eine Lösung in Inkrementen erstellt wird, während ein Zyklus immer wieder durchlaufen/verfeinert wird, um Feedback zu erfassen und entsprechende Anpassungen vorzunehmen.

Inkrementelle Entwicklung ist ein Muster, das bei den meisten Projekten angewendet wird. Sie basiert auf der Idee, dass ein System in einer Reihe von Inkrementen aufgebaut wird und jedes Inkrement eine Teilmenge der Funktionalität des endgültigen Systems hinzufügt. Das System wird im Laufe der Iterationen des Projekts immer weiter vervollständigt. Die Alternative zur inkrementellen Entwicklung wäre die Entwicklung des gesamten Systems mit einer einzigen Big-Bang-Integration am Ende.

PM²-Agile beruht auf der inkrementellen Entwicklung, die ein hervorragendes Instrument zur Erleichterung der Planung ist, da bei ihr die Lösung in Teile zerlegt und anhand einer spezifischen Strategie organisiert wird. Dies bildet die Grundlage für eine solide Release Planning-Aktivität.

Die iterative Entwicklung basiert auf der Idee, dass Zeit für das Review und die Verbesserung von Teilen des Systems auf der Grundlage des Feedbacks der Stakeholderinnen und Stakeholder und des Selbstlernprozesses des Teams eingeplant werden muss. Die alternative Strategie zur iterativen Entwicklung wäre, alles gleich beim ersten Mal richtig zu machen.

Bei PM²-Agile sind alle Iterationen in Timeboxen untergebracht und haben die gleiche Dauer. Dies ist aus drei Gründen wichtig:

- der „Herzschlag“ des Projekts wird festgelegt,
- die allgemeine Ungewissheit wird reduziert (es gibt viele Faktoren, die nicht vorhergesehen werden können, es kann jedoch gesteuert werden, wann eine Iteration endet),
- die Beurteilung der Leistung des Projektteams wird erleichtert.

Die Arbeit in kleinen und regelmäßigen Iterationen ist ein guter Weg, um wertvolles Feedback zu sammeln. Regelmäßige Iterationen stellen sicher, dass das Team sich innerhalb der Vorgaben bewegt, und kleine Iterationen machen es leichter, die sich aus dem Feedback ergebenden erforderlichen Änderungen umzusetzen.

Kleine und regelmäßige Iterationen unterstützen die Teams zudem dabei, Designentscheidungen auf den letzten vertretbaren Moment zu verschieben (sogenanntes Set-Based Design). Diese Iterationen

sind sehr wichtige Lernzyklen, die dem Team dabei helfen, seine Optionen im Laufe des Projekts einzugrenzen, anstatt von Anfang an alle Entscheidungen zu treffen, wenn noch keine ausreichenden Informationen vorliegen.

4.9 Architektur

Die in jedem IT-Projekt getroffenen spezifischen Software-Architekturentscheidungen ergänzen bei ihrer Implementierung die allgemeine Unternehmensarchitektur. Wie bei anderen agilen Praktiken wird bei PM²-Agile dieses Prinzip auch auf die Lösungsarchitektur angewendet, wodurch die Balance zwischen Autonomie und Kohärenz gewahrt wird.

Das agile Projektkernteam (A-PCT) sollte in einer Umgebung arbeiten, die kreative Designentscheidungen und Innovation unterstützt. Um eine langfristige Effizienz (z. B. durch die Wiederverwendung von Diensten und Komponenten), Interoperabilität und Flexibilität zu erreichen, sollte das Team den Prinzipien und Standards der Unternehmensarchitektur folgen.

Bei PM²-Agile werden zwei Hauptartefakte zur standardisierten Dokumentation der Lösungsarchitektur und der dazugehörigen Bereitstellungsaspekte verwendet, die von einem IT-Governance-Gremium definiert und einem Enterprise Architecture Office unterstützt werden:

- Systemarchitekturübersicht – Projekte, die einen Informationssystementwicklungsaspekt beinhalten, sollten dieses Artefakt verwenden. Es vermittelt die grundlegenden Ideen (Compliance) und veranschaulicht die wesentlichen Merkmale der vorgeschlagenen Architektur und ihrer wichtigsten Bausteine.
- Geschäftsmodell – Beschreibt die operativen Aspekte der Architektur, wobei der Schwerpunkt auf der IT-Systeminfrastruktur liegt. Das Geschäftsmodell empfiehlt sich besonders für Informationssysteme, die in einem Unternehmensrechenzentrum gehostet werden.

Diese beiden Artefakte sind eine bewährte Art der Dokumentation von bestimmten Entscheidungen hinsichtlich der Lösungsarchitektur. Sie gewährleisten jedoch keine strukturierte und vergleichbare Weise der Dokumentation, die eine Analyse der Auswirkungen oder der Compliance ermöglicht. Eine der Aufgaben eines Architecture Office ist die Verbesserung und Harmonisierung der angewendeten Unternehmensarchitekturmethoden und -techniken.

Da die Lösung im Laufe der Zeit klarer wird und sich die Softwarearchitektur und die Infrastrukturanforderungen weiterentwickeln, sollten beide Artefakte auf dem neuesten Stand gehalten werden. In Abhängigkeit von dem Fortschritt des Projekts und den Auswirkungen der Änderungen sollten die Artefakte in regelmäßigen Abständen erneut an das IT-Governance-Gremium übermittelt werden.

Je nach Organisation kann die Bedeutung von im Voraus getroffenen Entscheidungen, Mustern usw. im Hinblick auf das Architekturdesign und die Dokumentation variieren. Falls bereits eine Referenzarchitektur existiert, wird erwartet, dass architektonische Mechanismen, Muster und Stile verwendet werden, da sie bereits von der Organisation bewertet, ausgewählt und standardisiert wurden.

Bei PM²-Agile werden Softwaredesignentscheidungen mittels der Praxis der Evolutionären Architektur getroffen. Diese Praxis beschreibt, wie die Softwarearchitektur schrittweise aufgebaut und verbessert wird, während gleichzeitig während der Softwareentwicklung aufgetretene architektonische Probleme aufgedeckt und behoben werden. Hierdurch werden technische Risiken ohne einen signifikanten architektonischen Arbeitsaufwand im Vorfeld gemindert. Diese Praxis

- verbessert die Qualität und Produktivität, da weniger zeitaufwendige und fehleranfällige Korrekturen für spät entdeckte Probleme erforderlich sind, die auf Architekturmängel zurückzuführen sind. Dies ist möglich, weil die Architektur frühzeitig validiert wird, sodass wesentliche architektonische Probleme behoben werden können, bevor der Großteil der Entwicklungsarbeit abgeschlossen ist.
- reduziert die Markteinführungszeiten durch die Konzentration auf die Wiederverwendung vorhandener Komponenten. Sie verbessert die Konsistenz und Wartbarkeit des Systems, indem gemachte Erfahrungen aus der Entwicklung zurück in die Architektur einfließen.
- erhöht die Vorhersehbarkeit, indem die technischen Bereiche mit den höchsten Risiken zuerst identifiziert und implementiert werden. Sie verbessert die Fähigkeit des Teams zur Reaktion auf Änderungen, indem sie den Architekturzyklus verkürzt und den Zeitaufwand für die Überarbeitung der Architektur bei auftretenden Änderungen minimiert.

Die wichtigsten Grundsätze der Evolutionären Architektur sind:

- Durchführung von Architekturarbeiten „just in time“ für alle anderen Arbeiten,
- Dokumentation wichtiger architektonischer Entscheidungen und ungelöster Probleme,
- Implementierung und Test zentraler Funktionen als Weg zur Behebung architektonischer Probleme.

Es ist übliche Praxis, in frühen Phasen des Projektlebenszyklus eine grobe Architekturmodellierung durchzuführen. Dies hilft dem Team und anderen Stakeholderinnen und Stakeholdern dabei, sich über die während des Projekts zu verfolgende technische Strategie zu verständigen. Außerdem steigert sie die Produktivität, reduziert technische Risiken und Entwicklungszeiten, verbessert die Kommunikation und die Teamorganisation und ermöglicht eine Skalierung der agilen Softwareentwicklung.

Bei der Einführung der Evolutionären Architektur entwickeln sich die grundlegende Systemarchitekturübersicht und das grundlegende Geschäftsmodell weiter. Änderungen an diesen Artefakten sollten innerhalb des Projektkernteam (PCT) gesteuert und dokumentiert werden, um sicherzustellen, dass beide Artefakte auf aktuellem Stand sind und alle betroffenen Stakeholderinnen und Stakeholder über die wichtigsten Änderungen informiert werden.

4.10 Compliance und Sicherheit

Das PM²-Agile-Modell ist unternehmensorientiert und berücksichtigt bei den folgenden Aspekten die Richtlinien und Kommunikation der Organisation im Zusammenhang mit Informationssystemen:

- Dokumentenmanagement,
- Datenschutz,
- Sicherheit.

Dokumentenmanagement

Bei der Erstellung einer Lösung, die einen Dokumentenmanagementaspekt beinhaltet, unabhängig davon, welches Medium verwendet wird, ist es wichtig, bei der Implementierung die von der Organisation festgelegten Verfahren zu berücksichtigen.

PM²-Agile empfiehlt die Erstellung einer Checkliste auf Basis der festgelegten Verfahren, um zu überprüfen, ob bei dem Projekt Überlegungen zum Dokumentenmanagement zu berücksichtigen sind. Diese Checkliste sollte beim Entwurf der Systemarchitekturübersicht herangezogen werden.

Ein weiterer zentraler Aspekt des Dokumentenmanagementansatzes ist die Gewährleistung einer größtmöglichen Interoperabilität zwischen Systemen und Nutzenden.

Datenschutz

PM²-Agile empfiehlt die Definition von Handlungsempfehlungen zum Datenschutz als Teil des Softwareengineering, wobei diese auf die jeweiligen Rechtsvorschriften abgestimmt sein müssen. Standardmäßig konzentriert sich PM²-Agile auf die Datenschutz-Grundverordnung (Verordnung (EU) 2016/679), die Vorschriften und Anforderungen im Zusammenhang mit der Verarbeitung personenbezogener Daten natürlicher Personen enthält.

Bei der Entwicklung einer Lösung sind spezifische Anforderungen bezüglich der Datenverarbeitung zu dokumentieren. Der Umgang mit Datenschutzfragen erfordert außerdem eine enge Zusammenarbeit zwischen verschiedenen Rollen, um sicherzustellen, dass der Bedarf für die Verarbeitung personenbezogener Daten ordnungsgemäß gesteuert wird. Alle diese Überlegungen müssen in der Systemarchitekturübersicht berücksichtigt werden, wenn die grobe Strategie der Architektur vorgeschlagen wird.

Sicherheit

Da Sicherheit ein kritischer Aspekt in jeder Organisation ist, empfiehlt PM²-Agile bei der Entwicklung einer Lösung die Festlegung einer spezifischen Praxis für den Umgang mit diesem Thema. Zu dieser Praxis sollte ein Sicherheitsplan gehören, in dem alle sicherheitsrelevanten Aktivitäten dokumentiert und alle erforderlichen Maßnahmen auflistet werden, um sicherzustellen, dass das System in einer sicheren Umgebung entwickelt wird und die geeigneten Sicherheitsmerkmale enthält.

Der Sicherheitsplan sollte die Sicherheitsanforderungen definieren und einen schrittweisen Managementplan für die Einhaltung dieser Anforderungen enthalten. Er sollte zudem Teil einer umfassenden Erklärung zur Sicherheitszertifizierung sein, deren Zweck darin besteht, die technischen Merkmale des Systems zu testen und zu evaluieren, die für die Systemumgebung vorgesehenen administrativen, personellen und physischen Schutzmaßnahmen zu überprüfen und die formelle Übernahme aller Restrisiken durch die oberste Führungsebene zu dokumentieren.

Weitere in die Erklärung zur Sicherheitszertifizierung aufzunehmende Elemente sind der Sicherheitstestplan und -bericht sowie die Sicherheitsrisikobewertung.

4.11 Entwicklung

Im Mittelpunkt von PM²-Agile steht die Entwicklung von Informationssystemen, die Teil der Lösung des Projekts sind. In diesem Kontext werden durch Entwicklungsaktivitäten wie Konzept, Entwicklung, Bau oder Test die Spezifikationen zu etwas Greifbarem und ermöglichen es dem System, Gestalt anzunehmen.

Das Hauptziel der Entwicklungsaktivitäten besteht darin, sicherzustellen, dass die Anforderungen durch einen iterativen und inkrementellen Ansatz angemessen konzipiert, umgesetzt und verfügbar gemacht werden. Die Anforderungen sollten zudem die Einschränkungen im Hinblick auf Architektur, Infrastruktur und Governance berücksichtigen und den von den geschäftlichen Stakeholderinnen und Stakeholdern erwarteten Mehrwert bieten.

Um dazu beizutragen, dem Entwicklungsaspekt die erforderliche „Agilität“ zu verleihen, empfiehlt PM²-Agile drei unterschiedliche, sich jedoch ergänzende Praktiken: Kontinuierliche Integration, Evolutionäres Design und Testgetriebene Entwicklung.

Im Rahmen der Kontinuierlichen Integration integrieren die Teammitglieder ihre Arbeit häufig (mindestens täglich).

Der erforderliche Aufwand für die Integration eines Systems wächst im Laufe der Zeit exponentiell. Daher wird durch eine häufige Integration des Systems der Gesamtintegrationsaufwand reduziert, weil Integrationsprobleme früher identifiziert und gelöst werden. Dies führt zu einem Produkt mit höherer Qualität und besser vorhersehbaren Lieferabfolgen und -aktivitäten.

Das Prinzip der Kontinuierlichen Integration kann anhand der folgenden Aktivitäten beschrieben werden.

- Nach der Vornahme von Änderungen in ihren validierten Arbeitsbereichen führen Entwickler Tests von Modulen durch, bevor sie sie dem Team zur Verfügung stellen.
- Die Änderungen aller Entwickler werden in einem Integrationsarbeitsbereich zusammengeführt und häufig getestet (mindestens täglich, idealerweise aber jedes Mal, wenn ein neuer Änderungssatz verfügbar ist).

Durch die erste Aktivität wird sichergestellt, dass Änderungen in einer Konfiguration vorgenommen werden, von der bekannt ist, dass sie gut ist, und dass sie getestet wurden, bevor sie verfügbar gemacht werden. Durch die zweite Aktivität werden Integrationsprobleme frühzeitig identifiziert, sodass sie korrigiert werden können, wenn die Änderung im Gedächtnis des Entwicklers noch frisch ist.

Die Praxis des Evolutionären Designs basiert auf der Annahme, dass sich das Design im Laufe der Zeit weiterentwickelt. Sie minimiert den Dokumentationsbedarf und bietet dennoch Orientierung für das Treffen und Kommunizieren von Design-Entscheidungen.

In jeder Designrunde ergänzen, verfeinern und überarbeiten die Entwickler die Lösung. Dies lässt sich wie folgt zusammenfassen:

- Verstehen der Einzelheiten neuer Anforderungen.
- Identifizierung von Designelementen.
- Feststellen, wie Elemente bei der Realisierung des Szenarios zusammenwirken.
- Ausgestaltung von Designentscheidungen.
- Gestaltung der internen Elemente.
- Kommunikation des Designs.
- Verstehen der Architektur.
- Evaluierung des Designs.

Die Testgetriebene Entwicklung ist ein Entwicklungsansatz, bei dem vor dem Programmieren von tatsächlichem Code Testfälle erstellt werden. Diese Tests dienen als Leitplanken, da der zu entwickelnde Code diese Tests bestehen muss, um als akzeptabel zu gelten.

Die Testgetriebene Entwicklung verkürzt die Lieferzeiten, indem sie die für die Integration und Stabilisierung von Builds benötigte Zeit reduziert. Sie verbessert die Produktivität, da Fehler nahe am Zeitpunkt der Einführung gefunden und behoben werden.

Die Testgetriebene Entwicklung erhöht auch die Gesamtqualität der Software, indem sie gewährleistet, dass neu entwickelter Code vor dem Einchecken getestet und vorhandener Code einer Regressionsanalyse unterzogen wurde.

Die Praxis der Testgetriebenen Entwicklung verändert die Denkweise von Entwicklern bzw. Entwicklerinnen. Entwicklertests werden nicht als nachgeordnete Frage verstanden, sondern als Bestandteil des Alltags der Softwareentwicklung.

4.12 Software-Konfigurationsmanagement

Bei der Entwicklung einer Lösung werden Hunderte von Artefakten wie Dokumente, Code-Dateien, Skripte, Liefergegenstände usw. erstellt und müssen verwaltet werden. Bei PM²-Agile sieht das Software-Konfigurationsmanagement wesentliche Aktivitäten vor, mit denen die Änderungen an der Konfiguration konsistent kontrolliert werden und ihre Integrität und Nachverfolgbarkeit gewährleistet werden.

Das Software-Konfigurationsmanagement verfolgt vier Hauptziele:

- Identifizierung und Verwaltung aller Elemente, die Teil des Konfigurationsmanagements sind – Das agile Projektkernteam (A-PCT) muss alle Elemente (Klassen, Bibliotheken, Testskripte, Bereitstellungsdateien usw.) identifizieren, die es im Blick behalten muss, und sie mithilfe spezifischer Tools verwalten.
- Vereinfachung der Erstellung verschiedener Versionen der Lösung – Die Lösungsentwicklung ist ein inkrementeller Prozess, bei dem regelmäßig verschiedene Versionen demonstriert und bereitgestellt werden. Daher ist es von größter Wichtigkeit, eine bestimmte Version der Lösung einfach erstellen und liefern zu können.
- Sicherstellen der Aufrechterhaltung der Qualität bei der Weiterentwicklung der Konfiguration – Da sich die Konfiguration ständig weiterentwickelt, muss sichergestellt werden, dass die Qualität der Lösung gewahrt bleibt. Dies geschieht durch die Nachverfolgung der Konfigurationselemente und ihrer Änderungshistorie und die Fähigkeit, die Auswirkungen der Änderungen zu erkennen.
- Produktivitätssteigerung durch Fehlerreduzierung – Mit den richtigen Tools kann das Software-Konfigurationsmanagement so weit automatisiert werden, dass menschliches Eingreifen kaum noch nötig ist. Durch die drastische Reduzierung des menschlichen Eingreifens und somit von Fehlern nimmt die Produktivität zu.

Als Prozess umfasst das Software-Konfigurationsmanagement eine Reihe von Schritten, die bei PM²-Agile als Schlüssel für die Erreichung der genannten Ziele angesehen werden.

- Identifizierung – Eindeutige Identifizierung aller Konfigurationselemente, die in die Konfigurationsmanagement-Datenbank aufgenommen werden sollen. Die Verfolgung eines objektorientierten Ansatzes ist dabei in der Regel sinnvoll, da dadurch auch die Aggregation in ähnlichen Objekttypen (Dokumenten, Skripten usw.) erleichtert wird. Einige der am häufigsten verwendeten Eigenschaften sind Name, Version, Art des Objekts, benötigte Ressourcen usw.
- Änderungskontrolle – Die Änderungskontrolle ist bei der PM²-Methodik bereits als Prozess vorgesehen, der verschiedene Aktivitäten im Zusammenhang mit Projektänderungen definiert, darunter Dokumentation, Bewertung und Genehmigung.

- Versionskontrolle – Einrichtung einer Gruppe von Verfahren und Tools, die die Erstellung und Verwaltung von mehrfachen Instanzen der einzelnen Objekte in der Konfigurationsmanagement-Datenbank ermöglichen. Für die Softwareentwicklung sollten agile Teammitglieder (ATeM) in der Lage sein, alle relevanten Konfigurationsobjekte zusammenzustellen und eine bestimmte Version der Lösung zu erstellen.
- Konfigurationsaudit – Hierbei handelt es sich eindeutig um eine Qualitätssicherungsaktivität. Ihr Ziel ist es, bei der Vornahme von Änderungen die Aufrechterhaltung der Qualität sicherzustellen. Der Umfang dieser Aktivität ist von dem Kontext des Projekts abhängig. Typischerweise geht es um Fragen wie die Überprüfung, ob die Änderung ordnungsgemäß gekennzeichnet und dokumentiert und der Autor identifiziert wurde oder ob die Verfahren des Konfigurationsmanagements zur Feststellung, Aufzeichnung und Meldung der Änderung ordnungsgemäß eingehalten wurden. Mithilfe des Konfigurationsaudits kann sichergestellt werden, dass die richtigen Konfigurationsobjekte in einen bestimmten Build integriert wurden.
- Berichterstattung – Bereitstellung von Informationen an alle Personen innerhalb des Projekts (und der Organisation), die Informationen über die vorgenommenen Änderungen benötigen. Die Berichte sollten Angaben zum Initiator der Änderung, dem Datum/der Uhrzeit der Änderung, dem Umfang der Änderung sowie andere relevante Informationen enthalten. Die Berichterstattung wird ausgelöst, wenn ein Konfigurationsaudit durchgeführt wird oder wenn einem Konfigurationsobjekt neue oder aktualisierte Informationen zugewiesen werden, und sollte auf den Kommunikationsmanagementplan abgestimmt sein.

Wenn ein Konfigurationselement den Prozess durchläuft, ist es möglich, dass die in einem der Schritte vorgesehenen Aktionen nicht anwendbar sind. Beispielsweise sind bei einer Änderung an einem Konfigurationsskript die Aktionen des Schritts „Änderungskontrolle“ nicht erforderlich, da anzunehmen ist, dass kein Änderungsantrag benötigt wird.

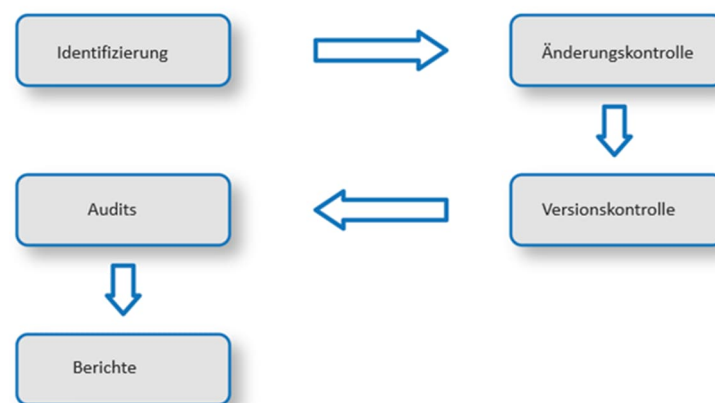


Abbildung 26: Schritte des Software-Konfigurationsmanagements

Alle relevanten entwicklungsspezifischen Überlegungen zum Software-Konfigurationsmanagement (z. B. zu verwendende Tools, anzuwendende Vorlagen) sollten in Abstimmung mit dem Qualitätsmanagementplan dokumentiert werden.

Im Folgenden sind die typischen Aktivitäten und wichtigsten Überlegungen bei der individuellen Anpassung des Konfigurationsmanagementprozesses aufgeführt.

- Zuordnung jedes Konfigurationselements zu einer bestimmten Version der Lösung,

- Führen eines Verzeichnisses aller Konfigurationselemente (Name, Erstellungsdatum, letzte Aktualisierung, Version usw.) zu Audit Zwecken,
- ggf. Beschreibung der Konventionen für die Benennung und Versionierung von Projekt- und Produktarbeitsergebnissen in Übereinstimmung mit den Unternehmensrichtlinien,
- Einführung einer standardisierten Produktverzeichnisstruktur, um Artefakte unter Versionskontrolle zu stellen,
- Synchronisationskontrolle zur Sicherstellung, dass sich parallele Änderungen nicht gegenseitig aufheben,
- Analyse von Fragestellungen im Zusammenhang mit der Einrichtung der Umgebung für das Konfigurationsmanagement, in die Folgendes einbezogen wird:
 - erwarteter Umfang der Produktdaten,
 - Verteilung des Produktteams,
 - physische Standorte der Server- und Client-Geräte.
- Einführung einer Standardvorlage für das Konfigurationsmanagementsystem des Projekts, in der geeignete Workstreams, Rollen und Zuständigkeiten festgelegt werden,
- Sicherstellung einer Konfigurationsmanagementkapazität für die Verwaltung von Änderungsanforderungen (einschließlich Verbesserungen und Fehlern), die vom Team und externen Stakeholderinnen und Stakeholdern generiert werden,
- Kenntnis und Einhaltung der Richtlinien für die Aufbewahrung und Freigabe von Projektmedien,
- Dokumentation der Einbeziehung von durch Anbieter und Unterauftragnehmer entwickelter Software in die Versionskontrolle.

4.13 Test

Agile Testaktivitäten sollten iterativ und inkrementell sein. Durch die Anwendung der Strategie „frühes Testen und häufiges Testen“ können Risiken früh im Lebenszyklus des Projekts erkannt und gemindert werden. Es wird empfohlen, die Testaktivitäten in jeder Iteration durchzuführen, beginnend mit den ersten Builds des Systems. Je nach Häufigkeit der neuen Builds kann eine Iteration mehrere Testzyklen umfassen.

Die Testaktivitäten und -techniken konzentrieren sich auf die Frage: „Was muss die Lösung leisten, bevor eine Anforderung als umgesetzt gilt?“. Durch Nutzertests und die Validierung schneller Prototypen (noch vor der Erstellung von Code) werden die Anforderungen mit spezifischen Erfüllungsbedingungen, die die Lösung einhalten muss, weiterentwickelt.

Hierzu zählen auch die Abnahmekriterien, die die Grenzen eines Arbeitselements definieren und verwendet werden, um seine Fertigstellung zu bestätigen.

PM²-Agile empfiehlt zwei Testpraktiken: Testmanagement und Concurrent Testing.

Testmanagement ist eine wichtige Überlegung für jede größere Softwareengineeringarbeit. Die Testmanagementpraxis bietet einen guten Ausgangspunkt für Praktizierende, die relativ neu in diesem Bereich sind, bietet jedoch auch ein strukturiertes Referenzmodell für erfahrenere Praktizierende. Die Tools, die bei der Planung, dem Design, der Implementierung, der Ausführung, der Bewertung und der

Verwaltung der Testaufgaben oder Arbeitsprodukte eingesetzt werden, sind ein wichtiger Aspekt und müssen mit den aktuellen Praktiken der Organisation im Einklang stehen.

Bei der Concurrent Testing-Praxis erfolgen Tests zeitgleich mit der Entwicklung in allen Iterationen. Dadurch wird verhindert, dass Teams die Tests in einer separaten Aktivität am Ende der Iterationen zusammenfassen. Durch Concurrent Testing wird das Konzept der parallel arbeitenden Funktionsteams gestärkt. Concurrent Testing erfordert ein hohes Maß an Integration und Kommunikation mit hoher Bandbreite zwischen Entwicklerinnen und Entwicklern und Testerinnen und Testern.

Beim Testen der gelieferten Inkremente sind zwei Punkte zu beachten:

- Die testbezogenen Aktivitäten sind parallel zur Anforderungserhebung und -verfeinerung zu berücksichtigen, damit das agile Projektkernteam (A-PCT) und die Stakeholderinnen und Stakeholder sich über spezifische Erfüllungsbedingungen für jede Anforderung verständigen können. Durch die Festlegung von Abnahmekriterien für jede Funktion und jede Story kann das Team überprüfen, ob das Gelieferte den Erwartungen der Nutzenden des Informationssystems entspricht.
- Tests sollten so häufig wie möglich durchgeführt werden. Idealerweise sollten alle Tests für jeden Build, der in die Mainline eingespeist wird, durchgeführt werden. Wenn dies nicht praktikabel ist, sollten Regressionstests für bestehende Funktionen durchgeführt werden, während sich der Testzyklus auf die im neuen Build fertiggestellten Arbeitselemente konzentriert. Selbst Testskripte, bei denen das Auftreten von Fehlern zu erwarten ist, liefern wertvolles Feedback. Sobald ein Testskript jedoch fehlerfrei durchläuft, sollten bei späteren Builds der Lösung keine Fehler mehr auftreten.

4.14 Deployment und Einführung

Unabhängig von der angewandten Methode muss immer sichergestellt werden, dass das in der Entwicklung befindliche Informationssystem effektiv in der technischen Infrastruktur der Organisation eingesetzt werden kann. Alle entsprechenden technischen und infrastrukturbezogenen Aspekte der Bereitstellung des Informationssystems werden im Geschäftsmodell beschrieben.

Neben den technischen Aspekten der Bereitstellung eines Informationssystems gibt es eine Reihe weiterer Überlegungen, die zu berücksichtigen sind, um eine reibungslose geschäftliche Transition zu ermöglichen. Aktivitäten wie Nutzertests, häufige Schulungen und Kommunikationssitzungen können ebenfalls zu einer besseren Aufnahme beitragen. Indem sichergestellt wird, dass Nutzende zudem ständig informiert werden und auf den Lieferzeitplan des Projekts eingestellt sind, wird eine optimale Bereitstellung der Lösung gewährleistet.

Die Deployment- und Einführungsaktivitäten sind außerdem entscheidend dafür, dass Abhängigkeiten mit anderen Projekten und Informationssystemen berücksichtigt werden.

5. Events

Beim Einsatz von PM²-Agile in einem Projekt muss eine Gruppe von Elementen berücksichtigt werden, um die Wahrscheinlichkeit einer erfolgreichen Bereitstellung zu erhöhen. Eines dieser Elemente ist ein Satz von kritischen Events, die dem agilen Projektkernteam (A-PCT) dabei helfen, eine Abstimmung in den folgenden Dimensionen zu erreichen:

- Koordination und Planung der Arbeiten,
- Art und Weise der Durchführung der Arbeiten,
- Untersuchung und Anpassung der Arbeiten und der unterstützenden Prozesse.

Diese Dimensionen werden durch den CIR-Rhythmus konkretisiert.

Der CIR-Rhythmus (Coordinate (koordinieren), Implement (implementieren), Review (überprüfen)) wird durch eine Kombination aus Events und technischen Praktiken unterstützt, die dafür sorgen sollen, dass der agile Ansatz strukturiert und konsistent ist. Alle drei der vorstehenden Aspekte spielen dabei eine entscheidende Rolle.

Der Koordinationsaspekt (Coordinate) konzentriert sich auf die Events, die die Abstimmung zwischen den Mitgliedern des agilen Projektkernteam (A-PCT) in Bezug auf Fragen wie Planung, Gegenstand der Entwicklung, Prioritäten usw. gewährleisten. Die PM²-Agile-Events, die diese Bemühungen unterstützen, sind:

- Iterationsplanung,
- Daily Stand-up,
- Release Planning.

Der Implementierungsaspekt (Implement) konzentriert sich auf die Erstellung der richtigen Lösung und stützt sich auf verschiedene technische Verfahren und Co-Creation-Techniken (statt auf spezifische Events), um sein Ziel zu erreichen. Da jedes agile Projektkernteam (A-PCT) seinen eigenen Satz von Werten, Arbeitsvereinbarungen und besonderen Fähigkeiten hat, schlägt PM²-Agile nicht explizit bestimmte Tools oder Events vor, um dem Team dabei zu helfen, seine Arbeitsweise zu definieren. Stattdessen wird der Einsatz verschiedener Tools und Techniken empfohlen, von denen einige im Abschnitt zu Tools und Techniken beschrieben sind.

Der Überprüfungsaspekt (Review) konzentriert sich auf die Aktivitäten und Events, die dazu dienen, die *Nützlichkeit* der entwickelten Lösung und die Angemessenheit des vom agilen Projektkernteam (A-PCT) angewandten Prozesses zu bewerten.

Zur Unterstützung der Erreichung dieser Ziele empfiehlt PM²-Agile zwei Hauptevents:

- Iterationsreview,
- Iterationsretrospektive.

Bei PM²-Agile wird anerkannt, wie wichtig diese Events und Aktivitäten sind, um den CIR-Rhythmus zu ermöglichen. Nichtsdestotrotz sind die in diesem Abschnitt beschriebenen Veranstaltungen und Events keine Patentlösung. Allein durch ihre Implementierung werden bestehende Probleme nicht gelöst. Die Wahrscheinlichkeit einer erfolgreichen agilen Umsetzung erhöht sich vielmehr durch eine angemessene Einrichtung von Events und Artefakten, mit richtig definierten Rollen und Zuständigkeiten, unterstützt durch die richtigen Tools und Techniken und geleitet von den richtigen Mindsets.

Die fünf vorgenannten Events werden in den folgenden Abschnitten beschrieben.

5.1 Iterationsplanung

Eine der Hauptsäulen des PM²-Agile-Modells ist die iterative Entwicklung. Sie ist ein kritischer Aspekt des gesamten Lebenszyklus eines agilen Projekts und wird durch das Konzept der Iteration ermöglicht, an dem sich die Entwicklung der Lösung orientiert. Jede Iteration trägt mit einem validierten Lösungsincrement zu einem inkrementellen Wachstum der Lösung bei.

Da Iterationen einen kurzen Zeitraum umfassen, sind eine klar definierte Strategie und ein Plan für die angestrebte Entwicklung unerlässlich, um sicherzustellen, dass das agile Projektkernteam (A-PCT) sich Iteration für Iteration auf eine sinnvolle Umsetzung konzentriert. Um diesem Erfordernis Rechnung zu tragen, empfiehlt PM²-Agile die Einrichtung eines spezifischen Events, der Iterationsplanung.

Die Iterationsplanung zielt darauf ab, einen Plan und eine Strategie zu erstellen, in denen die Verfügbarkeit und Kapazität des Teams berücksichtigt werden, um das vereinbarte Ziel zu erreichen.

Wie oft die Iterationsplanung stattfinden und wie lange sie dauern soll, sind zwei weitere wichtige Faktoren, die sorgfältig zu erwägen sind. Die Iterationsplanung sollte nicht häufiger als nötig durchgeführt werden, und es sollte dafür nicht mehr Zeit als tatsächlich notwendig aufgewendet werden. Die Art und Weise der Strukturierung und Organisation der Iterationsplanung kann den Unterschied zwischen einem erfolgreichen Event und einer unproduktiven Veranstaltung ausmachen. Daher empfiehlt PM²-Agile eine spezifische Struktur mit einer Reihe von Handlungsempfehlungen, die das Team dabei unterstützen, solch ein Event erfolgreich durchzuführen.

5.1.1 Häufigkeit und Dauer

Da die Iterationsplanung entwickelt wurde, um eine Strategie zur Erreichung eines bestimmten Ziels innerhalb der kurzen Zeitspanne einer Iteration zu erarbeiten, ist der beste Zeitpunkt für ihre Durchführung der Beginn der Iteration. PM²-Agile empfiehlt die Iterationsplanung als erste Aktivität, die vom agilen Projektkernteam (A-PCT) während der Iteration durchzuführen ist.

Ein weiterer wichtiger Aspekt ist die Dauer des Events. Ziel ist es, das richtige Gleichgewicht zwischen der investierten Zeit und der Klarheit/Objektivität des zu definierenden Plans zu finden. Da jedoch verschiedene Faktoren wie die Größe des Teams und der Iteration zu berücksichtigen sind, können keine allgemeingültigen Regeln für die Festlegung dieses Gleichgewichts angewendet werden. Daher basiert die Empfehlung von PM²-Agile auf der praktischen Erfahrung von Tausenden von Teams, die sowohl PM²-Agile als auch andere Methodiken und Rahmenwerke anwenden.

Für ein typisches agiles Projektkernteam (A-PCT) mit fünf bis neun agilen Teammitgliedern (ATeM) empfiehlt PM²-Agile, zwei Stunden pro Iterationswoche einzuplanen. Beispielsweise gilt es als bewährte Praxis, für acht agile Teammitglieder (ATeM), die in zweiwöchigen Iterationen arbeiten, vier Stunden einzuplanen.

5.1.2 Struktur einer Iterationsplanung

Die Iterationsplanung ist dazu bestimmt, das agile Projektkernteam (A-PCT) bei der Planung der inkrementellen Abarbeitung der Arbeitselementliste (WIL) zu unterstützen. Da sie die Arbeiten für die Implementierung einer Lösung abbildet, muss die Arbeitselementliste (WIL) stets priorisiert werden, um es dem Product Owner (PrOw) zu ermöglichen, ein relevantes und abgestimmtes Iterationsziel vorzuschlagen.

Mit dem Vorschlag des Iterationsziels kommuniziert der Product Owner (PrOw) seine Erwartungen. Anhand dieser Informationen und unter Berücksichtigung der Kapazitäten und Fähigkeiten legt das agile Projektkernteam (A-PCT) fest, was zu tun ist, wie es vorgehen will und wie lange es voraussichtlich dauern wird zu überprüfen, ob das vorgeschlagene Ziel erreichbar ist oder nicht.

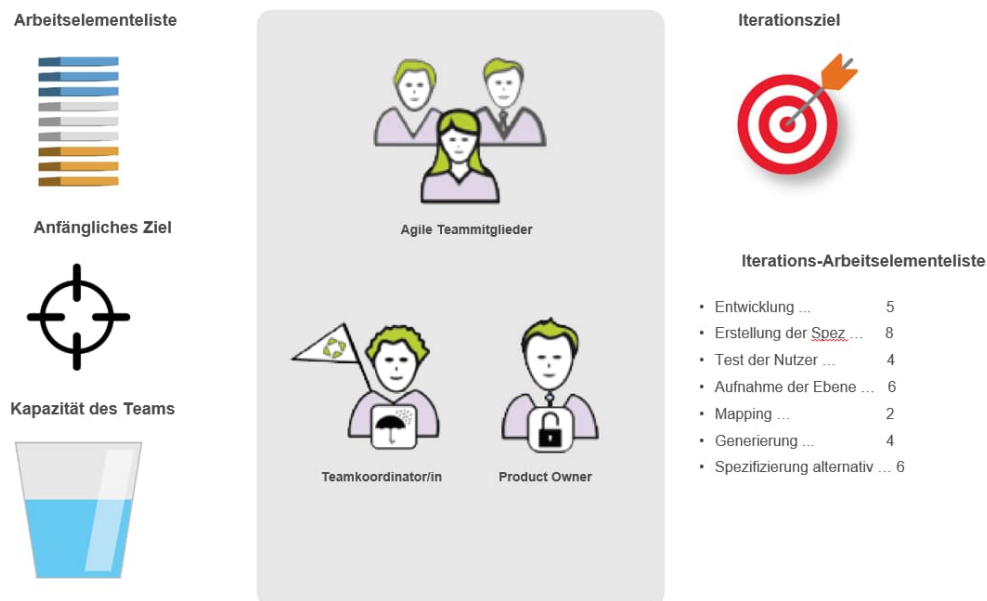


Abbildung 27: Bestandteile einer Iterationsplanung

Am Ende der Iterationsplanung sollte das agile Projektkernteam (A-PCT) ein klares und gut definiertes Iterationsziel und eine erstellte Arbeitselementliste (WIL) für die Iteration haben, die wichtige Bestandteile des Iterationsplans sind.

Nach einer Klärung der Bestandteile der Iterationsplanung ist es wichtig, ein Verständnis der Schritte für die ordnungsgemäße Implementierung zu erlangen.

Um größtmöglichen Nutzen aus der Iterationsplanung zu ziehen, schlägt PM²-Agile eine Abfolge von Schritten vor, in die das gesamte agile Projektkernteam (A-PCT) einbezogen wird. Zu diesen Schritten gehört, dass das Team seine Kapazität feststellt, die zu erledigende Arbeit definiert, eine vorläufige Einigung über das von dem Product Owner (PrOw) vorgeschlagene Iterationsziel erzielt und ein Vertrauensvotum hat.

5.1.2.1 Feststellung der Kapazität des Teams

Ein zentrales Element der Iterationsplanung ist die Frage, ob das agile Projektkernteam (A-PCT) in der Lage sein wird, das Iterationsziel zu erreichen. Um dies zu überprüfen, muss jedes agile Teammitglied (ATeM), das an der Lösung arbeitet, wissen, wie viel Zeit (vorzugsweise in Stunden) es für die nächste Iteration aufwenden kann. Dabei sollten auch der Urlaub der Teammitglieder, Feiertage und andere Ereignisse berücksichtigt werden.

Als Beispiel wird angenommen, dass an jedem Arbeitstag sieben Arbeitsstunden zur Verfügung stehen. Außerdem wird angenommen, dass eine zehn Kalendertage dauernde Iteration nur neun Arbeitstage umfasst (ein Tag ist Feiertag). Die Kapazität der agilen Teammitglieder (ATeM) berechnet sich wie folgt:

Kapazität: 9 Tage x 7 Stunden = 63 Stunden.

Somit stehen dem agilen Teammitglied (ATeM) 63 Stunden zur Verfügung, um dem Team beim Erreichen des Iterationsziels zu helfen.

Auch andere Aspekte wie ungeplante Ereignisse, Bugs, Besprechungen usw. können sich auf die Gesamtverfügbarkeit auswirken und sollten ebenfalls berücksichtigt werden. Diese Aspekte werden im nachstehenden Abschnitt zu den Handlungsempfehlungen beschrieben.

5.1.2.2 Vereinbarung des Iterationsziels

Die Definition des Iterationsziels ist der wichtigste Aspekt einer Iterationsplanung, da sich das agile Projektkernteam (A-PCT) dadurch auf das Wesentliche konzentrieren kann.

Sie fällt in die Zuständigkeit des Product Owners (PrOw) und muss mit einer priorisierten Arbeitselementliste (WIL) beginnen. Der Product Owner (PrOw) beurteilt die vorhandenen Elemente und formuliert ein sinnvolles Iterationsziel, das Mehrwert für die Lösung bietet.

Sobald das Ziel definiert wurde, leitet der Product Owner (PrOw) dieses an den Rest des agilen Projektkernteam (A-PCT) weiter, damit es sich auf die wichtigsten Elemente konzentrieren kann.

Es ist zu beachten, dass eine anfängliche Vereinbarung nicht bedeutet, dass das Iterationsziel unmittelbar genehmigt wird, da die an der Entwicklung der Lösung beteiligten agilen Teammitglieder (ATeM) ihre Verfügbarkeit und Kapazität bestätigen müssen.

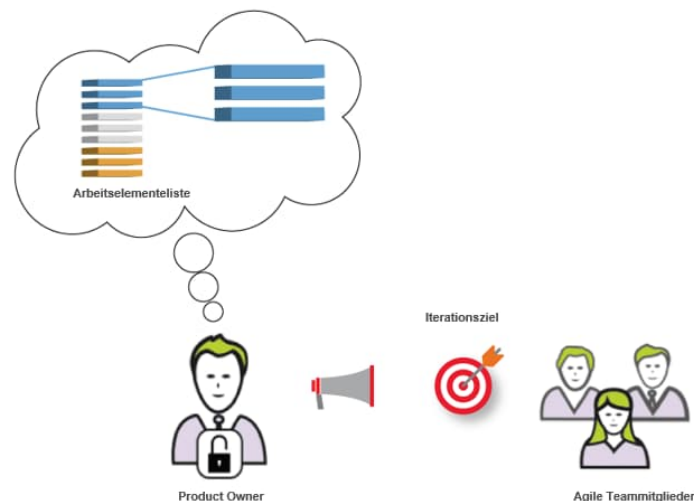


Abbildung 28: Vereinbarung des Iterationsziels

5.1.2.3 Festlegung der zu erledigenden Arbeit

Wenn die Verfügbarkeit des Teams und das definierte Iterationsziel vorliegen, kann das agile Projektkernteam (A-PCT) die zu erledigende Arbeit festlegen. Das Team sollte sich dabei auf die Elemente der Arbeitselementliste (WIL) konzentrieren, die einen Beitrag zu dem Iterationsziel leisten. Es legt für jedes Element die von ihm auszuführenden Aufgaben fest und überprüft, ob es über die Verfügbarkeit und Kapazität für deren Fertigstellung verfügt.

Das Team führt diese Routine für jedes relevante Element der Arbeitselementliste (WIL) durch, bis seine Kapazität erschöpft ist.

Zur Festlegung der zu erledigenden Arbeit empfiehlt PM²-Agile die folgenden Schritte:

1. Das agile Projektkernteam (A-PCT) wählt das erste Element der Arbeitselementliste (WIL) aus, das mit dem Iterationsziel im Zusammenhang steht.
2. Es identifiziert als Team die Aufgaben, die aus seiner Sicht erforderlich sind, um dieses Element zu erstellen und zu liefern.

3. Jede Aufgabe wird von einem agilen Teammitglied (ATeM) für die Umsetzung ausgewählt.
4. Das agile Teammitglied (ATeM) nimmt eine Schätzung der Aufgabe vor (diese sollte nicht umfangreicher als zwölf Stunden sein).
5. Jedes agile Teammitglied (ATeM) überprüft seine Kapazität nach Übernahme der Aufgabe.
6. Falls alle agilen Teammitglieder (ATeM) noch Kapazität haben, wählt das Team das nächste Arbeitselement von der Arbeitselementeliste (WIL) aus.
7. Stellt ein agiles Teammitglied (ATeM) fest, dass die Aufgabe nicht erfüllt werden kann, führen die anderen Teammitglieder eine Prüfung durch. Falls kein anderes Teammitglied die Aufgabe übernehmen kann, sollte das Team die Arbeit einstellen.
8. Sobald das Team seine volle Kapazität erreicht hat, ist es an der Zeit zu bewerten, ob das Iterationsziel erreicht werden kann oder angepasst werden muss. Dies ist das Ziel des nächsten Schritts, der Verpflichtungserklärung.

5.1.2.4 Verpflichtungserklärung

Durch die Verpflichtungserklärung wird die Vereinbarung zwischen dem Product Owner (PrOw) und den übrigen agilen Teammitgliedern (ATeM) entsprechend dem Iterationsziel formalisiert. Diese Vereinbarung ist von entscheidender Bedeutung, da sie die Erwartungen aller Beteiligten in Bezug auf die vom Team zu erbringende Leistung transparent macht.

Beim Eingehen der Verpflichtungserklärung existieren drei mögliche Szenarien. Das erste (und einfachste) ist, wenn das Team (auf Grundlage der in den vorherigen Schritten gesammelten Informationen) bestätigt, dass es das Iterationsziel erreichen kann.

Im zweiten Szenario stellen die agilen Teammitglieder (ATeM) fest, dass sie das Iterationsziel nicht erreichen können. In einem Gespräch mit dem Product Owner (PrOw) kann das Team Vorschläge für Änderungen an dem Iterationsziel unterbreiten, um dieses weniger ehrgeizig zu gestalten und somit erreichbar zu machen. Der Product Owner (PrOw) bewertet das von dem Team gegebene Feedback und versucht, eine entsprechende Anpassung vorzunehmen.

Im dritten Szenario stellen die agilen Teammitglieder (ATeM) fest, dass eine Lieferung über das Iterationsziel hinaus erreicht werden kann. In einem Gespräch mit dem Product Owner (PrOw) kann das Team bestätigen, dass die über das Iterationsziel hinaus gelieferten Arbeitselemente tatsächlich die relevantesten sind.

5.1.2.5 Vertrauensvotum

Im Rahmen eines Iterationsplanungsevents werden zahlreiche Aspekte, Meinungen und Anpassungen ausgetauscht und erörtert, wodurch sich die ursprüngliche Strategie und Ausrichtung deutlich verändern können. Daher ist es sehr wichtig, dass sich alle agilen Teammitglieder (ATeM) ein paar Minuten Zeit nehmen, um über die vorgeschlagene Strategie nachzudenken und mitzuteilen, ob sie damit einverstanden sind.

Zur Durchführung des Vertrauensvotums zeigen jedes agile Teammitglied (ATeM) und der Product Owner (PrOw) mit der Hand eine Zahl zwischen 1 und 5 an, die das Vertrauen in den vorgeschlagenen Plan wiedergibt. Eine „1“ bedeutet „überhaupt kein Vertrauen“ und eine „5“ bedeutet „ich bin mir zu 100 % sicher, dass wir die Aufgabe erfolgreich abschließen werden“.

Der Ablauf ist wie folgt:

1. Die agilen Teammitglieder (ATeM) und der Product Owner (PrOw) nehmen sich eine Minute Zeit, um ihr Vertrauen einzuschätzen.

2. Auf drei heben alle gleichzeitig die Hand.
3. Zunächst fordert die Moderatorin oder der Moderator die „5er“ auf, ihre Hände zu senken, danach die „4er“ und schließlich die „3er“.
4. Teammitglieder, die mit „2“ oder „1“ abgestimmt haben, sollten ihre Bedenken und Zweifel erklären.
5. Das gesamte agile Projektkernteam (A-PCT) nimmt daraufhin etwaige erforderliche Anpassungen vor, um diese Bedenken auszuräumen.

Sobald alle Bedenken besprochen wurden und es keine „1er“ oder „2er“ mehr gibt, endet das Vertrauensvotum und damit auch die Iterationsplanung.

5.1.3 Handlungsempfehlungen und Beteiligte

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert die Durchführung der Iterationsplanung und stellt sicher, dass das Team den optimalen Nutzen daraus zieht. Hilft dem Team bei der Einhaltung des Zeitplans und deckt eventuelle Anti-Pattern auf, die den Erfolg der Sitzung gefährden könnten.
Agiles Teammitglied (ATeM)	Ist für die Durchführung der Iterationsplanung zuständig, einschließlich der Einrichtung und Aufstellung der Arbeitselementliste (WIL).
Product Owner (PrOw)	Unterstützt die Durchführung der Iterationsplanung, indem der Produkt Owner (PrOw) ein Iterationsziel vorschlägt und alle erforderlichen Klarstellungen bereitstellt, die von den agilen Teammitgliedern (ATeM) bei der Aufstellung der Arbeitselementliste (WIL) angefordert werden. Ist für das mit dem agilen Projektkernteam (A-PCT) vereinbarte Iterationsziel (Output) verantwortlich.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Die Systemarchitektin bzw. der Systemarchitekt (ArOw) ist ebenfalls ein agiles Teammitglied (ATeM) und unterstützt den Product Owner (PrOw) bei der Definition des Iterationsziels, wenn dieses Enablement-Arbeiten beinhaltet, indem die Systemarchitektin bzw. der Systemarchitekt (ArOw) potenzielle Abhängigkeiten klärt oder architektonische Risiken mindert.
Projektleitung (PM)	Wird über das Iterationsziel und den vom agilen Projektkernteam (A-PCT) für seine Erreichung festgelegten Plan informiert

Handlungsempfehlungen

- Einplanen von Zeit für ungeplante Ereignisse – Da Bugs in der Produktionsumgebung, eine erforderliche Unterstützung oder andere ungeplante Ereignisse nicht vorhergesehen werden können, sollte das agile Projektkernteam (A-PCT) etwas Zeit für den Umgang damit einplanen. Beispielsweise bedeutet das Einplanen von 10 % für ungeplante Ereignisse, dass nach der Berechnung der Verfügbarkeit jedes Teammitglieds zusätzliche 10 % eingeplant werden, wodurch dem agilen Teammitglied (ATeM) weniger Zeit für die Bearbeitung der geplanten Elemente zur Verfügung steht.
- Feststellung der Kapazität der agilen Teammitglieder (ATeM) im Voraus – Wenn jedes Teammitglied seine individuelle Kapazität im Voraus feststellt, kann das Team während der Iterationsplanung Zeit sparen. Dabei sollte die Teamkoordinatorin oder der Teamkoordinator (TeCo) die operativen Regeln für dieses Verfahren klarstellen, insbesondere wenn zusätzliche Zeit für ungeplante Ereignisse erforderlich ist.

- Vorbereitung der Arbeitselementeliste (WIL) – Eine richtig priorisierte Arbeitselementeliste (WIL) kann den Unterschied zwischen einer intensiven, aber effizienten Iterationsplanung und einem enttäuschenden und demoralisierenden Event ausmachen. Es ist sicherzustellen, dass der Product Owner (PrOw) diese Aufgabe mit Unterstützung durch die Teamkoordinatorin oder den Teamkoordinator (TeCo) und die Systemarchitektin oder den Systemarchitekten (ArOw) wahrnimmt.
- Definition der Abnahmekriterien – Für die zuvor von dem Team ausgestalteten und vereinbarten Arbeitselemente müssen Abnahmekriterien existieren. Dies hilft dem Team bei der Festlegung seiner Implementierungsstrategie.
- Vorbereitung von zwei oder drei zusätzlichen Arbeitselementen – Nachdem das Team seine volle Kapazität erreicht und das Vertrauensvotum abgeschlossen hat, sollten noch zwei oder drei weitere Elemente in Arbeitseinheiten zerlegt werden. Sie werden nicht in die Iteration einbezogen, stehen aber zur Bearbeitung bereit, falls das Team zusätzliche Zeit hat oder ein Arbeitselement während der Iteration aufgegeben wird.
- „Schwer zu findende“ Aufgaben – Bei der Zerlegung eines Elements der Arbeitselementeliste (WIL) gibt es häufig einen Moment, in dem das Team Schwierigkeiten hat, weitere Aufgaben zu finden. Wenn dies geschieht, sollte es zum nächsten Arbeitselement übergehen.
- Zurückgreifen auf die Erfahrung aus der Vergangenheit – Dabei geht es darum, die Velocity des Teams in der Vergangenheit für die Feststellung seiner aktuellen Kapazität zu nutzen. Auch wenn PM²-Agile diese Praxis nicht empfiehlt, kann die Verwendung der historischen Velocity des Teams für den Product Owner (PrOw) nützlich sein, um einzuschätzen, wie realistisch das Iterationsziel ist, indem der Product Owner (PrOw) diesen Wert mit der Anzahl der Punkte vergleicht, die durch die in der Iteration vorgesehenen Arbeitselemente erzielt wurden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Iterationsplanung	I	I	S	S/A	S	R

5.2 Daily Stand-up

Ein grundlegender Aspekt der täglichen Routine eines Projekts, das agilen Werten und Prinzipien folgt, ist Kommunikation. Die Prinzipien, die PM²-Agile zugrunde liegen, machen deutlich, dass ohne angemessene Kommunikation jedes Projekt zum Scheitern verurteilt ist. Um ein solches Szenario zu vermeiden, empfiehlt PM²-Agile die Einrichtung eines Mechanismus, um die Ausrichtung des agilen Projektkernteams (A-PCT) auf sein Iterationsziel aufrechtzuerhalten und potenzielle Hindernisse zu identifizieren. Bei PM²-Agile wird dieser Mechanismus als Daily Stand-up bezeichnet. Bei diesem Event überprüfen die agilen Teammitglieder (ATeM) den „Herzschlag“ des Projekts und holen voneinander wechselseitig individuelle Zusagen bis zum nächsten Prüfpunkt (in der Regel 24 Stunden) ein.

Der Daily Stand-up dient weder für Status-Updates noch zur Problemlösung. Aufgeworfene Probleme werden registriert und von den maßgeblichen Teammitgliedern unmittelbar nach dem Meeting erörtert.

5.2.1 Häufigkeit und Dauer

Die erste zu klärende Frage lautet: Wie oft sollte diese Überprüfung stattfinden? Täglich? Einmal pro Woche? Dreimal pro Woche? Theoretisch lautet die Antwort „kommt darauf an“, da die Häufigkeit von

der Dynamik des Teams und der ausgeführten Arbeit abhängig ist. Bei eingebetteten agilen Werten und Prinzipien verlangt diese Dynamik jedoch eine häufigere Überprüfung. Daher empfiehlt PM²-Agile ein Daily Stand-up.

Die Dauer des Events steht mit der Anzahl der an dem Meeting teilnehmenden agilen Teammitglieder (ATeM) im Zusammenhang. Als Faustregel empfiehlt PM²-Agile, jedem agilen Teammitglied (ATeM) sechzig bis neunzig Sekunden zu gewähren, um relevante Informationen weiterzugeben und die Abstimmung mit dem Team zu gewährleisten. Bei einem Team mit zehn agilen Teammitgliedern (ATeM) sollte die Höchstdauer rund fünfzehn Minuten betragen.

5.2.2 Struktur eines Daily Stand-up

Auch wenn der Daily Stand-up ein recht kurzes und einfaches Event ist, ist er genauso wichtig wie andere Meetings im Rahmen von PM²-Agile.

Durch den Daily Stand-up soll sichergestellt werden, dass das agile Projektkernteam (A-PCT) darauf ausgerichtet ist, was gerade getan wird und was getan werden muss, um das Iterationsziel zu erreichen. Da jedes agile Teammitglied (ATeM) Updates in klarer und objektiver Form geben muss, empfiehlt PM²-Agile jedem agilen Teammitglied (ATeM), die folgenden Fragen zu berücksichtigen:

- Woran habe ich seit dem letzten Prüfpunkt auf dem Weg zum Iterationsziel gearbeitet?
- Woran werde ich bis zum nächsten Prüfpunkt arbeiten, um das Iterationsziel zu erreichen?
- Welche Blockaden gibt es bei mir, die uns daran hindern könnten, das Iterationsziel zu erreichen?
- Wie zuversichtlich bin ich, dass wir das Iterationsziel erreichen werden?

Der Daily Stand-up beginnt spontan, wenn ein agiles Teammitglied (ATeM) die Initiative ergreift und damit beginnt, Informationen auszutauschen. Sobald es fertig ist, sollte das nächste Teammitglied folgen. Während des Informationsaustauschs können Themen zur Sprache kommen, die eine ausführlichere Analyse oder Bewertung erfordern, insbesondere wenn das Iterationsziel gefährdet ist. In diesem Fall notiert die Teamkoordinatorin oder der Teamkoordinator (TeCo), welche agilen Teammitglieder (ATeM) teilnehmen sollen und registriert sie für ein Anschlussmeeting. Nachdem alle Beteiligten ihre Informationen ausgetauscht haben, erinnert sie die Teamkoordinatorin oder der Teamkoordinator (TeCo) an ihr Anschlussmeeting, damit sie die erforderlichen Vorkehrungen treffen können. Für die Organisation des Anschlussmeetings sind allein die agilen Teammitglieder (ATeM) zuständig, nicht die Teamkoordinatorin oder der Teamkoordinator (TeCo).

5.2.3 Handlungsempfehlungen und Beteiligte

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert die Durchführung des Daily Stand-up. Stellt sicher, dass jedes agile Teammitglied (ATeM) Gelegenheit hat, zu sprechen und die Handlungsempfehlung beachtet. Unterstützt die Konsolidierung für die Zwecke des während des Meetings anberaumten Anschlussmeetings.
Agiles Teammitglied (ATeM)	Ist für die Durchführung des Daily Stand-up zuständig und stellt Informationen über die ausgeführte Arbeit sowie deren Abstimmung mit dem Rest des Teams und dem Iterationsziel zur Verfügung.
Product Owner (PrOw)	Wird erforderlichenfalls über die Ergebnisse des Meetings informiert.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Fungiert in diesem Fall als reguläres agiles Teammitglied (ATeM).
Projektleitung (PM)	Wird über die Ergebnisse des Daily Stand-up informiert.

Handlungsempfehlungen

- **Selbe Zeit, selber Ort** – Der Daily Stand-up ist Bestandteil der täglichen Routine des agilen Projektkernteams (A-PCT). Sein Abhalten am selben Ort und zur selben Zeit hilft dabei, diese Routine zu etablieren, und ermöglicht es allen Beteiligten, ihre Zeitplanung besser zu strukturieren und zu optimieren.
- **Vorbereitung der Kommunikation** – Auch wenn es sich um ein kurzes Meeting handelt, sollte ein agiles Teammitglied (ATeM) einige wenige Minuten in seine angemessene Vorbereitung investieren.
- **Fokussierung aller Beteiligten** – Wenn ein Meeting nicht vorbereitet wird, kann schnell der Fokus verlorengehen. In diesem Fall sollte die Teamkoordinatorin oder der Teamkoordinator (TeCo) das agile Teammitglied (ATeM) wieder auf Kurs zu bringen.
- **Einsatz eines Kanban-Boards¹² zur Illustration der Kommunikation** – Ausgehend von dem Grundsatz, dass ein Bild mehr sagt als tausend Worte, sollte jedes agile Teammitglied (ATeM) seine Kommunikation nach Möglichkeit (entweder physisch oder digital) mit Elementen des Kanban-Boards gestalten.
- **Nutzung des Parkplatzes zur Steuerung von Aufgaben „in Warteschleife“** – Um die während des Daily Stand-up identifizierten Aufgaben „in Warteschleife“ sichtbar zu machen, kann die Teamkoordinatorin oder der Teamkoordinator (TeCo) eine vordefinierte Fläche auf dem Kanban-Board (den sogenannten „Parkplatz“) nutzen. Der Name des Themas und ein oder zwei Namen von beteiligten Personen sollten auf einem Post-it notiert werden, um Sichtbarkeit zu erlangen. Am Ende des Daily Stand-up sieht das Team, welche Aktivitäten anstehen und wer daran beteiligt ist.
- **Gesunder Menschenverstand** – Auch wenn das Daily Stand-up ein besonderes Meeting ist, sollten die üblichen Regeln des gesunden Menschenverstands gelten. Einige der Grundregeln sind, dass nicht unterbrochen, aufmerksam zugehört und transparent kommuniziert werden sollte.

¹² Weitere Informationen zum Kanban-Board können der Technik „Kanban-Methode“ entnommen werden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Daily Stand-up	–	I	S	I	R	R/A

5.3 Release Planning

Menschen leben heute in einem völlig anderen Rhythmus als vor zwanzig, fünfzehn oder sogar zehn Jahren. Sie verarbeiten jeden Tag eine große Menge an Informationen und treffende Tausende von Entscheidungen. Der Begriff VUCA¹³, der häufiger denn je zur Beschreibung der Welt, in der wir leben, verwendet wird, ist nur eines von vielen Anzeichen dafür, dass sich die Dinge verändert haben.

Gegenwärtig führen technologische, wirtschaftliche, soziale und politische Aspekte dazu, dass das, was gestern wahr war, heute möglicherweise nicht mehr gilt, entweder weil sich etwas verändert hat, nicht klar verstanden wurde oder zu komplex war, um es aufrechtzuerhalten.

Das ist die neue Realität, an der kein Weg vorbeiführt. Außerdem gilt: Organisationen, die sich anpassen können, sind die erfolgreichen.

Bei der Entwicklung einer Lösung sollte ein Team in der Lage sein, auf eine Organisationsstrategie (marketingorientiert, risikoorientiert, chancenorientiert) Bezug zu nehmen, die klar definiert, welche Lösung geliefert wird sowie wann und an wen sie geliefert wird. Da die Strategie das Team während des gesamten Projekts begleitet, ist es von größter Bedeutung, dass sie die Anforderungen des Unternehmens (und der Organisation) widerspiegelt. Dies ist jedoch nur der Anfang. Angesichts der gesamten Dynamik, die die VUCA-Welt kennzeichnet, muss das Unternehmen über Tools verfügen, die es ihm ermöglichen, seine Strategie regelmäßig mit dem agilen Projektkernteam (A-PCT) auszutauschen. An diesem Punkt empfiehlt PM²-Agile den Einsatz des Release Planning.

Das Release Planning ist dazu bestimmt, den Lernschritt des Lean Start-up-Modells von PM²-Agile umzusetzen. Durch die Betrachtung der gesammelten Daten ist es möglich, zu lernen und neue Ideen zu generieren, die die Erstellung der Lösung fördern. Durch die Durchführung von zwei kritischen Schritten des iterativen Lean UX-Prozesses von PM²-Agile – Evaluieren und Lernen sowie Aufnahme von Annahmen und Recherchen – konzentrieren sich das agile Projektkernteam (A-PCT) und alle anderen relevanten Stakeholderinnen und Stakeholder auf die wesentlichen Annahmen, die validiert werden müssen.

Das primäre Ziel des Release Planning ist es sicherzustellen, dass das Unternehmen dem agilen Projektkernteam (A-PCT) laufende Änderungen mitteilt, die sich auf die Lösungsstrategie auswirken. Das Team stellt fest, wie umfangreich und komplex diese Änderungen sind, sodass das Unternehmen entscheiden kann, was entwickelt werden und wann eine Freigabe erfolgen soll.

Wenn man zulässt, dass eine Releasestrategie veraltet, kann dies dazu führen, dass eine Lösung ganz oder teilweise unbrauchbar wird, was entsprechende negative Folgen hat. Die Frage, die man sich im Hinblick auf das Release Planning stellen muss, lautet daher nicht „soll es eingesetzt werden?“, sondern „wie oft soll es durchgeführt werden?“.

¹³ Volatility (Volatilität), Uncertainty (Ungewissheit), Complexity (Komplexität), Ambiguity (Mehrdeutigkeit) – Der Begriff wurde 1991 von den US-Streitkräften als Reaktion auf die extremen Bedingungen in Afghanistan und im Irak eingeführt. Das heutige geschäftliche Umfeld hat sich in einer sehr ähnlichen Weise verändert.

5.3.1 Häufigkeit und Dauer

Für Änderungen an der Releasestrategie einer Lösung kann es verschiedene Gründe oder Rechtfertigungen geben. Sie stehen jedoch immer mit den folgenden Perspektiven im Zusammenhang:

- Top-down – Eine Änderung der Strategie der Organisation als Reaktion auf eine Veränderung in ihrem Umfeld.
- Bottom-up – Auf Basis des Feedbacks des Product Owners (PrOw) oder eines agilen Teammitglieds (ATeM) im Zuge der Weiterentwicklung der Lösung.

Unabhängig von der Perspektive sind beim Release Planning drei Aspekte zu berücksichtigen, um den gewünschten Mehrwert zu gewährleisten:

- Pflege der Arbeitselementeliste (WIL) – Der Product Owner (PrOw) und andere geschäftliche Stakeholderinnen und Stakeholder entscheiden, welche Elemente möglicherweise in die Arbeitselementeliste (WIL) aufgenommen, dort aktualisiert oder von dieser entfernt werden müssen.
- Ausgestaltung der Arbeitselementeliste (WIL) – Der Product Owner (PrOw) und der Rest des agilen Projektkernteams (A-PCT) überprüfen die aus der Pflege der Arbeitselementeliste (WIL) stammenden Informationen und nehmen erforderlichenfalls neue Schätzungen vor. Außerdem führen sie die nächsten Arbeitselemente, an denen das Team arbeiten soll, näher aus und validieren sie, überprüfen Schätzungen usw. Bei Bedarf können zusätzliche Stakeholderinnen und Stakeholder wie die Anforderungsmanagerin oder der Anforderungsmanager (BM) und Vertreter der Nutzergemeinschaft aus dem Anforderungsteam (BIG) hinzugezogen werden.
- Priorisierung der Arbeitselementeliste (WIL) – Der Product Owner (PrOw) und die Stakeholderinnen und Stakeholder überprüfen die Arbeitselementeliste (WIL) und legen die Prioritäten neu fest.

PM²-Agile empfiehlt, diese Abfolge von Schritten mindestens einmal pro Iteration (unter Annahme einer zweiwöchigen Iteration) durchzuführen. Je länger eine Iteration jedoch dauert oder je volatiler der Kontext der Lösung ist, umso häufiger sollte ein Release Planning durchgeführt werden.

Es stellt sich die Frage, ob diese Schritte in Form eines einzigen Events oder mehrerer Events durchgeführt werden sollten. Die Erfahrung zeigt, dass die Durchführung dieser Schritte in Form von getrennten Events zu besseren Ergebnissen führt als die Durchführung in einem einzigen Event.

PM²-Agile gibt keine konkrete Empfehlung hinsichtlich der Dauer, da für jeden Schritt unterschiedliche Tools und Techniken eingesetzt werden können. So sind beispielsweise User Story-Workshops im ersten Schritt recht üblich und können mehrere Stunden dauern. Im Gegensatz dazu steht ein Ansatz, bei dem nur eine Person die Storys identifiziert und den übrigen Stakeholderinnen und Stakeholdern vorstellt. Design Blocks liefern ausgezeichnete Ergebnisse im ersten und zweiten Schritt. Ziel ist es, ein Gleichgewicht zwischen der investierten Zeit und der Zielsetzung des jeweiligen Schrittes zu finden.

5.3.2 Struktur des Release Planning

Das Hauptziel des Release Planning ist die Anpassung der Releasestrategie an die Veränderung des Umfelds der Organisation (Top-down) und des Umfelds der täglichen Arbeit und an das Feedback des Product Owners (PrOw) (Bottom-up).

In einem wirksamen Release Planning müssen die folgenden Aktivitäten berücksichtigt werden:

- Pflege der Arbeitselementeliste (WIL),

- Ausgestaltung der Arbeitselementliste (WIL),
- Priorisierung der Arbeitselementliste (WIL).

Jeder Schritt dient einem bestimmten Zweck, bezieht verschiedene Akteure ein und erzeugt unterschiedliche Ergebnisse. Eine priorisierte Arbeitselementliste (WIL), in der sich die aktuelle Lage und ein gemeinsames Verständnis der Vision widerspiegeln, ermöglicht im Ergebnis eine Aktualisierung des Releaseplans.

In den folgenden Abschnitten werden diese Schritte jeweils ausführlicher beschrieben.

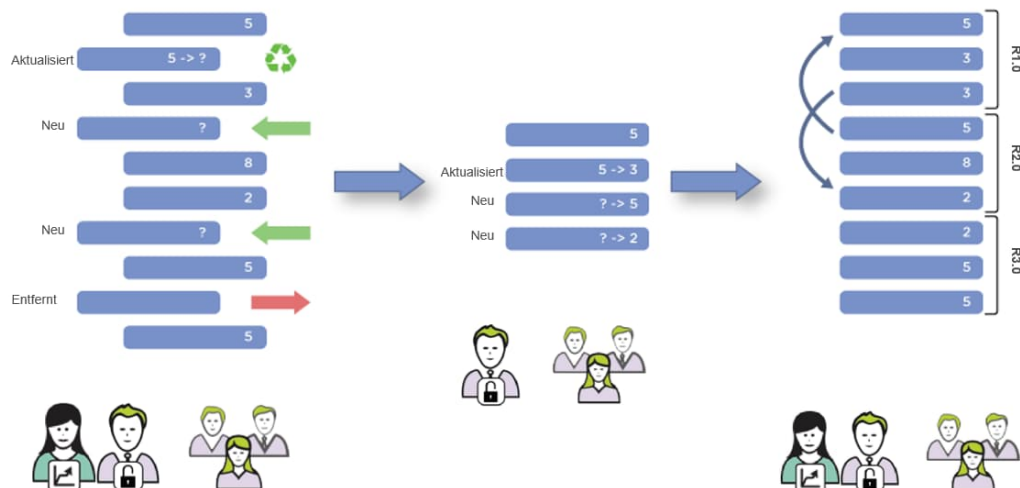


Abbildung 29: Die drei wichtigsten Schritte des Release Planning.

5.3.2.1 Pflege der Arbeitselementliste (WIL)

Die Product Owner bzw. der Product Owner (PrOw) und die Anforderungsmanagerin oder der Anforderungsmanager (BM) sollten ihre Sicht auf Änderungen, bei denen Auswirkungen auf die Arbeitselementliste (WIL) der Lösung anzunehmen sind, abstimmen und austauschen. Beim Lean UX-Prozess beginnt ein neuer Zyklus mit der Aufnahme neuer Annahmen und Hinterfragung der aktuellen Annahmen. Hierdurch wird sichergestellt, dass das agile Projektkernteam (A-PCT) seine Zeit in die Aufgaben mit dem größten Mehrwert investiert.

Bei dieser Aktivität „Pflege der Arbeitselementliste (WIL)“ kann eine Abstimmung zwischen dem Product Owner (PrOw) und der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) erreicht werden. Möglicherweise müssen auch andere Stakeholderinnen und Stakeholder in diese Diskussion einbezogen werden, ohne sich dabei auf die Arbeitselementliste (WIL) zu konzentrieren, sondern vielmehr auf die Funktionen, die zur Erfüllung der identifizierten Anforderungen geliefert werden. Um diesen Prozess zu unterstützen, kann der Product Owner (PrOw) ein separates Event organisieren.

Sobald eine Abstimmung zwischen dem Product Owner (PrOw) und der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) erfolgt ist, arbeiten diese mit dem Anforderungsteam (BIG) zusammen, um festzulegen, wie die vereinbarten Änderungen in der Arbeitselementliste (WIL) konkretisiert werden. Es können neue Elemente aufgenommen werden, andere können entfernt oder auch aktualisiert werden. Ganz egal, was der Fall ist, die Ziele und Abnahmekriterien müssen klar sein.

Als Tool zur Unterstützung dieser Aktivität bringen Design Blocks das agile Projektkernteam (A-PCT) und alle relevanten geschäftlichen Stakeholderinnen und Stakeholder zusammen, um Annahmen zu erörtern und zu hinterfragen, Hypothesen aufzustellen und verschiedene Wege für deren Überprüfung

zu definieren. PM²-Agile empfiehlt den Einsatz von Design Blocks, unterstützt durch kollaboratives Design, einschließlich Designstudios und Designsystemen (Design Systems), um die Abstimmung zu fördern.

PM²-Agile empfiehlt, dass die Teamkoordinatorin oder der Teamkoordinator (TeCo) und ein agiles Teammitglied (ATeM) mit UX Design-Kenntnissen dieses Event unterstützen und an ihm teilnehmen, wenn Tools wie die genannten eingesetzt werden.

Das Endergebnis sollte eine überarbeitete und priorisierte Arbeitselementliste (WIL) sein, die den Erwartungen des Product Owners (PrOw), der Anforderungsmanagerin bzw. des Anforderungsmanagers (BM) und des Anforderungsteams (BIG) entspricht. Sie bildet den Ausgangspunkt für die nächste Aktivität: die Ausgestaltung der Arbeitselementliste (WIL).

5.3.2.2 Ausgestaltung der Arbeitselementliste (WIL)

Nach der Erstellung der Arbeitselementliste (WIL) mit der Anforderungsmanagerin oder dem Anforderungsmanager (BM) und dem Anforderungsteam (BIG) und anderen Stakeholderinnen und Stakeholdern beschreibt der Product Owner (PrOw) den übrigen agilen Teammitgliedern (ATeM) die Änderungen. Im Gegenzug liefert das Team wertvolle Informationen zur Durchführbarkeit und zum Umfang der Änderungen (Schätzungen).

Der Product Owner (PrOw) und die agilen Teammitglieder (ATeM) überprüfen zudem die Arbeitselemente, bei denen erwartet wird, dass sie Teil der nächsten Iteration sein werden, und stellen sicher, dass alle Beteiligten zu einem gemeinsamen Verständnis und einer Einigung gelangen. Design Blocks spielen bei dieser Aktivität erneut eine zentrale Rolle.

Zusammengefasst sind die beiden wesentlichen Aktivitäten einer Sitzung zur Ausgestaltung der Arbeitselementliste (WIL):

- Erörterung der relevantesten Elemente der Arbeitselementliste (WIL),
- Analyse und Überprüfung von Schätzungen.

Es muss festgestellt werden, welche die relevantesten Elemente der Arbeitselementliste (WIL) sind. Als Erstes werden die Elemente betrachtet, die die höchste Priorität besitzen und in die nächste Iterationsplanung aufgenommen werden, als Zweites die Elemente, die in der vorangegangenen Sitzung zur Pflege der Arbeitselementliste (WIL) überprüft wurden.

Sobald die Arbeitselemente identifiziert wurden, sollte der Product Owner (PrOw) sie mit dem Rest des agilen Projektkernteams (A-PCT) besprechen.

Die Arbeitselemente, bei denen erwartet wird, dass sie Teil der nächsten Iterationsplanung sein werden, sollten um weitere Details ergänzt werden. Diese Aktivität ist eine sehr wichtige Voraussetzung für ein erfolgreiches Iterationsplanungsmeeting, da das Team festlegen muss, wie die Erstellung erfolgen soll.

Im Gegensatz dazu erfordern neue oder überarbeitete Elemente, die erst einige Iterationen später implementiert werden sollen, weniger Details. Nichtsdestotrotz müssen die Informationen klar genug sein, um es den agilen Teammitgliedern (ATeM) zu ermöglichen, das Konzept, die Abhängigkeiten und die logischen Verknüpfungen mit den anderen Elementen der Arbeitselementliste (WIL) zu verstehen.

Diese Implementierungsreihenfolge ist von entscheidender Bedeutung, da es nicht sinnvoll ist, Zeit in nicht prioritäre Elemente zu investieren, deren Implementierung in ferner Zukunft liegt.

Durch die Aktivität „Ausgestaltung der Arbeitselementeliste (WIL)“ wird der Schritt zur Aufnahme von Annahmen und Recherchen des Lean UX-Zyklus von PM²-Agile auf dem Weg zu einer optimalen Lösung umgesetzt. Diese Diskussionen helfen bei der Feststellung, wie die aufgestellten Hypothesen, die einer Überprüfung bedürfen, getestet werden können. Der Einsatz von Design Blocks ist eine gute Möglichkeit, diese Aktivitäten zu unterstützen.

Während der überwiegende Teil des Inputs für die Ausgestaltung der Arbeitselementeliste (WIL) vom Product Owner (PrOw) kommt, kann auch der Rest des agilen Projektkernteam (A-PCT) Elemente benennen, die in die Arbeitselementeliste (WIL) aufgenommen werden sollen. Dies gilt für technische Elemente (Enabler), bei denen die Systemarchitektin bzw. der Systemarchitekt (ArOw) dem Product Owner (PrOw) erklärt, wie wichtig sie sind, um die Implementierung der Gesamtlösung zu unterstützen.

Nach der Erörterung und Klärung aller Arbeitselemente sollte das agile Projektkernteam (A-PCT) eine Überprüfung vornehmen und relative Schätzungen bereitstellen. Dies ermöglicht es dem Product Owner (PrOw), der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) und anderen Stakeholderinnen und Stakeholdern, eine gut fundierte Priorisierung vorzunehmen, die zur nächsten Aktivität überleitet: der Priorisierung der Arbeitselementeliste (WIL).

5.3.2.3 Priorisierung der Arbeitselementeliste (WIL)

Als letzter Schritt im Release Planning-Zyklus passen der Product Owner (PrOw) und alle anderen Stakeholderinnen und Stakeholder die Releasestrategie für die Lösung der Endnutzenden an, indem sie die Arbeitselementeliste (WIL) priorisieren. Diese einfache Aktivität ist das Hauptergebnis des Release Planning.

Ausgehend von den Schätzungen, die von den agilen Teammitgliedern (ATeM) während der Aktivität „Ausgestaltung der Arbeitselementeliste (WIL)“ zur Verfügung gestellt wurden, priorisiert der Product Owner (PrOw) in Abstimmung mit der Anforderungsmanagerin oder dem Anforderungsmanager (BM) und anderen Stakeholderinnen und Stakeholdern die verschiedenen Elemente, überprüft den Inhalt der verschiedenen geplanten Releases und aktualisiert gegebenenfalls den aktuellen Releaseplan.

Während dieser Aktivität untersuchen der Product Owner (PrOw) und die anderen Stakeholderinnen und Stakeholder die von den übrigen agilen Teammitgliedern (ATeM) identifizierten Elemente und achten dabei auf Enabler, die die Integrität und Nutzbarkeit der Lösung beeinträchtigen könnten, wenn sie nicht wie geplant implementiert werden. Da weder der Product Owner (PrOw) noch die übrigen Stakeholderinnen und Stakeholder mit den meisten Enablement-Arbeitselementen vertraut sind, empfiehlt PM²-Agile die Anwesenheit der Systemarchitektin oder des Systemarchitekten (ArOw), um die Priorisierung zu unterstützen.

Am Ende dieser Aktivität wird ein überarbeiteter Releaseplan erstellt und mit dem gesamten Projektkernteam (PCT) geteilt. In Abhängigkeit von den Auswirkungen muss der geänderte Releaseplan möglicherweise vom Projektleitungsausschuss (PSC) überprüft werden, um die Änderungen zu vereinbaren.

5.3.3 Handlungsempfehlungen und Beteiligte

Das Release Planning umfasst drei wichtige Aktivitäten mit unterschiedlichen Zwecken, sowohl im Hinblick auf ihre Zielsetzung als auch im Hinblick auf die Rollen der verschiedenen Beteiligten. Um diese Informationen klarer zu machen, werden die Handlungsempfehlungen und Beteiligten daher im Kontext der jeweiligen Aktivität beschrieben.

5.3.3.1 Pflege der Arbeitselementeliste (WIL)

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert die Durchführung dieser Aktivität, insbesondere falls sie den Einsatz von Tools wie Design Blocks, User Story-Workshops oder User Story-Mapping beinhaltet. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) stellt sicher, dass die Arbeitselementeliste (WIL) am Ende der Aktivität die vereinbarten Änderungen widerspiegelt.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Unterstützt die Durchführung dieser Aktivität, indem die Anforderungsmanagerin bzw. der Anforderungsmanager (BM) erläutert, welche Auswirkungen auf die Arbeitselementeliste (WIL) die Änderungen/Anpassungen auf die Lösung oder Organisation haben können.
Agiles Teammitglied (ATeM)	Wird über den Output dieser Aktivität informiert. Das agile Teammitglied (ATeM) kann diese Aktivität zudem durch die Moderation und Leitung von Design Block-Sessions unterstützen, wenn es über einschlägige Erfahrung verfügt.
Product Owner (PrOw)	Ist für die Durchführung dieser Aktivität zuständig. Erstellt und aktualisiert alle Arbeitselemente, die für die Fertigstellung der Änderungen der Anforderungsmanagerin bzw. des Anforderungsmanagers (BM) benötigt werden, und ist für die Output des Events verantwortlich.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Wird über den Output dieser Aktivität informiert.
Anforderungsteam (BIG)	Die Nutzervertretung (URs), der Teil des Anforderungsteams (BIG) ist, wird von dem Product Owner (PrOw) zu den vorgesehenen Änderungen an der Arbeitselementeliste (WIL) konsultiert.
Projektleitung (PM)	Wird über den Output dieser Aktivität informiert.

Handlungsempfehlungen

- Genaue Beobachtung der Funktionen – Funktionen sind die Antwort auf die zu erfüllenden Anforderungen; sie müssen als Teil der Releasestrategie der Lösung organisiert und priorisiert werden. Hierdurch wird sichergestellt, dass die Aktivität „Pflege der Arbeitselementeliste (WIL)“ auf die globale Strategie ausgerichtet ist.
- Visualisierung – Eine Visualisierung ist bei der Diskussion von Ideen, Annahmen, zu prüfenden Hypothesen usw. von entscheidender Bedeutung. Design Blocks bieten die Umgebung, die Plattform und zusätzliche Tools, um sicherzustellen, dass alle Stakeholderinnen und Stakeholder zu einem gemeinsamen Verständnis und einer Einigung darüber kommen, was, wie und warum getan werden muss.
- Festlegung der Tools im Voraus – Festlegung der verschiedenen Tools, die für die Untersuchung und Weiterentwicklung der Arbeitselementeliste (WIL) verwendet werden können, und Vertrautmachen mit diesen; die Durchführung von Workshop mit Personen, die nicht mit den Konzepten vertraut sind, kann sehr frustrierend sein. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) kann bei der Erläuterung der Funktionsweise der Tools eine wichtige Rolle spielen. Auch agile Teammitglieder (ATeM) mit spezifischen Kenntnissen können wertvolle Unterstützung leisten.

- Vorbereitung der Veranstaltung im Voraus – Diese Events können sehr anstrengend sein und zahlreiche Inhalte abdecken. Recherchen im Vorfeld zu bestimmten Themen werden dringend empfohlen, um den Fokus der Beteiligten während der Sitzung sicherzustellen.
- Kenntnis der Arbeitselementliste (WIL) – Es gehört zur täglichen Arbeit des Product Owners (PrOw), sich mit der Arbeitselementliste (WIL) vertraut zu machen. Die Kenntnis der Arbeitselementliste (WIL) hilft dabei, zu analysieren, ob neue Elemente aufgenommen oder vorhandene Elemente entfernt werden müssen bzw. ob potenzielle Abhängigkeiten zwischen Arbeitselementen bestehen.

RAM (RASCI)	BIG	BM	PM	TeCo	PrOw	ArOw	ATeM
Pflege der Arbeitselementliste (WIL)	C	S	I	S	A/R	I	I/S

5.3.3.2 Ausgestaltung der Arbeitselementliste (WIL)

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert diese Aktivität und stellt dabei sicher, dass das Team alle Elemente versteht und sich über ihre Abnahme verständigt. Unterstützt das Team bei der Durchführung von Schätzungen und stellt sicher, dass dabei die Regeln eingehalten werden.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Wird von dem Product Owner (PrOw) über den Output dieser Aktivität informiert.
Agiles Teammitglied (ATeM)	Unterstützt die Durchführung des Meetings durch das Einbringen von relevanten Daten zu Schätzungen und von neuen Elementen, falls dies erforderlich ist. Ist für den Output der Aktivität verantwortlich. Außerdem unterstützt das Agile Teammitglied (ATeM) diese Aktivität durch die Moderation und Durchführung von Design Blocks-Sessions.
Product Owner (PrOw)	Ist für die Durchführung dieser Aktivität durch eine Erläuterung der relevanten Elemente zuständig.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Leistet als agiles Teammitglied (ATeM) einen Beitrag zu der Aktivität.
Projektleitung (PM)	Wird über den Output dieser Aktivität informiert.

Handlungsempfehlungen

- Teilen aller zu besprechenden Elemente mit allen Beteiligten im Vorfeld – Der Product Owner (PrOw) sollte die Elemente, die ihrer/seiner Meinung nach voraussichtlich besprochen werden, mit dem Rest des agilen Projektkernteam (A-PCT) teilen, um diesem eine Vorbereitung und Fokussierung zu ermöglichen.
- Erläuterung und Erörterung zuerst, Schätzung anschließend – Vor dem Beginn der Schätzung sollte sichergestellt werden, dass alle Elemente in angemessener Weise mit den agilen Teammitgliedern (ATeM) erörtert wurden. Dies fördert das Verständnis und zeigt mögliche Verbindungen und Abhängigkeiten zwischen den Elementen auf, die einen Mehrwert bieten, und ermöglicht relative Schätzungen.

- Visualisierung – Eine Visualisierung ist bei der Diskussion von Ideen, Annahmen, zu prüfenden Hypothesen usw. von entscheidender Bedeutung. Design Blocks bieten die Umgebung, die Plattform und zusätzliche Tools, um sicherzustellen, dass alle Stakeholderinnen und Stakeholder zu einem gemeinsamen Verständnis und einer Einigung darüber kommen, was, wie und warum getan werden muss.
- Präzision bei kurzfristigen Elementen, Prägnanz bei mittel- und langfristigen Elementen – Es sollte mehr Zeit in die Ausgestaltung kurzfristiger Arbeitselemente investiert werden; mittel- und langfristige Elemente sollten dagegen prägnanter, aber klar ausgestaltet werden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Ausgestaltung der Arbeitselementeliste (WIL)	I	I	S	R	I	S/A

5.3.3.3 Priorisierung der Arbeitselementeliste (WIL)

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert auf Aufforderung durch den Product Owner (PrOw) die Durchführung dieser Aktivität.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Unterstützt den Product Owner (PrOw) bei der Festlegung der Prioritäten der Arbeitselementeliste (WIL).
Agiles Teammitglied (ATeM)	Wird von dem Product Owner (PrOw) zur Validierung des neuen Releaseplans konsultiert.
Product Owner (PrOw)	Ist für die Durchführung dieser Aktivität zuständig. Legt zusammen mit der Anforderungsmanagerin oder dem Anforderungsmanager (BM) die Prioritäten der Arbeitselementeliste (WIL) fest. Ist für den Output dieses Events verantwortlich.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Wird während der Aktivität konsultiert, um sicherzustellen, dass alle Fragen und Zweifel zu den Enablement-Elementen der Arbeitselementeliste (WIL) geklärt bzw. ausgeräumt werden.
Anforderungsteam (BIG)	Wird als Nutzervertretung (URs) konsultiert, damit es Input zur Priorisierung der Arbeitselementeliste (WIL) gibt.
Projektleitung (PM)	Wird über den Output dieser Aktivität informiert.

Handlungsempfehlungen

- Berücksichtigung von Abhängigkeiten – Es ist unter Umständen nicht möglich, die Elemente der Arbeitselementeliste (WIL) voneinander unabhängig zu implementieren. In diesem Fall ist die empfohlene Vorgehensweise die Konsultation der Systemarchitektin oder des Systemarchitekten (ArOw), um ein besseres Verständnis der Auswirkungen der Verknüpfungen auf die Priorisierung zu erlangen.
- Finalisierung mit einer klar priorisierten Arbeitselementeliste (WIL) – Abbildung der Releasestrategie im Hinblick auf die Lösung und Vorgabe klarer Handlungsempfehlungen an das agile Projektkernteam (A-PCT), worauf es sich bei der Implementierung des Releaseplans als Nächstes konzentrieren soll.

- Aktualisierung des Releaseplans – Nach der Entscheidung über die Priorisierung müssen die Informationen mit dem Rest des agilen Projektkernteam (A-PCT) zur Aktualisierung/Validierung des Releaseplans unter Berücksichtigung der aktuellen Velocity geteilt werden.

RAM (RASCI)	BIG	BM	PM	TeCo	PrOw	ArOw	ATeM
Priorisierung der Arbeitselemente-liste (WIL)	C	S	I	S	R/A	C	C

5.4 Iterationsreview

Wie in den vorstehenden Abschnitten beschrieben, beginnt jede Iteration mit der Festlegung eines Iterationsziels durch den Product Owner (PrOw). Hierdurch wird der Rest des agilen Projektkernteam (A-PCT) dazu veranlasst, eine Strategie zu entwickeln, von der es glaubt, dass sie ihm bei der Erreichung des Ziels helfen wird. Im Verlauf der Iteration setzt das Team diese Strategie um, indem es alle relevanten Elemente entwickelt.

Leider ist nicht alles schwarz oder weiß, und Missverständnisse sind bei kreativen Tätigkeiten wie der Softwareentwicklung keine Seltenheit. Daher ist es zwingend erforderlich, die von den agilen Teammitgliedern (ATeM) entwickelte Arbeit ständig zu prüfen, um ihre Abstimmung auf die Erwartungen des/der Product Owner (PrOw) zu gewährleisten. In diesem Zusammenhang empfiehlt PM²-Agile den Einsatz des Iterationsreview-Events.

Der Hauptzweck des Iterationsreviews besteht darin, zu beurteilen, ob das Iterationsziel erreicht wurde. Dies kann anhand einer praktischen Demonstration dessen geschehen, was die agilen Teammitglieder (ATeM) entwickelt haben. Die Demonstration ermöglicht es dem Product Owner (PrOw) und anderen Stakeholderinnen und Stakeholdern, Feedback zu geben, und hilft den agilen Teammitgliedern (ATeM) zu beurteilen, ob ihre Arbeit in die richtige Richtung geht.

Daher sollte sich der größte Teil der in die Vorbereitung des Iterationsreviews investierten Zeit auf diese Demo konzentrieren.

In den folgenden Abschnitten wird ausführlich beschrieben, wie ein typisches Iterationsreview-Meeting strukturiert werden muss und wie häufig es stattfinden sollte.

5.4.1 Häufigkeit und Dauer

Der Iterationsreview steht in engem Zusammenhang mit dem in den vorstehenden Abschnitten beschriebenen Iterationsplanungsmeeting und führt die Planungs- und Review-Aktivitäten eines Iterationszyklus zusammen.

Am Ende der Iteration wird in dem Iterationsreview-Meeting bewertet, wie erfolgreich das agile Projektkernteam (A-PCT) bei der Erreichung des Iterationsziels war. PM²-Agile empfiehlt, den Iterationsreview als Abschlussmeeting am letzten Tag der Iteration durchzuführen.

Mehrere Faktoren können Einfluss auf die Dauer des Meetings haben. Eine längere Iteration bedeutet zum Beispiel, dass das Team theoretisch mehr Arbeit entwickeln und liefern kann, was zu einer längeren Demo führt. Wenn mehr Arbeit geliefert wird, steigt auch die Wahrscheinlichkeit, dass mehr Probleme gefunden oder erörtert werden, und damit auch der Zeitaufwand für die Erörterung dieser Probleme.

Auch wenn es keine etablierte Dauer für das Iterationsreview-Meeting gibt, sollte das agile Projektteam (A-PCT) daher während der Demo eine objektive Durchführung anstreben. Ausgehend von der praktischen Erfahrung und unter Annahme einer zweiwöchigen Iteration werden bei PM²-Agile 90 bis 120 Minuten als gute Praxis angesehen.

5.4.2 Struktur eines Iterationsreviews

Gemäß dem agilen Prinzip „Funktionierende Software ist das wichtigste Fortschrittsmaß“ ist das zentrale Element dieses Events eine von den agilen Teammitgliedern (ATeM) durchgeführte Demo, mit der sie den geschäftlichen Stakeholderinnen und Stakeholdern die Erfolge der Iteration vorstellen. Um ein erfolgreiches Event zu gewährleisten, müssen jedoch Vorbereitungen getroffen, Schlussfolgerungen gezogen und verschiedene Maßnahmen als Ergebnis des Iterationsreviews geplant werden.

Um ein effizientes Meeting sicherzustellen, muss der Iterationsreview drei verschiedene Momente umfassen, die sorgfältig vorzubereiten sind:

- Abstimmung/Prüfung von Erwartungen – Kurze zehn- bis fünfzehnminütige Präsentation zur Abstimmung der Erwartungen an den Inhalt der Demo. Diese sollte einen Vergleich zwischen dem, was zugesagt und dem, was geliefert wurde, beinhalten.
- Demo – Unter Einhaltung spezifischer Handlungsempfehlungen stellt entweder der Product Owner (PrOw) oder ein anderes agiles Teammitglied (ATeM) den während der Iteration erzielten Fortschritt vor.
- Nachbereitung und Überprüfung der Arbeitselementliste (WIL) – Nach der Demo sollten Schlussfolgerungen gezogen werden, und im Anschluss daran sollte eine Erörterung der Auswirkungen auf die Arbeitselementliste (WIL) erfolgen.



Abbildung 30: Struktur eines Iterationsreviews

Diese Momente werden in den nachstehenden Abschnitten beschrieben.

5.4.2.1 Abstimmung von Erwartungen

Das agile Projektteam (A-PCT) bemüht sich darum, Erwartungen zu vermeiden, die nicht mit denen der geschäftlichen Stakeholderinnen und Stakeholder abgestimmt sind, insbesondere während eines Iterationsreviews. Eine mangelnde Abstimmung führt zu Frustration, die wiederum zu fehlender Objektivität führt, die das Ergebnis des Meetings beeinträchtigen kann. Daher empfiehlt PM²-Agile, eines der agilen Teammitglieder (ATeM) zu benennen, das einige der Kernpunkte anspricht, die für die Demo wichtig sind. Es handelt sich um eine zehn- bis fünfzehnminütige Präsentation, in der die folgenden Elemente behandelt werden müssen:

- Umfang – Beschreibung des Iterationsziels und der entsprechenden Arbeitselemente, die das agile Projektteam (ATeM) zu Beginn der Iteration zugesagt hat.

- Erreicht – Die Arbeitselemente, von denen das agile Projektkernteam (A-PCT) glaubt, dass es sie abgeschlossen hat, und die es demonstrieren wird.
- Nicht erreicht – Die Arbeitselemente, die das agile Projektkernteam (A-PCT) nicht liefern konnte. Dies kann erörtert werden, sofern die Nichterreichung nicht mit Risiken und Problemen im Zusammenhang steht, die zu einem späteren Zeitpunkt erörtert werden.
- Neue/Aufgegebene Elemente – Diese Informationen müssen unbedingt an die geschäftlichen Stakeholderinnen und Stakeholder weitergegeben werden, vor allem, wenn neue Elemente in die Arbeitselementeliste (WIL) aufgenommen wurden.
- Schwierigkeiten/Hindernisse/Risiken – Diese Elemente sollten kurz benannt werden, und es sollte beschrieben werden, wie sie sich auf das Team und die betroffenen Arbeitselemente auswirken.

Mit dieser kurzen Einführung (die wahrscheinlich Anlass für einige Fragen gibt) stellt das agile Projektkernteam (A-PCT) sicher, dass alle Beteiligten bei dem Iterationsreview-Event am selben Strang ziehen. Durch diese Aktivität wird gewährleistet, dass alle Beteiligten für die Demo bereit sind.

5.4.2.2 Demo der Lösung

Die Demo ist für sich genommen der kritischste Moment dieses Events. Es ist der Zeitpunkt, an dem das agile Projektkernteam (A-PCT) den geschäftlichen Stakeholderinnen und Stakeholdern und dem Product Owner (PrOw) das Ergebnis der Iteration vorstellt.

Um eine Demo ordnungsgemäß zu moderieren und durchzuführen, müssen einige Entscheidungen im Voraus getroffen und vereinbart werden:

- Single-Point-Demo oder Hands-on-Demo – Eine Single-Point-Demo bedeutet, dass der Product Owner (PrOw) oder ein anderes agiles Teammitglied (ATeM) die Demo durchführt, während eine Hands-on-Session den geschäftlichen Stakeholderinnen und Stakeholdern die Möglichkeit bietet, die demonstrierte Lösung „anzufassen“ und zu nutzen. PM²-Agile empfiehlt keine bestimmte Option, da sie von dem Wunsch und Engagement der Stakeholderinnen und Stakeholder abhängig ist. Stakeholderinnen und Stakeholdern die Möglichkeit zum Nutzen und Ausprobieren einer Lösung zu geben, ist für die Durchführung der Demo optimal. Ein Hands-on-Ansatz erfordert jedoch mehr Aufwand bei der Moderation und Durchführung der Demo, da sich alle Beteiligten während der gesamten Sitzung auf das konzentrieren müssen, was für die Validierung erforderlich ist.
- Wer soll die Demo durchführen – Der Product Owner (PrOw) ist eine gute Kandidatin bzw. ein guter Kandidat für die Durchführung der Demo. Der Product Owner (PrOw) verfügt über die entsprechenden Befugnisse und steht in engem Kontakt zu den geschäftlichen Stakeholderinnen und Stakeholdern, die sich möglicherweise wohler fühlen, wenn der Product Owner (PrOw) die Demo durchführt. Es ist jedoch auch ein sinnvoller Ansatz, wenn ein agiles Teammitglied (ATeM) die Demo durchführt, da in diesem Fall der Product Owner (PrOw) für die Unterstützung der Stakeholderinnen und Stakeholder bei der Session zur Verfügung steht.
- Welche Testszenarien sollen verwendet werden – Zur Evaluierung der Demo sollten die während der Entwicklung von den agilen Teammitgliedern (ATeM) erstellten Testfälle genutzt werden. Möglicherweise möchten die geschäftlichen Stakeholderinnen und Stakeholder jedoch auch ihre eigenen Szenarien erstellen und verwenden. In diesem Fall muss das agile Projektkernteam (A-PCT) gewährleisten, dass diese Tests auf die Abnahmekriterien der Storys ab-

gestimmt sind, da sich die Demo an diesen orientiert. Nichtsdestotrotz muss unabhängig davon, welches Szenario verwendet wird, dieses im Vorfeld vereinbart werden, sodass die Moderatorin oder der Moderator das Meeting in angemessener Weise vorbereiten kann.

Wenn diese Elemente richtig definiert und vereinbart sind, sollte es dem agilen Projektkernteam (A-PCT) klar sein, von wem und wie die Demo durchgeführt wird.

Auch bei einem gut definierten Moderationsplan ist eine angemessene Struktur für die Demo wichtig. Da mehrere Elemente präsentiert werden, jeweils mit eigenen Abnahmekriterien und Testfällen und mit unterschiedlichen aufkommenden Fragen, muss das agile Projektkernteam (A-PCT) gewährleisten, dass alle diese Elemente in angemessener Weise demonstriert werden und das Feedback erfasst wird. Um bei der Erreichung dieses Ziels zu helfen, sieht PM²-Agile für jedes demonstrierte Element die folgende Abfolge von Schritten vor:

- Identifizierung und Beschreibung des Elements – Die Person, die die Demo durchführt, beginnt mit einer Vorstellung des zu demonstrierenden Elements. Hierzu gehört eine Beschreibung des Arbeitselements und der entsprechenden Abnahmekriterien. Sie kann mit der Angabe der verschiedenen durchzuführenden Testfälle enden.
- Demo des Elements – Während der Demonstration des Elements sollten die einzelnen Testszenarien klar benannt werden, einschließlich der Abnahmekriterien (sehr wichtig), die durch das Szenario validiert werden sollen. Dies hilft den Stakeholderinnen und Stakeholdern dabei, die Nützlichkeit der durchgeführten Tests klar zu erkennen, und stärkt ihr Vertrauen, wenn sie das Element abnehmen müssen.
- Bitte um Bestätigung des Elements – Wenn alle Abnahmekriterien erfüllt sind, sollte das agile Teammitglied (ATeM) um eine Bestätigung bitten, dass das Element genau das tut, was vereinbart wurde. Mit anderen Worten: Das Element soll als erledigt gekennzeichnet werden. Falls andererseits einige Abnahmekriterien nicht erfüllt sind oder neue Arbeiten zu erledigen sind, sollten diese notiert werden und im Nachbereitungsschritt erörtert werden.

Wie bereits erwähnt wurde, müssen diese Schritte für jedes zu demonstrierende Element wiederholt werden. Sobald alle Elemente erledigt sind, kann das Event mit dem nächsten Schritt fortgesetzt werden: Nachbereitung und Überprüfung der Arbeitselementeliste (WIL).

5.4.2.3 Nachbereitung und Überprüfung der Arbeitselementeliste (WIL)

Während der Demo der Lösung werden zahlreiche Informationen ausgetauscht. Deshalb ist es wichtig, genügend Zeit für die Konsolidierung einzuplanen, da die betreffenden Schlussfolgerungen genutzt werden, um eine ausgestaltete und priorisierte Arbeitselementeliste (WIL) zu führen und das nächste Iterationsplanungsmeeting vorzubereiten.

Nach dem Abschluss der Demo muss ein agiles Teammitglied (ATeM) oder auch die Teamkoordinatorin oder der Teamkoordinator (TeCo) jedes einzelne demonstrierte Element durchgehen und Folgendes prüfen:

- Wurde das Element abgenommen oder nicht? Falls nicht, sollte klar sein, was der Grund war (Abnahmekriterien nicht erfüllt, Bug gefunden usw.).
- Handelt es sich um ein neues Element? Handelt es sich um ein Element, das von der Arbeitselementeliste (WIL) entfernt wird?

- Bei neuen Elementen muss eine Erläuterung für den Product Owner (PrOw) und die Stakeholderinnen und Stakeholder angefordert werden. Dies ermöglicht es den agilen Teammitgliedern (ATeM), während der unmittelbar darauffolgenden Überprüfung der Arbeitselementeliste (WIL) grobe Schätzungen bereitzustellen.

Sobald das agile Projektkernteam (A-PCT) alle Elemente konsolidiert hat, kann es zur Ausgestaltung der Arbeitselementeliste (WIL) auf Grundlage, der bislang in dem Meeting gewonnenen Erkenntnisse übergehen. Diese Aktivität ist recht wichtig, weil durch sie eine Arbeitselementeliste (WIL) entsteht, die im Iterationsplanungsmeeting verwendet werden kann. Deshalb sind die folgenden Punkte durchzugehen:

- Arbeitselemente, die die Abnahmekriterien nicht erfüllt haben – Sollten diese wieder an den Anfang der Arbeitselementeliste (WIL) gesetzt werden, um in die nächste Iteration aufgenommen und in dieser korrigiert werden zu können, oder sollten sie auf eine spätere Iteration verschoben werden?
- Neue Elemente auf der Arbeitselementeliste (WIL) – Die agilen Teammitgliedern (ATeM) müssen anhand der von dem Product Owner (PrOw) und den geschäftlichen Stakeholderinnen und Stakeholdern zur Verfügung gestellten Informationen relative Schätzungen für jedes neu aufgenommene Element vereinbaren (wann immer dies möglich ist). Dies hilft dem Product Owner (PrOw) dabei, diese auf der Arbeitselementeliste (WIL) zu priorisieren.

Je nach Anzahl der neuen Storys und des erhaltenen Feedbacks sind die agilen Teammitglieder (ATeM) möglicherweise nicht dazu in der Lage, ihre Analyse während des Meetings abzuschließen. Wenn sie glauben, dass dies der Fall sein wird, sollten sie sich auf jeden Fall auf die Elemente konzentrieren, die voraussichtlich in der nächsten Iteration bearbeitet werden. Alle übrigen Elemente können nach dem Iterationsreview als Teil des Release Planning für die nächste Iteration behandelt werden.

Nach Abschluss des Iterationsreviews (oder ein paar Stunden danach, wenn die Agilen Teammitglieder (ATeM) nach dem Meeting noch einige Schätzungen vornehmen müssen) muss das agile Projektkernteam (A-PCT) über eine ordnungsgemäß ausgestaltete und priorisierte Arbeitselementeliste (WIL) verfügen, die es als Grundlage für die Planung der nächsten Iteration verwenden kann. Der Erfolg eines Iterationsreviews hängt nicht allein von der Anzahl der genehmigten Arbeitselemente ab. Der beste Erfolgsindikator ist zweifellos ein solides und relevantes Feedback an das Team, das ihm dabei hilft, sich an den Bedürfnissen der Stakeholderinnen und Stakeholder zu orientieren.

5.4.3 Handlungsempfehlungen und Beteiligte

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert – soweit dies von den agilen Teammitgliedern (ATeM) angefordert wird – mehrere Aktivitäten des Events. Die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) stellt sicher, dass die zuvor vereinbarte Struktur des Meetings eingehalten wird, ebenso wie die definierten Timeboxen. Die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) kann die Nachbereitung leiten, indem die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) die Erörterung der einzelnen relevanten Arbeitselemente moderiert.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Wird zu den demonstrierten Elementen und neuen/entfernten Arbeitselementen konsultiert.

Agiles Teammitglied (ATeM)	Ist für den Aufbau und die Durchführung dieses Events zuständig. Das Agile Teammitglied (ATeM) nimmt zu Beginn eine Abstimmung der Erwartungen vor. Das Agile Teammitglied (ATeM) kann die Demo durchführen oder diese Aufgabe auf den Product Owner (PrOw) übertragen. In diesem Fall unterstützt das Agile Teammitglied (ATeM) den Product Owner (PrOw), indem das Agile Teammitglied (ATeM) die übrigen Stakeholderinnen und Stakeholder beim Verfolgen der Demo begleitet. Das Agile Teammitglied (ATeM) veranlasst die Nachbereitungssession und stellt die relativen Schätzungen zur Verfügung.
Product Owner (PrOw)	Ist für den Output des Meetings verantwortlich. Der Product Owner (PrOw) unterstützt die Durchführung des Events, indem der Product Owner (PrOw) Feedback zu den demonstrierten Elementen gibt (sie entsprechend den Abnahmekriterien annimmt oder ablehnt) und beim Aufbau des Events hilft, wenn der Product Owner (PrOw) die Demo durchführt.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Fungiert als agiles Teammitglied (ATeM).
Anforderungsteam (BIG)	Wird zu den demonstrierten Elementen und neuen/entfernten Arbeitselementen konsultiert.
Projektleitung (PM)	Wird über den Output des Events informiert.

Handlungsempfehlungen

- Demonstration der Arbeitselemente an den Product Owner (PrOw) im Vorfeld – Während der Iteration sollten die agilen Teammitglieder (ATeM) dem Product Owner (PrOw) jedes Arbeitselement demonstrieren, sobald es fertig ist. Fertig bedeutet nicht unbedingt perfekt, aber gut genug, um es dem Product Owner (PrOw) zu ermöglichen, das Element zu verstehen und zu beurteilen, ob die Entwicklung in die richtige Richtung geht. Zum Zeitpunkt der Demonstration des Arbeitselements im Iterationsreview ist der Product Owner (PrOw) so bereits mit dem Element vertraut und kann daher die Abnahmekriterien besser validieren.
- Demonstration von technischen Arbeitselementen – Jedes Element, das Teil der Arbeitselementeliste (WIL) ist, muss dem Product Owner (PrOw) vorgeführt werden. Agile Teammitglieder (ATeM) tendieren jedoch dazu, diese Regel zu vergessen, wenn sie technische Arbeitselemente (Enablement-Arbeit) erstellen, hauptsächlich weil diese „unsichtbar“ sind und die Abnahmekriterien von den agilen Teammitgliedern (ATeM) festgelegt wurden. Da diese Art von Arbeit eine wesentliche Rolle für die Lösung insgesamt spielt, ist es sehr wichtig, dass dem Product Owner (PrOw) nicht nur ihre Existenz bewusst ist, sondern der Product Owner (PrOw) auch ihre Auswirkungen auf die Lösung insgesamt versteht. Aus Gründen der Transparenz empfiehlt PM²-Agile dringend, dass diese Elemente ebenfalls im Rahmen des Iterationsreviews vorgeführt werden.
- Kurze und prägnante Demo – Eine Demo ist keine Spielwiese für Stichprobentests (diese sollten während der Iteration erfolgen, wenn Arbeitselemente erstmalig demonstriert werden). Bei einer Demo ist es das Ziel des agilen Projektkerntteams (A-PCT) zu bestätigen, dass eine Reihe von Elementen ordnungsgemäß implementiert wurde, sodass ein Iterationsziel erfüllt werden kann. Um dies zu erreichen, darf sich die Demo ausschließlich auf die Elemente konzentrieren, deren Validierung vereinbart wurde, und auf nichts anderes. Dadurch bleiben der Product Owner (PrOw) und andere Stakeholderinnen und Stakeholder auf das Ziel ausgerichtet. Dafür sind solide Moderationskompetenzen und die Hilfe des gesamten agilen Projektkerntteams (A-PCT) wichtig.

- Erstellung von auf die Abnahmekriterien ausgerichteten Testfällen – Für kurze und prägnante Demos werden aussagekräftige Testfälle benötigt. Dies wird durch eine Entwicklung von Testszenarien auf Basis der zu validierenden Abnahmekriterien erreicht. Dadurch wird sichergestellt, dass die Demo nur die absolut notwendige Zeit in Anspruch nimmt, da sich die Empfängerinnen bzw. Empfänger auf das Wesentliche konzentrieren.
- Offenheit für Änderungen – Für viele Teams ist es schwierig zu akzeptieren, dass Änderungen Teil des Review-Meeting-Prozesses sind. Sie sind nicht nur normal, sondern sogar erwünscht. Änderungen bedeuten, dass das Unternehmen auf der Suche nach der optimalen Lösung ist, was für den Erfolg des Projekts unverzichtbar ist. Wenn andererseits die geschäftlichen Stakeholderinnen und Stakeholder immer wieder Anfragen stellen, die die Richtung der ausgeführten Arbeit ändern, kann dies auf ein systemisches Problem hindeuten, das wahrscheinlich entweder mit einer unklaren Vision der Lösung oder damit zusammenhängt, dass sich der Product Owner (PrOw) nicht genug Zeit für das Team nimmt usw. Was auch immer der Fall ist, die Teamkoordinatorin oder der Teamkoordinator (TeCo) muss eingreifen und die Angelegenheit untersuchen.
- Aktualisierung des Releaseplans – Nach dem Abschluss des Iterationsreviews werden neue Daten zur Velocity des agilen Projektkernteams (A-PCT) generiert. PM²-Agile empfiehlt, diese Daten im Releaseplan zu berücksichtigen, da sie Auswirkungen auf den Inhalt der Liefergegenstände im nächsten Release haben. Diese Aktualisierung sollte von der Teamkoordinatorin oder dem Teamkoordinator (TeCo) vorgenommen und anschließend an den Product Owner (PrOw) kommuniziert werden. Falls das Team jedoch nach dem Iterationsreview eine Iterationsretrospektive durchführt, kann die Aktualisierung im Anschluss erfolgen, da die Retrospektive in der Regel zur Aufnahme neuer Arbeitselemente auf die Arbeitselementeliste (WIL) führt, was Auswirkungen auf den Releaseplan haben kann.

RAM (RASCI)	BIG	BM	PM	TeCo	PrOw	ArOw	ATeM
Iterationsreview	C	C	I	S	A/S	R	R

5.5 Iterationsretrospektive

Die Idee einer Retrospektive ist in der PM²-Methode nicht neu. Beispielsweise zielt eines ihrer wichtigsten Events – die Projektabschlussitzung – darauf ab, die gemachten Erfahrungen und die Empfehlungen für die Zeit nach dem Projekt zu erfassen, um den Teams dabei zu helfen, von der im Rahmen des Projekts gewonnenen Erfahrung zu profitieren. Darüber hinaus weist die PM²-Methode auf die Notwendigkeit hin, diese Verbesserungsmöglichkeiten während des gesamten Projekts zu erfassen und nicht erst am Ende, wenn die meisten von ihnen möglicherweise schon vergessen sind.

Bei PM²-Agile gibt es zwei Hauptelemente, die die Retrospektive einen Schritt weiter vorantreiben. Einerseits findet das Retrospektive-Event (Iterationsretrospektive) häufig statt (am Ende jeder Iteration) und ermöglicht so eine ständige und regelmäßige Bewertung der Leistung des agilen Projektkernteams (A-PCT) durch das Team selbst. Andererseits wartet das agile Projektkernteam (A-PCT) nicht bis zum Ende des Projekts, bis es Anpassungen vornimmt. Verbesserungsmöglichkeiten werden so schnell wie möglich in die Arbeit des Teams integriert und ermöglichen es ihm, sich weiterentwickeln und zu verbessern.

Die Iterationsretrospektive ist Teil der Review-Phase des CIR-Rhythmus, der PM²-Agile unterstützt, und fungiert als Abschluss des Zyklus. Am Ende des Projekts – dem Zeitpunkt, an dem die gemachten Erfahrungen erfasst werden – ist die Datenbank, in der das agile Projektkernteam (A-PCT) die Protokolle der Iterationsretrospektiven aufbewahrt, eine wertvolle Informationsquelle, die die gemachten Erfahrungen und die Empfehlungen für die Zeit nach dem Projekt wesentlich bereichert.

Außerdem ist es wichtig, darauf hinzuweisen, dass im Mittelpunkt des Iterationsretrospektive-Meetings die tatsächlichen Probleme eines Teams stehen. Während dieses Meetings findet das agile Projektkernteam (A-PCT) Lösungen, die implementiert werden können, ohne auf die Genehmigung der Leitung warten zu müssen. Da diese Verbesserungen vom agilen Projektkernteam (A-PCT) selbst ausgewählt und ihm nicht auferlegt werden, fühlen sich die agilen Teammitglieder (ATeM) ihrem Erfolg noch stärker verpflichtet.

Es gibt mehrere Möglichkeiten, eine Iterationsretrospektive zu organisieren und zu strukturieren. PM²-Agile empfiehlt eine Struktur mit hinreichender Flexibilität, um die Dauer und die Aktivitäten, die zum Erreichen des Ziels der einzelnen Schritte des Meetings durchgeführt werden, problemlos anzupassen.

In den folgenden Abschnitten werden die erwartete Dauer und die verschiedenen Schritte der Iterationsretrospektive näher erläutert.

5.5.1 Häufigkeit und Dauer

Die Häufigkeit einer Iterationsretrospektive ist auf die anderen Events, die der Vorbereitung und Evaluierung einer Iteration dienen, abgestimmt. Wie die Iterationsplanung und der Iterationsreview findet die Iterationsretrospektive einmal pro Iteration statt. Da das Ziel dieses Events darin besteht, die Ereignisse während der Iteration zu untersuchen und Verbesserungsmöglichkeiten zu identifizieren, überrascht es nicht, dass dieses Event am Ende der Iteration stattfindet. Dies gestattet es dem agilen Projektkernteam (A-PCT), so viele Daten wie möglich zu erfassen und während des Meetings zu analysieren, um so zur Gewinnung wichtiger Erkenntnisse beizutragen.

Die Dauer der Iterationsretrospektive hängt von mehreren verschiedenen Faktoren ab.

- Größe des Teams – Es liegt auf der Hand, dass mit zunehmender Anzahl der am Iterationsreview teilnehmenden Teammitglieder (ATeM) auch die Anzahl der erfassten Inputs steigt, was wiederum dazu führt, dass mehr Zeit für deren Analyse und Verarbeitung benötigt wird.
- Größe der Iteration – Bei längeren Iterationen ist es wahrscheinlicher, dass eine größere Anzahl von Ereignissen oder anderen Themen vorliegt, die das Team besprechen möchte, was theoretisch bedeutet, dass auch mehr Zeit erforderlich ist.
- Ausmaß von Stress/Reibung – In Abhängigkeit vom Ausmaß an Stress, Kontroversen und anderen belastenden Aspekten muss den agilen Teammitgliedern (ATeM) mehr Zeit gegeben werden, um Dampf abzulassen.

Auch wenn sich andere Faktoren ebenfalls auf die Dauer der Iterationsretrospektive auswirken können, ist es sehr wichtig sicherzustellen, dass die für die einzelnen Abschnitte der Sitzung vorgesehene Zeit eingehalten wird, da die Retrospektive sonst möglicherweise nicht die gewünschten Ergebnisse liefert. PM²-Agile empfiehlt in Abhängigkeit von den beschriebenen Faktoren eine Dauer zwischen 60 und 120 Minuten.

Der nächste Abschnitt enthält ausführliche Informationen zur Struktur der Iterationsretrospektive.

5.5.2 Struktur einer Iterationsretrospektive

Das Hauptziel einer Iterationsretrospektive besteht darin, Teams bei ihrer ständigen Verbesserung zu unterstützen. Damit dies gelingt, muss eine Umgebung geschaffen werden, in der sich alle agilen Teammitglieder (ATeM) sicher und wohl fühlen, um konkrete Daten auszutauschen, Probleme und Verbesserungsmöglichkeiten zu identifizieren und Aktionspläne aufzustellen. Diese Umgebung muss während jeder Iterationsretrospektive geschaffen werden, wozu eine strenge, aber flexible Struktur erforderlich ist. Strenge, weil sie immer eine Abfolge von Schritten beinhalten sollte, und flexibel, damit der Inhalt jedes Schritts angepasst werden kann, ohne dass sich dadurch sein Zweck verändert.

PM²-Agile beschränkt sich nicht darauf festzustellen, was gut gelaufen ist, was verbessert werden muss und was nicht mehr gemacht werden darf, sondern schlägt eine stärker agnostische Struktur vor, die auf den folgenden Schritten basiert:

- Begrüßung – Schaffung einer Umgebung, in der sich alle Beteiligten ermutigt und wohl fühlen, teilzunehmen und Meinungen auszutauschen.
- Erhebung von harten Daten – Erhebung aller relevanten Daten, um zu einem gemeinsamen Verständnis der Realität zu gelangen, auf die sich das Team konzentrieren möchte. Hierzu können Events, Kennzahlen usw. zählen.
- Verstehen von Daten und Generierung von Ideen – Hierbei geht es darum, die Ursachen von Problemen und Erfolgsmuster zu verstehen. Generierung möglicher Lösungen mit mehreren alternativen Szenarien.
- Priorisierung und Auswahl – Da es in der Regel nicht möglich ist, alle Vorschläge zu implementieren, muss das Team priorisieren und entscheiden, welche Vorschläge übernommen werden sollen.
- Abschluss – Ende des Events und Sicherstellung der Nachverfolgung aller Erkenntnisse.

In den folgenden Abschnitten werden diese Schritte jeweils etwas ausführlicher beschrieben.

5.5.2.1 Begrüßung

Auch wenn dieser Schritt vielleicht etwas simpel und nicht so wichtig wie die anderen erscheinen mag, handelt es sich tatsächlich um den wichtigsten Schritt, um eine erfolgreiche Retrospektive zu gewährleisten.

Jedes Mal, wenn die agilen Teammitglieder (ATeM) zusammenkommen, um Themen zu erörtern, die ihre individuelle Leistung beinhalten, ist es sehr wichtig, dass sie sich wohlfühlen und wissen, dass ihre Zeit gut investiert ist.

Der erste wichtige Punkt ist dabei, dafür zu sorgen, dass alle Beteiligten das Gefühl haben, dass ihre Meinung zählt und sie somit kein Problem damit haben, sie zu teilen. Als ausgezeichnete und einfache Übung kann die Teamkoordinatorin oder der Teamkoordinator (TeCo) jedes agile Teammitglied (ATeM) auffordern, in einem Satz zu beschreiben, wie es sich während der Iteration gefühlt hat oder was es von der Iterationsretrospektive erwartet. Dadurch macht die Teamkoordinatorin oder der Teamkoordinator (TeCo) deutlich, dass erwartet wird, dass alle Beteiligten an der Diskussion teilnehmen. Wird diese Übung nicht durchgeführt oder wird es Beteiligten gestattet zu schweigen, bedeutet dies, dass es für sie in Ordnung ist, für den Rest der Retrospektive weiter zu schweigen. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) kann auf andere Aktivitäten zurückgreifen, um dasselbe Ziel zu erreichen.

Nach dieser Aufwärmübung muss die Teamkoordinatorin oder der Teamkoordinator (TeCo) den Beteiligten die Vorgehensweise, d. h. die zu ergreifenden Schritte, die entsprechenden Aktivitäten und was von jeder und jedem von ihnen erwartet wird, erläutern. Schließlich muss die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) die Timebox für das Event kommunizieren.

Der letzte in diesem Schritt zu berücksichtigende Punkt ist eine angemessene Definition von Regeln für das Event. Es sind diese Regeln, die eine Umgebung schaffen, die es den agilen Teammitgliedern (ATeM) erlaubt, sich auch bei schwierigen Gesprächen wohlfühlen. Mit anderen Worten geht es darum, die während des Events akzeptablen Verhaltensweisen zu definieren.

Wenn das agile Projektkernteam (A-PCT) bereits über Arbeitsvereinbarungen verfügt, sollten diese genutzt werden. Schließlich gelten dieselben Grundsätze auch für die Iterationsretrospektive. Falls nicht, sollten diese sofort erstellt werden. Dies nimmt etwas Zeit der betreffenden Iterationsretrospektive in Anspruch, ist jedoch für die folgenden sehr nützlich. Das Team ist jetzt bereit, zum nächsten Schritt überzugehen.

5.5.2.2 Erhebung von harten Daten

Wenn das gesamte agile Projektkernteam (A-PCT) startbereit ist, kommt der Zeitpunkt der Erhebung aller zu analysierenden Daten. Mit diesem Ansatz soll sichergestellt werden, dass alle agilen Teammitglieder (ATeM) fokussiert sind und dieselbe Sicht der Realität haben. Andernfalls hat jedes Mitglied seine eigene Meinung auf Grundlage seiner eigenen Sicht der Daten, die es analysieren möchte.

Als Ausgangspunkt sollte die Teamkoordinatorin oder der Teamkoordinator (TeCo) die übrigen agilen Teammitglieder (ATeM) bei der Erhebung der harten Daten im Zusammenhang mit Ereignissen, Kennzahlen, abgeschlossenen Arbeitselementen usw. unterstützen. Ereignisse können dabei alle Arten von Momenten mit Relevanz für eines der agilen Teammitglieder (ATeM) wie Meetings, wichtige Entscheidungen, Feiern usw. sein. Kennzahlen können Burnup-Charts und Burndown-Charts, Velocity-Charts, der Prozentsatz der fehlerhaften Builds, der Prozentsatz der Testabdeckung usw. sein. Je mehr Daten das agile Projektkernteam (A-PCT) zusammenstellt, umso leichter ist es, zu einer gemeinsamen Sicht der Realität zu gelangen.

Es ist zu beachten, dass die Teamkoordinatorin oder der Teamkoordinator (TeCo) zwar bei Aufgaben helfen kann, jedoch stets das agile Teammitglieder (ATeM) entscheidet, welche Daten es verwenden möchte.

PM²-Agile empfiehlt, dass die Teamkoordinatorin oder der Teamkoordinator (TeCo) und die übrigen agilen Teammitglieder (ATeM) diese harten Daten sichtbar vorhalten, sei es geordnet nach technischem Bereich, Teambereich oder chronologisch in einer Zeitleiste usw. Diese Vorgehensweise erleichtert es, Daten zu visualisieren und Verbindungen zwischen ihnen herzustellen, Muster zu erkennen usw.

Es gibt einen zweiten und sehr wichtigen Aspekt, den die agilen Teammitglieder (ATeM) bei der Analyse der Daten nicht außer Acht lassen dürfen. Dabei geht es um menschliche Gefühle. Sie helfen zu verstehen, was für die einzelnen agilen Teammitglieder (ATeM) wichtig ist und wie es ihnen in bestimmten Momenten ging. Wenn man die Gefühle mit einem Datensatz, einem Muster usw. verknüpft, wird es wesentlich leichter, sie zu verstehen. Dabei kann es um die Suche nach der Ursache eines Problems, das Verstehen eines Musters usw. gehen. Dies ist ein sehr wichtiger Aspekt, da es sich um eine wichtige Voraussetzung für den nächsten Schritt handelt.

5.5.2.3 Verstehen von Daten und Generierung von Ideen

Es ist sehr wichtig, über einen Satz von harten Daten zu verfügen, die von den verschiedenen agilen Teammitgliedern (ATeM) jeweils aus einer einzigartigen Perspektive analysiert werden können. Das Vorliegen von Daten allein reicht jedoch nicht, da Daten eine Bedeutung haben müssen. Das ist der Zeitpunkt, an dem das agile Projektkernteam (A-PCT) zum nächsten Schritt übergehen sollte. Dieser dritte Schritt konzentriert sich auf die Auswertung und das Verstehen der Daten sowie die Generierung von Ideen, die es dem Team ermöglichen, einige der identifizierten Probleme zu lösen, Muster zu verstehen, bestimmte Aspekte zu verbessern usw. Wenn ein Team eine Lösung als Antwort auf ein Problem implementiert, ohne Alternativen in Betracht zu ziehen, ist das Endergebnis in der Regel nicht optimal. Aus diesem Grund spielen das Verstehen der harten Daten und das Generieren von Ideen eine grundlegende Rolle, da das agile Projektkernteam (A-PCT) nur so seine Leistung verbessern kann.

5.5.2.4 Priorisierung und Auswahl

Zu diesem Zeitpunkt sollte dem agilen Projektkernteam (A-PCT) eine Reihe von zu implementierenden Maßnahmen für mögliche Verbesserungen vorliegen. Dies ist der richtige Moment, um diejenigen Maßnahmen zu priorisieren und auszuwählen, an denen das Team in der nächsten Iteration (oder dem nächsten Release) arbeiten möchte. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) sollte das Team zu den Maßnahmen hinführen, die eindeutig den größten Nutzen bringen und deren Umsetzung das Team zusagen kann.

Ein weiterer wichtiger Aspekt ist die Art und Weise, in der die vom Team gewählten Maßnahmen hinsichtlich ihrer Bearbeitung geplant werden. Um sicherzustellen, dass diese Maßnahmen jederzeit sichtbar sind und wie jedes andere Element behandelt werden, empfiehlt PM²-Agile ihre Aufnahme in die Arbeitselementliste (WIL).

Der letzte und wichtige, zu berücksichtigende Punkt in diesem Schritt ist es, sicherzustellen, dass sich für jedes (während der Iterationsretrospektive oder während der Iterationsplanung) geplante Arbeitselement agile Teammitglieder (ATeM) verpflichten, alle erforderlichen Aufgaben auszuführen. Wenn eine Aufgabe dem agilen Projektkernteam (A-PCT), aber keinem agilen Teammitglied (ATeM) zugewiesen wird, wird sie am Ende nicht erledigt.

5.5.2.5 Abschluss

Eine Iterationsretrospektive kann aus psychologischer Sicht herausfordernd sein. Daher sollte sich das Team nicht nur ein paar Minuten Zeit für die Vorbereitung des Meetings nehmen, sondern auch einige Minuten dafür einplanen, einen ordnungsgemäßen Abschluss des Meetings sicherzustellen.

Der erste Aspekt, auf den sich die Teamkoordinatorin oder der Teamkoordinator (TeCo) konzentrieren sollte, ist es, einen verbindlichen Abschluss zu gewährleisten. Die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) darf es nicht gestatten, dass in dieser Phase noch offene Themen und Diskussionen verbleiben.

In dieser Phase entscheidet das Team, wie Diskussionsergebnisse und Erfahrungen festgehalten werden. Neben den Eintragungen in der Arbeitselementliste (WIL) gibt es verschiedene andere Elemente – Boards, Fotos, Ausdrucke digitaler Bilder usw. –, deren Informationen zu erfassen und speichern sind. All dieses Wissen gehört den agilen Teammitgliedern (ATeM), die es sich aneignen und verinnerlichen müssen, und nicht der Teamkoordinatorin oder dem Teamkoordinator (TeCo).

Schließlich muss die Teamkoordinatorin oder der Teamkoordinator (TeCo) das Event abschließen und dabei die gesamte vom agilen Projektkernteam (A-PCT) geleistete Arbeit berücksichtigen, nicht nur

während der Iterationsretrospektive, sondern auch während der gesamten Iteration. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) kann die Retrospektive außerdem einer kurzen Untersuchung und Anpassung unterziehen, um festzustellen, welche Punkte verbessert werden können.

5.5.3 Handlungsempfehlungen und Beteiligte

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt/moderiert die Iterationsretrospektive, indem die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) alle erforderlichen Aktivitäten, die in den einzelnen Schritten des Meetings durchgeführt werden sollen, auswählt, vorbereitet und moderiert.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Wird erforderlichenfalls über den Output der Retrospektive informiert.
Agiles Teammitglied (ATeM)	Ist für die Durchführung aller im Rahmen dieses Events vorgesehenen Aktivitäten zuständig. Hierzu gehört die Erhebung von harten Daten, deren Analyse und die Festlegung von Aktionsplänen. Ist für alle Arbeitselemente und Aktivitäten verantwortlich, die sich als Output dieses Events ergeben.
Product Owner (PrOw)	Wird erforderlichenfalls über den Output der Retrospektive informiert.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Fungiert als agiles Teammitglied (ATeM).
Projektleitung (PM)	Wird über den Output des Events informiert.

Handlungsempfehlungen

- Festlegung von Arbeitsvereinbarungen – Arbeitsvereinbarungen sind für die täglichen Aktivitäten des agilen Projektkernteams (A-PCT) einschließlich der verschiedenen Events sehr wichtig. Dies gilt insbesondere für die Iterationsretrospektive. Aufgrund ihrer Formate und Inhalte werden diese Vereinbarungen sogar noch wichtiger, da sie einen entscheidenden Beitrag zu einem reibungslosen Event leisten.
- Engagierte Moderatorin oder engagierter Moderator – Eine Iterationsretrospektive erfordert die aktive und fokussierte Teilnahme aller agilen Teammitglieder (ATeM). Dies ist mit der gleichzeitigen aktiven und engagierten Teilnahme als Moderatorin oder Moderator nicht vereinbar. Die richtige Moderatorin oder der richtige Moderator ist die Teamkoordinatorin oder der Teamkoordinator (TeCo).
- Versenden der Tagesordnung im Voraus an alle Teilnehmenden – Auch wenn die Tagesordnung im ersten Teil der Iterationsretrospektive erläutert wird, ermöglicht es das Versenden der Tagesordnung im Voraus (einschließlich der Aktivitäten) allen agilen Teammitgliedern (ATeM) zu verstehen, wie sie ihre Zeit investieren werden, und bei Bedarf spezifische Materialien vorzubereiten.
- Vermeiden eines „Niemandlands“ der Zuständigkeiten – Agile Teams neigen dazu, Aktionspunkte zu erstellen, für die sie sich nicht zuständig fühlen, indem sie sie anderen Personen zuweisen.

- Dadurch sind sie nicht mehr dazu in der Lage zu steuern, wie sie besser werden können, und schränken die Retrospektive hinsichtlich ihres Hauptziels, der Verbesserung des agilen Projektkernteams (A-PCT), sehr stark ein.
- Dokumentation aller Erfahrungen, Handlungen und der verantwortlichen Personen – Eines der wichtigsten Ergebnisse jeder Retrospektive ist das Wissen, das das agile Projektkernteam (A-PCT) gewinnt. Da das Gedächtnis jedoch nicht alle Informationen behalten kann, müssen die Outputs in einem Protokoll der Retrospektive (dies kann ein Dokument, ein Wiki usw. sein) dokumentiert werden. Dieses wird bei der nächsten Iterationsretrospektive, jedoch auch am Ende des Projekts, wenn die gemachten Erfahrungen erfasst werden, als Referenz herangezogen. Weitere Informationen zu Projektprotokollen enthält Abschnitt 7.3.2 Projektprotokolle.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Iterationsretrospektive	I	I	S	I	R	R/A

6. Rollen und Zuständigkeiten

Das folgende Diagramm gibt einen Überblick über die PM²-Projektorganisation.

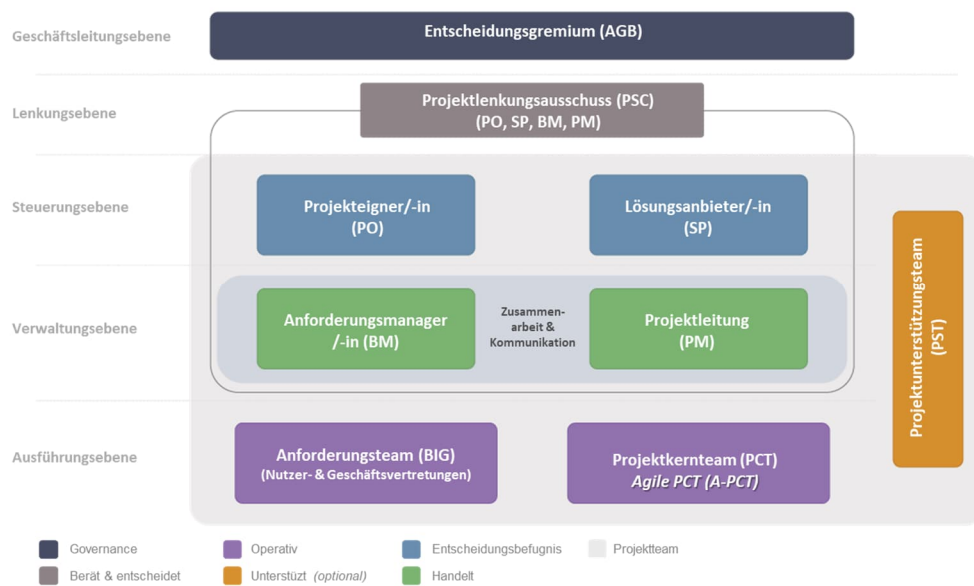


Abbildung 31: PM²-Projektorganisation

Die nachstehende Tabelle enthält eine Kurzbeschreibung der primären Rollen.

Rolle	Kurzbeschreibung
Steuerungsebene	
Projekteignerin bzw. Projekteigner (PO)	Die Projekteignerin bzw. der Projekteigner (PO) ist die wichtigste Entscheidungsinstanz des Projekts und für seinen Erfolg verantwortlich.
Lösungsanbieterin bzw. Lösungsanbieter (SP)	Übernimmt die Gesamtverantwortung für die Liefergegenstände des Projekts.
Verwaltungsebene	
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Beauftragter der Projekteignerin bzw. des Projekteigners (PO), arbeitet mit der Projektleitung (PM) zusammen.
Projektleitung (PM)	Ist für das Gesamtprojekt und seine Liefergegenstände zuständig.
Ausführungsebene	
Projektkernteam (PCT)	Spielt eine Schlüsselrolle bei der Projektdurchführung.
Anforderungsteam (BIG)	Plant und implementiert die geschäftlichen Änderungsaktivitäten.
Nutzerververtretung (URs)	Vertritt die Interessen der Nutzenden im Rahmen des Projekts.

Bei PM²-Agile wird diese Projektorganisation in der Ausführungsebene erweitert. Basierend auf den verschiedenen an IT-Projekten beteiligten Rollen und Zuständigkeiten werden in den folgenden Abschnitten die spezifischen Rollen und Zuständigkeiten im Rahmen von PM²-Agile vorgestellt.

Es folgen einige nützliche Hinweise zur Erinnerung.

- Es gibt nur ein Projektteam, das alle in der Projektorganisation beschriebenen Rollen umfasst.
- Es gibt nur ein Projektkernteam (PCT).
- Ein Team, das den PM²-Agile-Ansatz anwendet, wird als agiles Projektkernteam (A-PCT) bezeichnet.

6.1 PM²-Agile-Projektorganisation

- Teamkoordinatorin bzw. Teamkoordinator (TeCo)
- Product Owner (PrOw)
- Systemarchitektin bzw. Systemarchitekt (ArOw)
- Agiles Teammitglied (ATeM)

Die nachfolgende Abbildung zeigt die Rollen und Zuständigkeiten bei der Organisation und Verwaltung des Projektkernteam im Rahmen des PM²-Agile-Ansatzes.

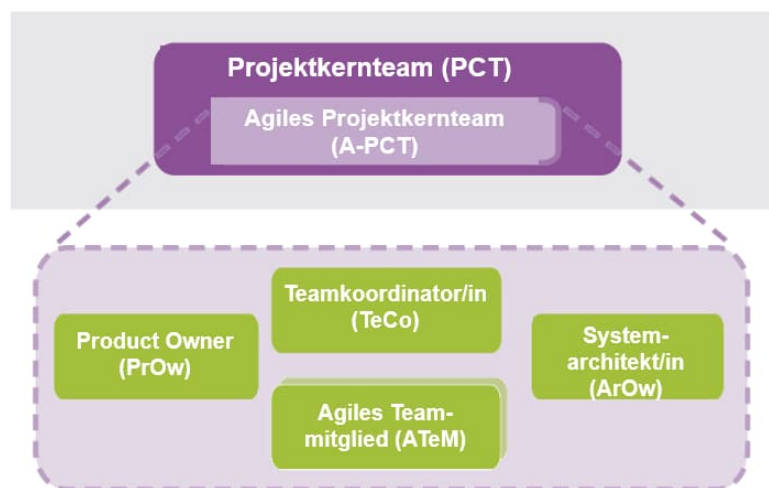


Abbildung 32: Primäre Rollen des agilen Projektkernteam (A-PCT)

6.2 Stakeholderinnen und Stakeholder

Stakeholderinnen und Stakeholder des Projekts sind Personen (oder Gruppen), die Einfluss auf die Aktivitäten während der Dauer eines Projekts und auf den Output oder die Ergebnisse des Projekts haben oder davon beeinflusst werden können. Stakeholderinnen und Stakeholder können unmittelbar an den Projektarbeiten beteiligt oder Mitglieder anderer interner oder externer Organisationen sein (z. B. Auftragnehmer, Lieferanten, Nutzende oder allgemeine Öffentlichkeit).

In Abhängigkeit von der Komplexität und dem Umfang eines Projekts kann es wenige oder viele Stakeholderinnen und Stakeholder geben. Ihre wirksame Einbeziehung ist jedoch ein entscheidender Erfolgsfaktor.

6.3 Kernrollen

In den folgenden Abschnitten werden die Kernrollen und -zuständigkeiten des agilen Projektkernteams (A-PCT) beschrieben.

6.3.1 Teamkoordinatorin oder Teamkoordinator (TeCo)

Die Teamkoordinatorin oder der Teamkoordinator (TeCo) fungiert als Moderatorin bzw. Moderator und Team-Coach, deren bzw. dessen Hauptaufgabe darin besteht, die Bedingungen (z. B. Ressourcen, Lösung von Problemen) zu schaffen und aufrechtzuerhalten, die es dem Team ermöglichen, bestimmte Zielsetzungen zu erreichen und erfolgreich zu sein. Im Mittelpunkt der Zuständigkeiten dieser Rolle stehen typischerweise die methodischen Aspekte der Arbeit und die Abschirmung des agilen Projektkernteams (A-PCT) von Projektmanagementfragen. Die Teamkoordinatorin oder der Teamkoordinator (TeCo) moderiert oder leitet das Team bei der Durchführung von technischen Managementaktivitäten anstatt diese Aufgaben selbst zu übernehmen.

Zuständigkeiten

- Hilft dem agilen Projektkernteam (A-PCT) bei der fortlaufenden Verbesserung seiner Arbeitsweise, um als Team effizienter zu werden.
- Fördert das kollaborative und kooperative Arbeitsumfeld innerhalb des agilen Projektkernteams (A-PCT).
- Unterstützt die Planungs- und Schätzungsaktivitäten des agilen Projektkernteams (A-PCT).
- Ist für die Berichterstattung über den Arbeitsfortschritt (Entwicklungsstatusbericht) an die Projektleitung (PM) zuständig.
- Stellt sicher, dass sich das agile Projektkernteam (A-PCT) voll auf die lieferbezogenen Aktivitäten und das Erreichen der festgelegten spezifischen Ziele konzentrieren kann.
- Unterstützt die Entscheidungsfindung innerhalb des agilen Projektkernteams (A-PCT).
- Arbeitet aktiv an der Identifizierung von Hindernissen, die das Team an der Erreichung der Iterationsziele hindern, und hilft bei deren Überwindung.

6.3.2 Product Owner (PrOw)

Der Product Owner (PrOw) vertritt in erster Linie die Anliegen von Kundinnen und Kunden und Endnutzenden. Diese Rolle sollte ein tiefgehendes Verständnis der Anforderungen und Wünsche der Stakeholderinnen und Stakeholder entwickeln. Dieses Verständnis ermöglicht es dem Product Owner (PrOw), Arbeitselemente zu erfassen und Prioritäten für diese festzulegen. Diese Rolle repräsentiert die „einheitliche Stimme der Stakeholderinnen und Stakeholder innerhalb des agilen Projektkernteams (A-PCT)“ und sollte idealerweise in derselben physischen Umgebung wie der Rest des Teams arbeiten.

Zuständigkeiten

- Führt eine kontinuierliche Priorisierung der vom agilen Projektkernteam (A-PCT) zu bearbeitenden Arbeitselementeliste (WIL) in Abstimmung mit dem Feedback der Stakeholderinnen und Stakeholder und des agilen Projektkernteams (A-PCT) durch.
- Klärt etwaige bereichsbezogene Fragen des agilen Projektkernteams (A-PCT) oder stellt sicher, dass ein offener Kanal zu den relevanten Stakeholderinnen und Stakeholdern für die Zusammenarbeit und Klärung vorhanden ist.
- Unterstützt die Anforderungserhebungs- und Modellierungssessions.

- Stellt sicher, dass die Stakeholdergemeinschaft repräsentiert ist.
- Unterstützt die Präsentation der Zwischen-Outputs des Projekts an die Stakeholdergemeinschaft (Demos).
- Stellt sicher, dass die Stakeholderinnen und Stakeholder den Nutzen des vom agilen Projektkernteam (A-PCT) verfolgten agilen Ansatzes verstehen.

6.3.3 Systemarchitektin oder Systemarchitekt (ArOw)

Die Systemarchitektin bzw. der Systemarchitekt (ArOw) ist die Lösungsarchitektin bzw. der Lösungsarchitekt, die bzw. der für die architektonischen Entscheidungen des agilen Projektkernteam zuständig ist. Die Systemarchitektin bzw. der Systemarchitekt (ArOw) unterstützt die Erstellung und Weiterentwicklung des übergeordneten Lösungsdesigns und beurteilt getätigte oder geplante Investitionen in andere Informationssysteme/Komponenten.

Da die Architektur kritische Projektrisiken verursachen kann, muss sichergestellt werden, dass das Team diese Risiken mindert. Obwohl die Systemarchitektin bzw. der Systemarchitekt (ArOw) in der Regel die leitende Entwicklerin oder der leitende Entwickler im Team ist (und manchmal auch als technische/r Architekt/-in, Softwarearchitekt/-in oder Lösungsarchitekt/-in bezeichnet wird), handelt es sich nicht um eine hierarchische Position, an die andere Teammitglieder berichten. Die Systemarchitektin bzw. der Systemarchitekt (ArOw) sollte einen soliden fachlichen Hintergrund und ein tiefgehendes Verständnis des Geschäftsbereichs besitzen, und von ihr/ihm wird erwartet, wie jedes andere Teammitglied Arbeit abzuzeichnen und zu liefern.

Zuständigkeiten

- Leitet die Erstellung und Weiterentwicklung der Architektur des Informationssystems.
- Unterstützt einen kollaborativen, teambasierten Ansatz und vermeidet es, eine bestimmte architektonische Richtung vorzuschreiben.
- Leitet die technischen Diskussionen, ist jedoch nicht allein für die Architektur zuständig.
- Leitet die anfängliche Architekturplanung zu Beginn des Projekts und unterstützt die anfängliche Anforderungsplanung (insbesondere im Hinblick auf das Verständnis und die Weiterentwicklung der nichtfunktionalen Anforderungen an das Informationssystem). Hierbei liegt der Schwerpunkt auf dem Lebenszyklus des Projekts sowie auf der Entwicklung und Wartbarkeit des Informationssystems.
- Gewährleistet die Abstimmung der Architektur des Informationssystems auf die Handlungsempfehlungen, die Empfehlungen des Systemarchitekturbüros (AO) und die Unterstützung der festgelegten Grundsätze der Unternehmensarchitektur.
- Nutzt in der Organisation getätigte und geplante IT-Investitionen durch die ständige Förderung einer Kultur der Wiederverwendung und Interoperabilität innerhalb des agilen Projektkernteam (A-PCT).
- Leistet einen Beitrag zum Satz an wiederverwendbaren IT-Assets der Organisation, indem der übergeordnete Bereich, der das Informationssystem unterstützen soll, und die Gesamt-IT-Strategie berücksichtigt wird.
- Informiert die Teamkoordinatorin bzw. den Teamkoordinator (TeCo) und die Projektleitung (PM) über die wichtigsten architektonischen Risiken und wirkt an der Festlegung einer geeigneten Risikomanagementstrategie mit.

6.3.4 Agiles Teammitglied (ATeM)

Das agile Teammitglied (ATeM) konzentriert sich auf die Produktion des eigentlichen Informationssystems, das Teil der Lösung des Projekts für die Bedürfnisse der Stakeholderinnen und Stakeholder ist. Diese Rolle umfasst verschiedene Disziplinen der Informationssystementwicklung wie Architektur, Analyse, Design, Programmierung, Testen, Planung und Schätzung. Agile Teammitglieder (ATeM) verfügen über interdisziplinäre Kompetenzen, der Spezialisierungsgrad in den einzelnen Disziplinen variiert jedoch von Person zu Person. Unabhängig von ihrem Hintergrund und ihrer Erfahrung arbeiten agile Teammitglieder (ATeM) mit den übrigen Mitgliedern des agilen Projektkernteams (A-PCT) in einem kompetenzfördernden Umfeld zusammen. Sie können als Generalisten angesehen werden, im Gegensatz zu traditionelleren, hochspezialisierten Rollen wie Entwicklerinnen und Entwicklern, Analytikerinnen und Analytikern und Testerinnen und Testern.

Zuständigkeiten

- Beteiligt sich an der Planung und Schätzung von Iterationen und Releases.
- Beteiligt sich am Design der Lösungsarchitektur.
- Entwickelt gemeinsam mit dem Design der Lösungsarchitektur einen Teil des Informationssystems.
- Testet Entwicklungen.
- Informiert die Teamkoordinatorin oder den Teamkoordinator (TeCo) über die Fortschritte.
- Legt in jeder Iteration die beste Arbeitsweise fest.
- Kommuniziert mit dem Rest des agilen Projektkernteams (A-PCT) und arbeitet mit diesem zusammen.

6.4 Andere Rollen

Neben den bislang beschriebenen PM²-Agile Kernrollen können weitere Rollen Teil des Lebenszyklus des Projekts sein. Typischerweise treten diese anderen Rollen dem agilen Projektkernteam (A-PCT) vorübergehend bei und helfen dem Team bei der Bewältigung und Überwindung bestimmter Herausforderungen, sowohl aus geschäftlicher als auch aus technischer Perspektive.

Fachkundige Person

Der Product Owner (PrOw) gilt als „einheitliche Stimme der Stakeholderinnen und Stakeholder“. Das Einladen fachkundiger Personen, wie etwa Business-Analysten für den Geschäftsbereich, den das Informationssystem unterstützt, hilft dem agilen Projektkernteam (A-PCT) jedoch dabei, den übergeordneten Kontext und den Umfang des Projekts besser zu verstehen.

Technischer Sachverständiger

Das agile Projektkernteam (A-PCT) setzt sich aus Personen mit interdisziplinären Kompetenzen zusammen, die die meisten technischen Herausforderungen während des Projekts bewältigen können. Bei einigen Projekten kann es jedoch hilfreich sein, einen technischen Sachverständigen bzw. eine Sachverständige hinzuziehen, wenn auch nur zeitweise. Technische Sachverständige besitzen spezielles Wissen und spezielle Erfahrung in einem bestimmten Bereich der Softwareentwicklung und helfen dem agilen Projektkernteam (A-PCT) bei der Bewältigung und Überwindung technischer Herausfor-

derungen. Zu den Spezialgebieten können u. a. Softwarearchitektur, Nutzererlebnis (UX), Sicherheit oder Datenbankverwaltung zählen. Beispielsweise kann ein technischer Sachverständiger bzw. eine technische Sachverständige aus dem Systemarchitekturbüro (AO) dem Team bei der Entwicklung einer mit der übergeordneten Unternehmensarchitektur kompatiblen Lösung helfen.

Unabhängige Testerinnen und Tester

Das agile Projektkernteam (A-PCT) ist für die Qualität einer inkrementell und iterativ gelieferten Lösung zuständig. Das Testen der Software ist Teil der Zuständigkeiten der agilen Teammitglieder (ATeM). Einige Organisationen benötigen jedoch eine Validierung der Software des Projekts durch ein Qualitätssicherungsteam, das parallel zum agilen Projektkernteam (A-PCT) arbeitet.

6.5 Die Projektleitung (PM) und die Anforderungsmanagerin bzw. der Anforderungsmanager (BM) in einem PM²-Agile-Projekt

Bei der Einführung von PM²-Agile im Rahmen der PM²-Methodik sind Änderungen im Hinblick auf die Rollen des Governance-Modells zu erwarten. Diese Änderungen spiegeln einen Ansatz wider, der u. a. Aspekte wie dezentrale Entscheidungsfindung, selbst organisierte Teams, ständige Einbeziehung der geschäftlichen Stakeholderinnen und Stakeholder und kontinuierliche Verbesserung unterstützt.

Die Einführung neuer Rollen wie Product Owner (PrOw) und Teamkoordinatorin oder Teamkoordinator (TeCo) und die Anerkennung des Entwicklungsteams als „selbst organisierte Einheit“ hat Auswirkungen auf die PM²-Rollen wie Projektleitung (PM) und Anforderungsmanagerin oder Anforderungsmanager (BM). Diese Auswirkungen sind Ausdruck der Notwendigkeit, die Zuständigkeiten mit den Prinzipien von PM²-Agile in Einklang zu bringen.

6.5.1 Projektleitung (PM)

Im Kontext von der PM²-Methodik hat die Projektleitung (PM) verschiedene Zuständigkeiten, zu denen u. a. die folgenden zählen:

- Einhaltung der Methodik,
- Team- und Fortschrittsmanagement,
- Management der Erwartungen von Stakeholderinnen und Stakeholdern.

Das durch PM²-Agile definierte erweiterte Governance-Modell sieht eine Reihe neuer Rollen vor, die einige üblicherweise der Projektleitung (PM) zugewiesenen Zuständigkeiten übernehmen.

Einhaltung der Methodik

Im Kontext der PM²-Methodik ist es eine der Zuständigkeiten der Projektleitung (PM), das Projektkernteam (PCT) aktiv zur ordnungsgemäßen Anwendung der im Projekthandbuch beschriebenen Methodik zu ermutigen und zu schulen.

Bei der Einführung von PM²-Agile wird diese Zuständigkeit gemeinsam mit der Teamkoordinatorin bzw. dem Teamkoordinator (TeCo) wahrgenommen, die bzw. der gewährleistet, dass alle Aspekte im Zusammenhang mit agilen Werten und Prinzipien in Übereinstimmung mit dem Entwicklungshandbuch eingehalten werden. Beispielsweise ist die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) dafür zuständig, zu ermöglichen und zu gewährleisten, dass das Team die vereinbarten agilen Events (Iterationsplanung, -review, -retrospektive) ordnungsgemäß durchführt. Die Projektleitung (PM)

nimmt an diesen Events als reguläres agiles Teammitglied (ATeM) teil, während die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) als Moderatorin bzw. Moderator fungieren sollte.

Team- und Fortschrittsmanagement

Laut der Definition der PM²-Methode ist die Projektleitung (PM) für die Sicherstellung der effizienten Nutzung von Ressourcen zuständig. Sie weist den einzelnen Teammitgliedern die Arbeit zu und überwacht sie, bis sie abgeschlossen ist.

Bei der Einführung von PM²-Agile führen die agilen Prinzipien von selbst organisierten Teams und dezentraler Entscheidungsfindung dazu, dass das agile Projektkernteam (A-PCT) selbst für die Organisation seiner Arbeit (wie im Entwicklungsarbeitsplan beschrieben) zuständig ist. Dieser Ansatz ermöglicht es der Projektleitung (PM), sich auf die vom agilen Projektkernteam (A-PCT) über die Teamkoordinatorin oder den Teamkoordinator (TeCo) identifizierten Probleme und Hindernisse zu konzentrieren.

Management der Erwartungen von geschäftlichen Stakeholderinnen und Stakeholdern

Bei PM²-Agile wird die neue Rolle des Product Owners (PrOw) zur permanenten Kontaktstelle zwischen der anfordernden und der leistenden Organisation. Im Hinblick auf den Umfang der Lösung und den dazugehörigen Zeitrahmen ist der Product Owner (PrOw) in Zusammenarbeit mit dem agilen Projektkernteam (A-PCT), der Anforderungsmanagerin oder dem Anforderungsmanager (BM) und dem Anforderungsteam (BIG) für die Abstimmung aller Erwartungen zuständig.

Die Projektleitung (PM) muss möglicherweise zusammen mit der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) eingreifen, wenn die Erwartungen nicht abgestimmt sind oder der Product Owner (PrOw) und der Rest des agilen Projektkernteam (A-PCT) nicht zu einem Konsens gelangen können.

6.5.2 Anforderungsmanagerin bzw. Anforderungsmanager (BM)

Laut der Definition der PM²-Methodik spielt die Anforderungsmanagerin oder der Anforderungsmanager (BM) eine zentrale Rolle bei der Koordination der kundenseitigen Aktivitäten und Rollen (z. B. Vertretung der Nutzenden und geschäftlichen Stakeholderinnen und Stakeholder) und stellt sicher, dass die Liefergegenstände des Projekts den Bedürfnissen der geschäftlichen Stakeholderinnen und Stakeholder und der Nutzergemeinschaft entsprechen. Durch die Einführung von PM²-Agile wird der Handlungsspielraum der Anforderungsmanagerin oder des Anforderungsmanagers (BM) deutlich erweitert, da durch die Einbeziehung der neuen Rolle des Product Owners (PrOw) eine permanente und aktive „Kundenstimme“ aufseiten des Leistungserbringers geschaffen wird.

Bindeglied zwischen der Nutzervertretung (URs) und der leistenden Organisation

Der Product Owner (PrOw) treibt die Implementierung der Lösung auf Basis einer priorisierten Arbeitselementeliste (WIL) voran. Die priorisierte Arbeitselementeliste (WIL) spiegelt die Strategie und die Anforderungen der Kundenorganisation wider und ist das Ergebnis der von der Nutzervertretung (URs) und dem Product Owner (PrOw) während der Ausgestaltungssessions geleisteten regelmäßigen Arbeit.

Darüber hinaus stellt der Product Owner (PrOw) sicher, dass die Nutzervertretung (URs) sich aktiv an dem Iterationsreview beteiligt, um den Iterations-Output des Entwicklungsteams zu validieren. Diese wiederkehrenden Aktivitäten machen den Product Owner (PrOw) zur richtigen Person, um bei der Einführung von PM²-Agile als Bindeglied zwischen der Nutzervertretung (URs) und der leistenden Organisation zu fungieren.

Befugnisse und Abstimmung mit dem Product Owner (PrOw)

Da der Product Owner (PrOw) Teil des agilen Projektkerntteams (A-PCT) ist, kommt es zu einer dynamischen und beschleunigten Interaktion zwischen „Geschäft“ und „IT“. Weil der Product Owner (PrOw) die geschäftliche Strategie und Vision für das entwickelte Produkt verkörpert, muss der Product Owner (PrOw):

- von der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) die erforderlichen Befugnisse erhalten, um die notwendigen täglichen Entscheidungen zu treffen, die es dem Entwicklungsteam ermöglichen, Fortschritte auf dem Weg zum Endprodukt zu machen, ohne auf bestimmte Entscheidungen warten zu müssen,
- sich mit der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) und dem Anforderungsteam (BIG) abstimmen, um sicherzustellen, dass die Arbeitselementliste (WIL) die geschäftliche Strategie und Vision widerspiegelt, sodass sich das Team konsistent und regelmäßig auf die wichtigsten Elemente konzentriert.

Ein mit Befugnissen ausgestatteter Product Owner (PrOw), der mit der Anforderungsmanagerin oder dem Anforderungsmanager (BM) und dem Anforderungsteam (BIG) abgestimmt agiert, lässt die wahren Vorteile von PM²-Agile sichtbar werden.

6.6 RAM (RASCI) - Dokumentation von Verantwortlichkeiten

Die Verantwortlichkeitsmatrix (Responsibility Assignment Matrix, RAM) ist eine Methode zur Darstellung und Klärung der Rollen und Verantwortlichkeiten für eine bestimmte Aktivität. Die RAM ist auch als RASCI-Tabelle (ausgesprochen rasky) bekannt. RASCI steht für:

RASCI		Beschreibung
R	Responsible (zuständig)	Erledigen die Arbeit und leiten andere, die unterstützen (support) (die ebenfalls Arbeit erledigen).
A	Accountable (verantwortlich)	Autorisieren und genehmigen Arbeit. Nur eine Person ist verantwortlich (accountable).
S	Supports (unterstützt)	Arbeiten als Mitglieder eines Teams mit der zuständigen Person zusammen. Im Gegensatz zu Rollen mit der Zuständigkeit „wird konsultiert“ (Consulted) helfen die Rollen mit der Zuständigkeit „unterstützt“ (Supports) bei der Erledigung der Aufgabe.
C	Consulted (wird konsultiert)	Geben Input zu der Aktivität.
I	Informed (wird informiert)	Werden informiert (auf aktuellem Stand gehalten).

Dieser Leitfaden enthält eine RAM-(RASCI)-Tabelle für jedes der in Abschnitt 7 vorgestellten Artefakte. Das nachstehende Beispiel zeigt die RAM für die an der Erstellung des Entwicklungshandbuchs beteiligten Rollen.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Entwicklungshandbuch	I	A	R	S	S	S

- Accountable (verantwortlich): Die Projektleitung (PM) ist verantwortlich (stellt sicher, dass die Arbeit erledigt wird).
- Responsible (zuständig): Die Teamkoordinatorin bzw. der Teamkoordinator (TeCo) ist für die Erstellung des Entwicklungshandbuchs zuständig.
- Supports (unterstützt): Der Product Owner (PrOw) und die Systemarchitektin oder der Systemarchitekt (ArOw) arbeiten zusammen mit den agilen Teammitgliedern (ATeM) an der Weiterentwicklung des Entwicklungshandbuchs. Die endgültige Zuständigkeit liegt jedoch in den Händen der Teamkoordinatorin oder des Teamkoordinators (TeCo).
- Informed (wird informiert): Die Anforderungsmanagerin bzw. der Anforderungsmanager (BM) wird informiert, wenn das Dokument fertiggestellt ist.

7. Artefakte

Die Dokumentation, der in der agilen Ebene geplanten und durchgeführten Arbeiten, trägt entscheidend dazu bei, die Transparenz und Koordination zwischen den verschiedenen Ebenen der PM²-Projektorganisation (z. B. zwischen den agilen Steuerungs-, Verwaltungs- und Durchführungsebenen) zu erhöhen.

PM²-Agile wird durch eine Reihe von Artefakten unterstützt, die zur Erfassung und Dokumentation von Informationen über den Managementansatz und zur Berichterstattung über Aktivitäten, Meilensteine, Probleme und Fortschritte des agilen Projektkernteams (A-PCT) dienen.

Diese Artefakte fallen in drei Kategorien: Agile-spezifische Artefakte, Koordinierungs- und Berichtsartefakte und IT-Governance-Artefakte (die für alle IT-Projekte unabhängig vom gewählten Managementansatz gelten).

Art des Artefakts	Beschreibung
IT-Governance-Artefakte	Enthalten die von der IT-Governance der Organisation geforderten Informationen. Sie dokumentieren den Business Case, den Projektauftrag, die Systemarchitekturübersicht und das Geschäftsmodell.
Agile-spezifische Artefakte	Erfassen Informationen zur Planung Agile-spezifischer Prozesse, Aktivitäten, Releases, Iterationen und anderer Meilensteine.
Koordinierungs- und Berichtsartefakte	Erfassen die notwendigen Informationen, um die Aktivitäten des Gesamtprojekts mit denen des agilen Projektkernteams (A-PCT) zu koordinieren und sicherzustellen, dass die Projektleitung (PM) einen Überblick über die bereichsspezifischen Aktivitäten, Probleme, Meilensteine und Fortschritte hat.

Die folgenden Artefakte unterstützen die Projektarbeit auf Grundlage von PM²-Agile:

- Business Case und Projektauftrag
- Systemarchitekturübersicht
- Geschäftsmodell
- Entwicklungshandbuch (wird Bestandteil des gesamten Projekthandbuchs)
- Entwicklungsarbeitsplan (wird Bestandteil des gesamten Projektarbeitsplans)
- Agile-Protokolle
- Bereitstellungsplan
- Testpläne
- Entwicklungsstatusbericht (wird Bestandteil der Projektstatusberichte)

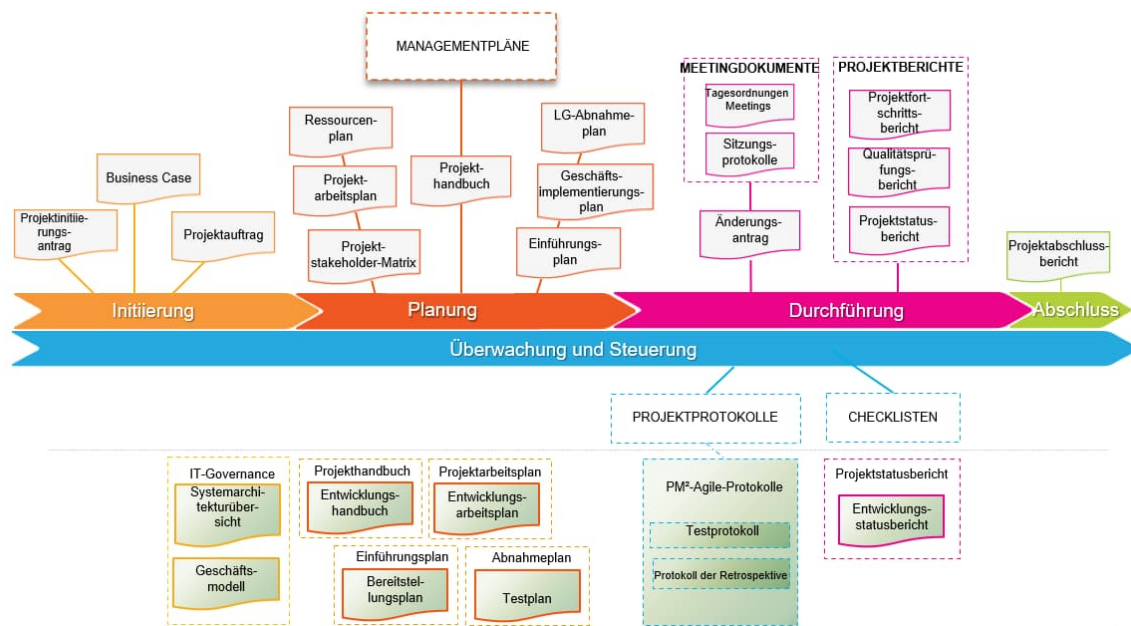


Abbildung 33: (Kombinierte) Artefaktlandschaft der PM²-Methode und PM²-Agile

7.1 IT-Governance-Artefakte

Das Vorhandensein eines Entscheidungsgremiums (AGB) ist ein sehr wichtiger Aspekt, der in der organischen Struktur eines PM²-Projekts vorgesehen ist. Ein wichtiges Ziel der Evaluierung und Governance von Projekten besteht darin sicherzustellen, dass vor Erteilung der Budgetgenehmigung für das Projekt eine Zusammenfassung der Ziele, des Umfangs, des geschätzten Budgets, der Roadmap, der Risiken, der übergeordneten Liefergegenstände, der zentralen Architekturelemente der technischen Lösung und der Hosting-Anforderungen des vorgeschlagenen Projekts vorgelegt wird.

In der Initiierungsphase schlägt das IT-Governance-Gremium einen Mindestsatz von Dokumenten bestehend aus einem Business Case, einem Projektauftrag, einer Systemarchitekturübersicht und einem Geschäftsmodell vor.

Dieser Satz von IT-Governance-Dokumenten zielt darauf ab, die Informationstechnologie-Governance innerhalb der Organisation zu verbessern, indem die getätigten und die geplanten IT-Investitionen ausbalanciert und genutzt werden.

7.1.1 Business Case

Zweck des Business Case ist es, die Beweggründe für das Projekt zu dokumentieren, die Abstimmung auf die strategischen Zielsetzungen der Organisation zu beschreiben, die Investition in Form von Zeit und Arbeit zu rechtfertigen und den Budgetbedarf festzustellen. Bei umfangreichen strategischen Projekten kann der Business Case auch eine Bewertung der Auswirkungen und Risiken sowie eine ausführlichere Kosten-Nutzen-Analyse umfassen.

Der Business Case bietet Entscheidungsträgerinnen und Entscheidungsträgern die erforderlichen Informationen, um zu beurteilen, ob die Durchführung des Projekts sinnvoll ist. Der Business Case ist ein lebendiges Dokument und sollte daher bei kritischen Projekt-Meilensteinen erneut überprüft werden, um zu bestätigen, dass der erwartete Nutzen weiterhin erzielt werden kann, sich die Kosten/Fristen

innerhalb des Budgets/Zeitplans bewegen und das Projekt für die Organisation weiterhin von Bedeutung ist und fortgesetzt werden sollte.

Weitere Informationen über den Business-Case enthalten das PM²-Portal und der Leitfaden. Die nachstehende Tabelle und Abbildung bieten einen Überblick über die an der Erstellung dieses Artefakts beteiligten Rollen.

RAM (RASCI)	AGB	PSC	PO	BM	UR	SP	PM	PCT
Business Case	I	C	A	R	C	S	S	–

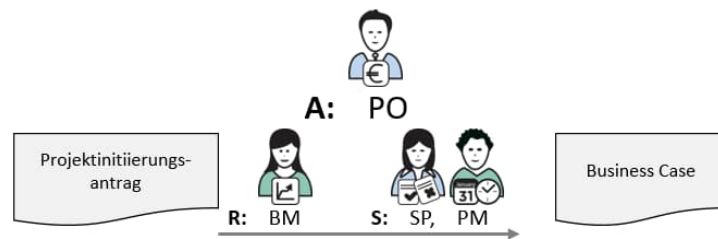


Abbildung 34: Business Case — Input und primäre Rollen

7.1.2 Projektauftrag

Der Projektauftrag bildet die Grundlage für die ausführlichere Projektplanung. Er definiert die Zielsetzungen des Projekts (z. B. Umfang, Zeitrahmen, Kosten, Qualität), die übergeordneten Anforderungen, Risiken und Beschränkungen, Meilensteine und den Liefergegenstand/die Liefergegenstände.

Der Auftrag, der ein wesentliches Element des Projektgenehmigungsverfahrens bildet, enthält die Was-, Wie- und Wann-Grundlagen des Projekts und bietet eine Grundlage für die Messung des Fortschritts. Der Projektauftrag kann zwar von der Anforderungsmanagerin oder dem Anforderungsmanager (BM) initiiert werden, es liegt jedoch letztendlich in der Zuständigkeit der Projektleitung (PM), ihn zu vervollständigen und zur Genehmigung vorzulegen.

Die nachstehende Tabelle und Abbildung bieten einen Überblick über die an der Erstellung dieses Artefakts beteiligten Rollen.

RAM (RASCI)	AGB	PSC	PO	BM	UR	SP	PM	PCT
Projektauftrag	I	A	C	S	C	S	R	C

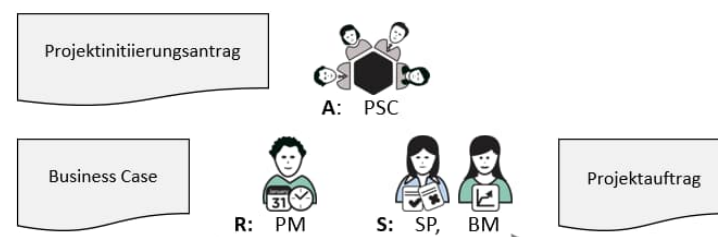


Abbildung 35: Projektauftrag – Input und primäre Rollen

7.1.3 Systemarchitekturübersicht

Die Systemarchitekturübersicht bietet einen groben Überblick über die zu entwickelnde Softwarearchitektur des Informationssystems, einschließlich der wichtigsten Bausteine der Architektur und einer Beschreibung des Beitrags der Architektur zu den Gesamtkapazitäten des Systems.

In der Systemarchitekturübersicht werden die getroffenen kritischen Architekturentscheidungen und die Art und Weise, wie das Informationssystem Compliance-Fragen (z. B. Sicherheit, Datenschutz, Betrugsprävention) berücksichtigt, formell dokumentiert. Ist eine IT-Referenzarchitektur vorhanden, wird in der Systemarchitekturübersicht dokumentiert, wie das Informationssystem die Referenzarchitektur-Leitlinien nutzt, um Konsistenz und Effizienz in der gesamten IT-Landschaft zu gewährleisten. Die Systemarchitekturübersicht enthält außerdem eine Checkliste der häufig in der Organisation verwendeten Komponenten/Dienste, um Fragen der Wiederverwendbarkeit zu berücksichtigen.

Ein wichtiger Aspekt der Systemarchitekturübersicht ist die Kommunikation zwischen den Stakeholderinnen und Stakeholdern, um die Struktur des Informationssystems zu verstehen und die Auswirkungen des gewählten Architekturansatzes zu validieren. Dieses Artefakt ist zudem von entscheidender Bedeutung, wenn Abhängigkeiten zwischen Informationssystemen bestehen, da es einen groben Überblick darüber verschafft, aus welchen Bausteinen das Informationssystem besteht und welche Art von Schnittstellen zur Unterstützung der Interoperabilität möglicherweise vorhanden sind.

Bei der Entwicklung von Lösungen, die auf neuen Ansätzen oder Technologien beruhen, ist die Systemarchitekturübersicht ganz besonders wichtig. Bei Systemen, für die bereits eine gut definierte Architektur vorhanden ist, kann dieses Artefakt weniger ausführlich sein und auf die bestehende Architektur verweisen.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt die Systemarchitektin bzw. den Systemarchitekten (ArOw), indem der Prozess des Einholens erforderlicher Informationen von der Projektebene erleichtert wird.
Agiles Teammitglied (ATeM)	Unterstützt die Definition der Systemarchitekturübersicht und trägt aktiv dazu bei.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Ist für die Erstellung der Systemarchitekturübersicht und die Sicherstellung der Einhaltung der Standards der Organisation in Bezug auf die IT-Architektur (Referenzarchitektur) zuständig.
Product Owner (PrOw)	Wird über die Systemarchitekturübersicht informiert, damit ein allgemeines Verständnis der wichtigsten architektonischen Entscheidungen und Beschränkungen erreicht wird.
Projektleitung (PM)	Die Projektleitung (PM) ist für den Output der Systemarchitekturübersicht verantwortlich und muss die grundlegenden architektonischen Entscheidungen und Beschränkungen verstehen und sicherstellen, dass diese den Stakeholderinnen und Stakeholdern bekannt sind.

Handlungsempfehlungen

- Verständnis des zu lösenden Problems – Es ist von größter Wichtigkeit, vor der Definition einer Lösung und der dazugehörigen Architektur ein klares Verständnis des zu lösenden Problems und der zu erfüllenden Anforderungen zu haben. Der Business Case und der Projektauftrag enthalten diese Informationen.
- Anwendung sowohl interner als auch externer Standards – In der IT-Landschaft gibt es zahlreiche international anerkannte Architekturstandards und Architekturmuster, die bei der Entscheidungsfindung und Definition der Softwarearchitektur helfen können. Zusätzlich kann die Organisation selbst über ihre eigenen Standards verfügen, mit denen das agile Projektkernteam (A-PCT) und insbesondere die Systemarchitektin oder der Systemarchitekt (ArOw) vollständig vertraut sein sollten.
- Bezugnahme auf den bestehenden Komponentenkatalog – Falls ein Systemarchitekturbüro einen Komponentenkatalog führt, sollte seine Nutzung evaluiert und dabei sichergestellt werden, dass die Stakeholderinnen und Stakeholder den Kompromiss zwischen Kosten, Risiko und Funktionalität verstehen und akzeptieren.
- Einsatz verschiedener Modelle – Da die Architektur aus vielen unterschiedlichen Perspektiven betrachtet werden kann, empfiehlt PM²-Agile den Einsatz mehrerer UML-Modelle (Klassen- und Paketdiagramme, Objektdiagramme, Komponentendiagramme usw.).
- Berücksichtigung von Compliance-Fragen – Jede Organisation verfügt über Standards und Richtlinien, die sich auf die Gestaltung der Lösungsarchitektur auswirken können. Daher müssen Compliance-Fragen bei der Architektur des Informationssystems berücksichtigt werden, insbesondere im Hinblick auf Sicherheit, Dokumentenmanagement und Datenschutz.
- Anstreben einer robusten Architektur – Schaffung einer Architektur mit den folgenden Eigenschaften (die nachfolgende Liste ist nicht erschöpfend):
 - konsistent – alle Teile der Architektur harmonisieren, und es existiert ein gemeinsamer Ansatz im Hinblick auf Partitionierung und Zuweisung,
 - lose Verbindung – die Architekturkomponenten sind voneinander unabhängig und lose miteinander verbunden,
 - generisch – die Architekturkomponenten funktionieren in unterschiedlichen Umgebungen und Kontexten,
 - erweiterbar – die Architektur lässt sich erweitern,
 - einfach – eine gute Architektur sollte „so einfach wie möglich, aber nicht einfacher“ sein.
- Übersicht mit gerade ausreichendem Detaillierungsgrad – Es muss sichergestellt werden, dass die Systemarchitekturübersicht in der Anfangsphase detailliert genug ist, um einen Überblick über den technischen Ansatz insgesamt zu vermitteln.
- Aktualisierung der Systemarchitekturübersicht – Mit Voranschreiten des Projekts und zunehmendem Verständnis der Lösung sollte die Systemarchitekturübersicht weiter ausgestaltet werden, um die Weiterentwicklung des Informationssystems in angemessener Weise zu unterstützen. Da die Architektur durch die Entwicklung des Informationssystems „getestet“ wird, muss sichergestellt werden, dass alle Entscheidungen und Änderungen in die Systemarchitekturübersicht aufgenommen werden.

- Unterstützung durch das Geschäftsmodell – Es muss eine Abstimmung zwischen der Systemarchitekturübersicht und dem Geschäftsmodell sichergestellt werden, sodass der „geschäftliche“ Aspekt der Architektur des IT-Systems auf aktuellem Stand ist und die Hosting-Anforderungen identifiziert und gesteuert werden.
- Quelle für Enablement-Arbeit – Da die Systemarchitekturübersicht eine Darstellung der die Lösung unterstützenden Architektur beinhaltet, dient sie als wertvolle Quelle für Enabler, die in die Arbeitselementeliste (WIL) aufgenommen werden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Systemarchitekturübersicht	I	A	S	I	R	S

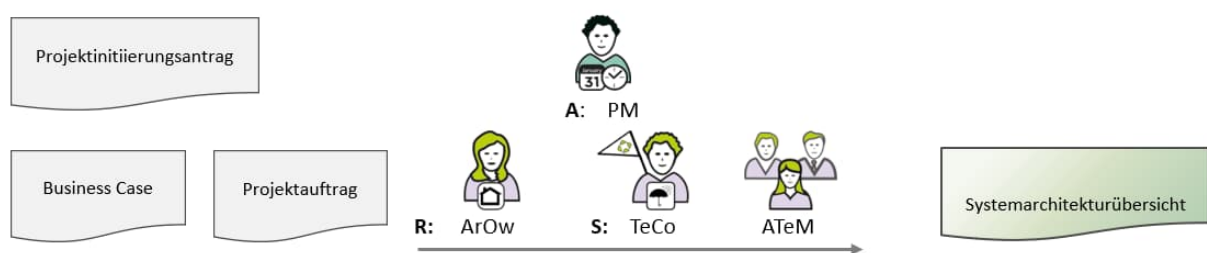


Abbildung 36: Systemarchitekturübersicht – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Systemarchitekturübersicht	Erstellt	Aktualisiert	Aktualisiert	–

7.1.4 Geschäftsmodell

Das Geschäftsmodell beschreibt die „geschäftlichen“ Aspekte der Architektur eines IT-Systems mit Schwerpunkt auf der Infrastruktur des Systems, sowohl aus logischer als auch aus physischer Perspektive. Es beschreibt die erforderlichen geschäftlichen Merkmale und das Netzwerk aus Computersystemen, dazugehörigen Peripheriegeräten, Software, Middleware und Anwendungssoftware der Architektur, die zur Unterstützung der Nutzenden des Systems eingesetzt werden.

Die geschäftliche Verteilung, die in Bereitstellungseinheiten der Komponenten eines Systems erfolgen kann, sollte definiert werden. Die Knoten, die Platzierung von Knoten und Nutzenden an verschiedenen Standorten und die Verbindungen zwischen Knoten, die zur Unterstützung der erforderlichen Interaktionen zwischen den Komponenten benötigt werden, sollten ebenfalls detailliert beschrieben werden, um die funktionalen und nicht-funktionalen Anforderungen des Systems zu erfüllen.

Bei IT-Projekten, die im Rechenzentrum eines Dritten gehostet werden, liefert das Geschäftsmodell einen wichtigen Input, da es für die ordnungsgemäße Unterstützung und Pflege des Systems äußerst wichtig ist.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Leistet Unterstützung, indem der Prozess des Einholens erforderlicher Informationen von der Verwaltungsebene erleichtert wird.
Agiles Teammitglied (ATeM)	Unterstützt die Definition des Geschäftsmodells und trägt aktiv dazu bei.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Ist für die Erstellung des Geschäftsmodells und die Sicherstellung der Einhaltung der Standards der Organisation in Bezug auf IT-Architektur (Referenzarchitektur), Hosting-Leitlinien usw. zuständig.
Product Owner (PrOw)	Wird informiert, damit sowohl aus logischer als auch aus physischer Perspektive ein allgemeines Verständnis der wichtigsten architektonischen Entscheidungen und Beschränkungen sowie der IT-System-Infrastruktur vorhanden ist.
Projektleitung (PM)	Die Projektleitung (PM) ist für den Output des Geschäftsmodells verantwortlich, versteht und vermittelt die wichtigsten Aspekte der benötigten IT-System-Infrastruktur und richtet einen Kommunikationskanal mit den Hosting-Diensten ein.

Handlungsempfehlungen

- Verständnis des zu lösenden Problems – Es muss sichergestellt werden, dass der Projektkontext und die vorgeschlagene Lösung verstanden werden, indem der Business Case und Projektauftrag geprüft werden.
- Aktualisierung des Geschäftsmodells – Mit Voranschreiten des Projekts und zunehmendem Verständnis der Lösung sollte das Geschäftsmodell weiter ausgestaltet werden, um die geschäftlichen Merkmale des IT-Systems näher zu beschreiben. Das Netzwerk aus Computersystemen und dazugehörigen Peripheriegeräten sowie die Systemsoftware, Middleware und Anwendungssoftware, die für die Unterstützung der Nutzenden des Systems erforderlich sind, sollten sowohl aus logischer als auch aus physischer Perspektive auf Architekturebene beschrieben werden.
- Einsatz verschiedener Modelle – Da die IT-Infrastruktur aus unterschiedlichen Perspektiven betrachtet werden kann, empfiehlt PM²-Agile den Einsatz mehrerer UML-Modelle (z. B. Deployment-Diagramme).
- Berücksichtigung von Compliance-Fragen – Ähnlich wie bei der Systemarchitekturübersicht können die Standards und Richtlinien der Organisation Auswirkungen auf das Infrastrukturedesign der Lösung haben. Es muss sichergestellt werden, dass in der Architektur des Informationssystems Compliance-Fragen wie Sicherheit, Dokumentenmanagement, Datenschutz usw. berücksichtigt werden.
- Steuerung der Durchführbarkeit des Geschäftsmodells – Die geschäftliche Komplexität des Systems muss gesteuert und seine Durchführbarkeit sowohl in der Entwicklungs- als auch in der Betriebsphase sichergestellt werden. Dies ist der Grund, warum die Abstimmung zwischen der Systemarchitekturübersicht und dem Geschäftsmodell so eine wichtige Aktivität ist.
- Einbeziehung aller Beteiligten – Es muss sichergestellt werden, dass das gesamte agile Projektkernteam (A-PCT) und die allgemeine Stakeholdergemeinschaft über wichtige geschäftliche Entscheidungen, die Auswirkungen auf die Lösung haben könnten, informiert werden.

RAM (RASCI)	PM	TeCo	PrOw	ArOw	ATeM
Geschäftsmodell	A	S	I	R	S

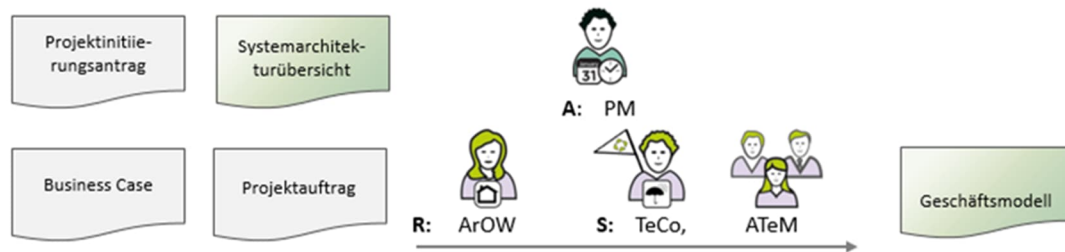


Abbildung 37: Geschäftsmodell – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Geschäftsmodell	Erstellt	Aktualisiert	Aktualisiert	–

7.2 Agile-spezifische Artefakte

PM²-Agile-spezifische Artefakte werden von einem selbst organisierten agilen Projektkernteam (A-PCT) erstellt, in dem alle Mitglieder gleichermaßen zuständig sind. Die Koordinierungsrolle für die einzelnen Artefakte ist sehr wichtig, kann jedoch von einem beliebigen Teammitglied übernommen werden. Für jedes Artefakt existieren allerdings standardmäßige Rollen und Zuständigkeiten. Diese werden dringend empfohlen, um die Koordination mit anderen Teams und der Verwaltungsebene zu erleichtern.

7.2.1 Entwicklungshandbuch

In dem Entwicklungshandbuch werden der für die Entwicklung des Informationssystems (als Teil der Gesamtlösung) ausgewählte Ansatz und die Beziehung zwischen dem bereichsspezifischen Aspekt des Projekts zur Ebene des Gesamtprojekts dokumentiert. Das Entwicklungshandbuch ist eine Erweiterung des Projekthandbuchs und sollte auf die wichtigsten Kontrollprozesse, Projektrichtlinien, Regeln und den übergeordneten Managementansatz, die darin beschrieben sind, abgestimmt sein. Das Entwicklungshandbuch wird zwar als separates Artefakt benannt, muss jedoch kein separater Informationsträger sein. Je nach Art des Projekts kann es einen einzigen Container für beide Artefakte und sogar ein synchronisiertes Konfigurationsmanagement geben.

Das Entwicklungshandbuch erweitert das Projekthandbuch in den folgenden Bereichen:

- Rollen und Zuständigkeiten – Es dokumentiert Abweichungen von den Standardrollen im Rahmen von PM²-Agile und der Standard-RASCI-Tabelle der Zuständigkeiten.
- Projektmanagementpläne – Es gewährleistet, dass wichtige Informationen im Zusammenhang mit dem agilen Entwicklungsaspekt des Projekts in den Managementplänen berücksichtigt und dokumentiert werden. Dies kann etwa der ausgewählte Ansatz für die Steuerung von Risiken sein, die zwar bereichsspezifisch sind, jedoch möglicherweise auf Projektebene sichtbar sind und gesteuert werden müssen.

- Bereichsspezifische Artefakte – Es legt fest, welche Agil-spezifischen Artefakte verwendet werden sollen und dokumentiert etwaige Erwägungen zu individuellen Anpassungen (z. B. für die Umsetzung des Entwicklungsarbeitsplans verwendete Tools).

Das Entwicklungshandbuch und der Entwicklungsarbeitsplan bilden zusammen die Grundlage für die Steuerung der Entwicklungsaktivitäten des Projekts und sind unverzichtbare Referenzmaterialien für alle Projektmitglieder und Stakeholderinnen und Stakeholder, insbesondere für das agile Projektkernteam (A-PCT).

Daher ist das Entwicklungshandbuch ein lebendiges Dokument. Alle Änderungen oder Verbesserungen, die als Ergebnis von Events wie Iterationsretrospektiven oder Iterationsreviews vorgeschlagen werden, sollten darin dokumentiert werden. In der Abschlussphase wird das Entwicklungshandbuch außerdem zu einem unverzichtbaren Bezugspunkt für die Projektabschlusssitzung und sollte ordnungsgemäß abgeschlossen und archiviert werden.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist in enger Abstimmung mit der Projektleitung (PM) für die Erstellung des Entwicklungshandbuchs und die Sicherstellung seiner Ausrichtung an dem Projekthandbuch zuständig.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo), indem zu technischen Fragen beraten wird und technische Beschränkungen aufgezeigt werden, die Auswirkungen auf den zu definierenden Ansatz haben können.
Product Owner (PrOw)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo), indem Erkenntnisse und Informationen über die Realität der Antrag stellenden Organisation bereit gestellt werden, die Auswirkungen auf den zu definierenden Ansatz haben können.
Agiles Teammitglied (ATeM)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo), indem zusätzliche Informationen über die Tools und Techniken bereitgestellt werden, deren Einsatz das Team zur Unterstützung der Entwicklung der Lösung plant.
Projektleitung (PM)	Als verantwortliche Stelle muss die Projektleitung (PM) wissen, wie das agile Projektkernteam (A-PCT) die Steuerung des Entwicklungsaspekts des Projekts plant, und hat sicherzustellen, dass die für das Projekthandbuch relevanten Informationen aus dem Entwicklungshandbuch übernommen werden.

Handlungsempfehlungen

- Nutzung des Projekthandbuchs als Ausgangspunkt – Bei PM²-Projekten ist das Projekthandbuch eine Referenz und der Ausgangspunkt für die Erstellung des Entwicklungshandbuchs. Es muss entschieden werden, ob das Entwicklungshandbuch ein eigenständiger Informationsträger sein oder in das Projekthandbuch eingebettet werden soll.
- Anpassung des Projektlebenszyklus – PM²-Agile führt den CIR-Rhythmus (Coordinate, Implement, Review (Koordinieren, Umsetzen, Überprüfen)) ein. Dadurch, d. h. durch häufige Releases und Iterationen, werden eine Überschneidung der Standardphasengrenzen und mehr Flexibilität ermöglicht.

- Klare Beschreibung der Messung des Projektfortschritts – Da sich die agile Planung deutlich von traditionelleren Ansätzen unterscheidet, ist der Einsatz von PM²-Agile mit einer anderen Art der Messung des Fortschritts als bei Standard-PM²-Projekten verbunden. Diese muss ordnungsgemäß beschrieben werden, wozu auch ein Verweis auf relevante Tools wie die Agile Schätzung gehört.
- Beachtung von Änderungen der zentralen Projektprozesse – Bei PM²-Agile beruht ein Teil der Planung auf einem vollkommen anderen Ansatz im Hinblick auf das Anforderungsmanagement als bei der Standard-PM²-Methodik. Der Einsatz von Storys als Arbeitseinheiten, die den Fortschritt vorantreiben, muss allen Stakeholderinnen und Stakeholdern klar sein. Da bei PM²-Agile das Prinzip „Zusammenarbeit mit Stakeholderinnen und Stakeholdern mehr als Vertragsverhandlung“ gilt, muss zudem der Änderungsmanagement-Prozess angepasst werden, um dieser Agilität Rechnung zu tragen. Andere Prozesse wie Ressourcenmanagement und Kommunikationsmanagement sollten ebenfalls entsprechend angepasst werden.
- Ergänzung neuer PM²-Agile-Rollen und -Zuständigkeiten – Bei PM²-Agile wird das agile Projektkernteam (A-PCT) Teil des Projekts, und mit ihm wird die PM²-Projektorganisation um eine Reihe neuer Rollen und Zuständigkeiten ergänzt. Manche dieser Rollen stehen eng mit anderen Standardrollen in Verbindung und führen zu einer Reorganisation einiger ihrer bestehenden Zuständigkeiten. Diese Änderungen müssen aus dem Entwicklungshandbuch klar hervorgehen.
- Lebendiges Artefakt – Als Teil des CIR-Rhythmus führt der Review-Aspekt zu häufigen Änderungen an den Projektbestandteilen, gleich ob sich diese auf den Prozess, das Team oder die Lösung beziehen. Daher müssen diese Änderungen im Entwicklungshandbuch berücksichtigt werden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Entwicklungshandbuch	I	A	R	S	S	S

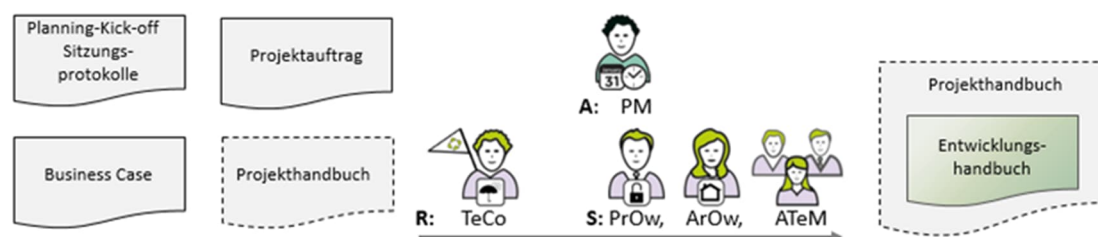


Abbildung 38: Entwicklungshandbuch: Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Entwicklungshandbuch	–	Erstellt	Aktualisiert	–

7.2.2 Entwicklungsarbeitsplan

Im Entwicklungsarbeitsplan werden die relevanten Informationen zusammengestellt, die bei der Steuerung des Entwicklungsaspekts des Projekts helfen. Er wird als Container verschiedener Perspektiven angesehen, was getan werden muss und wie die Entwicklungsziele erreicht werden sollen. Er beinhaltet die Arbeitselementliste (WIL), den entsprechenden Releaseplan und die unterstützenden Iterationen. Diese drei Elemente gestatten es dem agilen Projektkernteam (A-PCT), die Erwartungen der anderen Mitglieder der Projektorganisation zu fixieren und zu verwalten.

Der Entwicklungsarbeitsplan ist ein lebendiges Artefakt, das auf aktuellem Stand gehalten werden muss, indem neue Informationen und der Status des Arbeitsfortschritts eingearbeitet werden, sobald sie verfügbar sind. Er sollte so häufig aktualisiert werden wie nötig, um sich verändernde geschäftliche Prioritäten und Anforderungen, einschließlich Änderungen an der Arbeitselementliste (WIL) oder dem Inhalt des Releaseplans, zu erfassen und aufzunehmen.

Der Entwicklungsarbeitsplan wird Teil des Projektarbeitsplans und sollte daher auf wichtige von der Projektleitung (PM) und dem gesamten Projektkernteam (PCT) festgelegte Projektmeilensteine abgestimmt werden.

Der Entwicklungsarbeitsplan kann unter Verwendung unterschiedlicher Formate und Medien (z. B. MS Office-basierte Dokumentation, Anwendungen wie Atlassian JIRA, IBM Rational Team Concert usw.) dokumentiert werden.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist für die Erhebung von Daten aller Aktivitäten zuständig, die an der Definition des Entwicklungsarbeitsplans im Allgemeinen und des Releaseplans im Besonderen beteiligt sind.
Agiles Teammitglied (ATeM)	Unterstützt die Definition des Entwicklungsarbeitsplans und spielt eine zentrale Rolle bei der Definition der Iterationspläne.
Product Owner (PrOw)	Der Product Owner (PrOw) unterstützt ebenfalls die Definition des Entwicklungsarbeitsplans in seiner Gesamtheit. Der Product Owner (PrOw) ist für die Arbeitselementliste (WIL) zuständig und leistet einen wichtigen Beitrag zur Definition der Release- und Iterationspläne.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo), indem im Kontext der einzelnen Iterationen zu technischen Fragen berät und technische Beschränkungen aufgezeigt werden. Unterstützt den Product Owner (PrOw) bei der Priorisierung der Arbeitselementliste (WIL) und Organisation der Releasepläne.
Projektleitung (PM)	Muss als verantwortliche Stelle wissen, wie das agile Projektkernteam (A-PCT) plant, die Lösung zu liefern, und wie diese Lieferung auf die übergeordneten Projektmeilensteine abgestimmt werden soll.

Handlungsempfehlungen

Der Entwicklungsarbeitsplan besteht aus zwei Teilen:

- Arbeitselementliste (WIL) – Die Arbeitselementliste (WIL) enthält alle Elemente (z. B. User Storys, Bugs, Enabler Storys), die innerhalb des für den IT-Bereich spezifischen Aspekts des Projekts bearbeitet werden müssen, und alle geplanten Arbeiten, die bei dem aktuellen oder zukünftigen Projekten Auswirkungen auf das Informationssystem haben können. Alle Arbeitselemente müssen an einem bestimmten Punkt priorisiert und geschätzt werden, da die Arbeitselementliste (WIL) die wichtigste Projektgrundlage ist.
- Zeitpläne – Zeitpläne existieren auf zwei Detailebenen: ein Zeitplan auf der Ebene von Releases (Releaseplan) und ein Zeitplan auf der Ebene von Iterationen (Iterationsplan). Bei einem Releaseplan liegt der Schwerpunkt hauptsächlich darauf, was getan werden muss und wann. Bei einem Iterationsplan liegt der Schwerpunkt darauf, wie etwas von dem agilen Projektteam (A-PCT) getan werden soll.

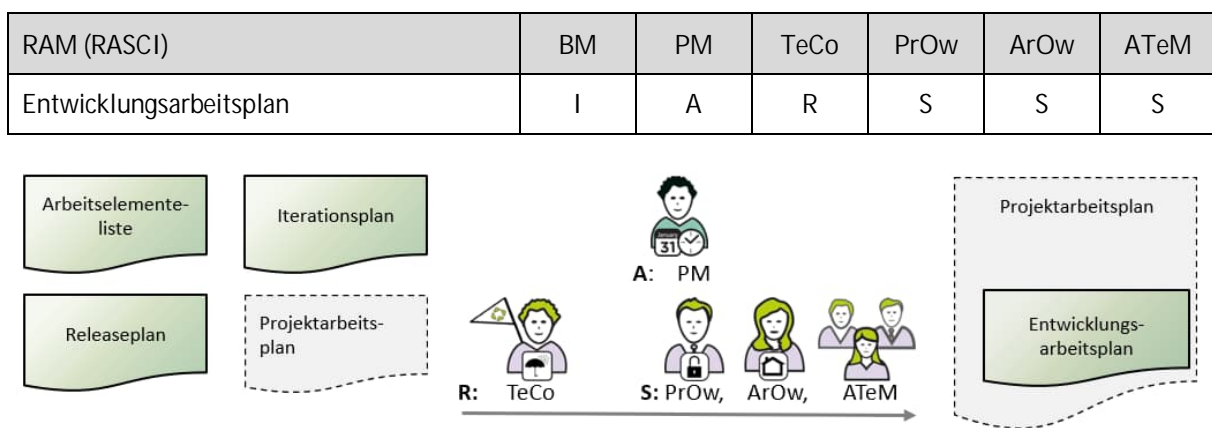


Abbildung 39: Entwicklungsarbeitsplan – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Entwicklungsarbeitsplan	–	Erstellt	Aktualisiert	–

7.2.2.1 Arbeitselementliste (WIL)

Als Teil des Entwicklungsarbeitsplans ist die Arbeitselementliste (WIL) eine Aufstellung der während eines Projekts zu erledigenden Arbeit. Arbeitselemente sollten priorisiert, hinsichtlich ihres Aufwands geschätzt und hinsichtlich ihres Fortschritts nachverfolgt werden.

Die Arbeitselementliste (WIL) enthält typischerweise die folgenden Elemente:

- Fehler (oder Bugs),
- Enabler Storys,
- User Storys.

Die Arbeitselementliste (WIL) wird von dem Product Owner (PrOw) über das gesamte Projekt hinweg gepflegt.

Während jeder Iteration überprüfen der Product Owner (PrOw), das Anforderungsteam (BIG) und die Anforderungsmanagerin bzw. der Anforderungsmanager (BM) die Arbeitselementeliste (WIL), um sicherzustellen, dass in ihr die maßgeblichen zu implementierenden Arbeitselemente berücksichtigt sind. Das Ergebnis dieser Überprüfung wird anschließend mit dem Rest des agilen Projektkernteams (A-PCT) erörtert, sodass einige relative Schätzungen definiert und Abhängigkeiten identifiziert werden können, was dem Product Owner (PrOw) bei der Priorisierung der Arbeitselementeliste (WIL) hilft.

Ein Arbeitselement ist kein Vertrag zwischen dem Entwicklungsteam und dem Product Owner (PrOw), sondern eher ein Hinweis auf etwas, das implementiert werden muss. Daher nimmt in der Regel jedes Arbeitselement für die Durchführung der Arbeiten Bezug auf zusätzliche relevante Informationen (geschäftsbezogene Regelungen, Diagramme, Spezifikationen von Nutzerschnittstellen usw.).

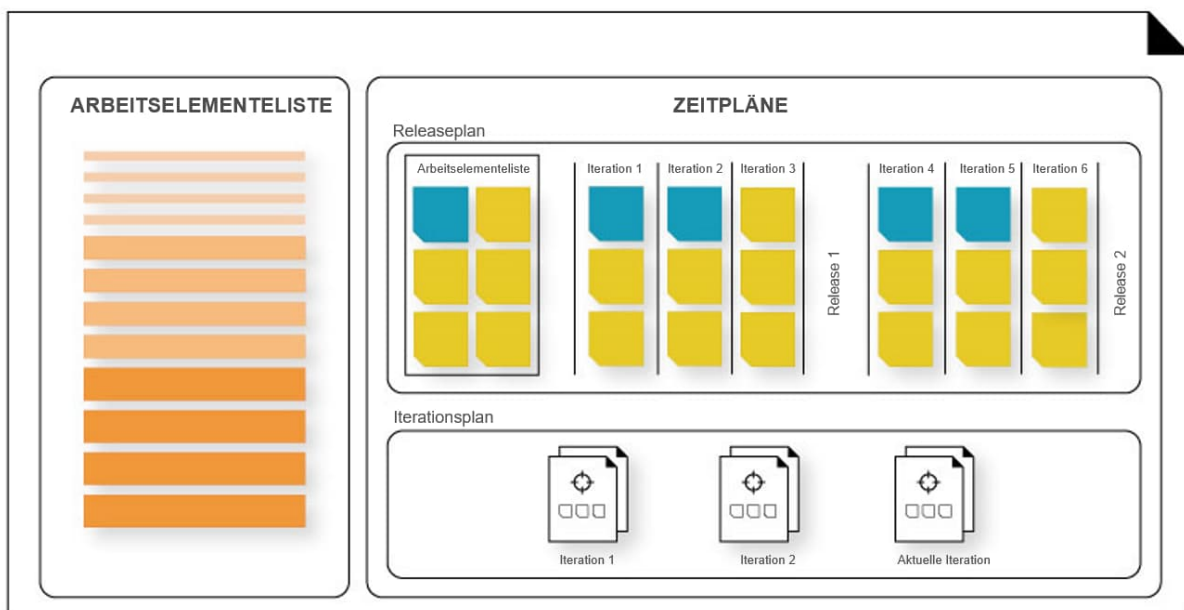


Abbildung 40: Entwicklungsarbeitsplan

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Wird von dem Team und dem Product Owner (PrOw) konsultiert, um zu gewährleisten, dass die Arbeitselementliste (WIL) die erforderliche Konsistenz und Form aufweist, um es dem agilen Projektkernteam (A-PCT) zu ermöglichen, mit ihr zu arbeiten. Wird von der Projektleitung (PM) konsultiert, um die Abstimmung mit dem Projektauftrag zu verifizieren.
Agiles Teammitglied (ATeM)	Unterstützt den Product Owner (PrOw) durch die Bereitstellung von Input zu Schätzungen, Abhängigkeiten usw., die Auswirkungen auf die Organisation und Priorisierung der Arbeitselementliste (WIL) haben können.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Unterstützt den Product Owner (PrOw) durch die Bereitstellung von Input zu Enabler-Arbeitselementen, die geliefert werden müssen, um die geschäftsbezogenen Arbeitselemente zu unterstützen.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Unterstützt als Vertreter der Projekteignerin bzw. des Projekteigners (PO) bei den täglichen Projektaktivitäten den Product Owner (PrOw) durch Input zu den geschäftlichen Prioritäten, Zielsetzungen und Nutzerbedürfnissen.
Product Owner (PrOw)	Ist in enger Abstimmung mit der Anforderungsmanagerin bzw. dem Anforderungsmanager (BM) für die Definition und Pflege der Arbeitselementliste (WIL) zuständig und befugt, alle erforderlichen alltäglichen Entscheidungen des agilen Projektkernteams (A-PCT) zu treffen.
Projektleitung (PM)	Muss als verantwortliche Stelle wissen, wie die Arbeitselementliste (WIL) (als Teil des Entwicklungsarbeitsplans) auf den in dem Projektauftrag definierten Umfang abgestimmt ist.

Handlungsempfehlungen

- Projektauftrag als Ausgangspunkt von Arbeitselementen – Die im Projektauftrag definierte Liste der Features (deren Erstellung eine der ersten Aufgaben bei der Erstellung des Entwicklungsarbeitsplans ist) ist der wichtigste Ausgangspunkt für die Aufnahme von Arbeitselementen in die Arbeitselementliste (WIL).
- Systemarchitekturübersicht und Geschäftsmodell – Diese beiden Artefakte bilden eine wichtige Quelle von Enabler Storys, die die Umsetzung des „geschäftlichen Teils“ der Lösung unterstützen.
- Arbeitsbeginn mit einer priorisierten Arbeitselementliste (WIL) – Die Arbeitselementliste (WIL) muss nicht vollständig, aber priorisiert worden sein, damit das agile Projektkernteam (A-PCT) mit der Arbeit beginnen kann. Ohne eine priorisierte Arbeitselementliste (WIL) besteht das Risiko, dass das Team Ressourcen in weniger wichtige Elemente investiert.
- Pflege der Arbeitselementliste (WIL) durch den Product Owner (PrOw) – Der Product Owner (PrOw) ist für die Pflege der aktualisierten und priorisierten Arbeitselementliste (WIL) zuständig und stellt sicher, dass das Team jederzeit an den relevantesten Arbeitselementen arbeitet.

- Bezugnahme auf Release Planning, Iterationsreview und Iterationsretrospektive – Als Teil des ständigen Lernprozesses und aufgrund des sich verändernden Kontextes der Lösung ist es sehr wichtig, auf Events wie Release Planning, Iterationsreview und Iterationsretrospektive Bezug zu nehmen. Diese liefern kontinuierlich Input und erzeugen Arbeitselemente, die in die Arbeitselementeliste (WIL) aufgenommen werden müssen.
- Projektprotokolle als Quelle von Arbeitselementen – Bei der Verwaltung der Projektprotokolle (Änderungsanforderungen, Risiken, Probleme und Entscheidungen) können sich verschiedene Arbeitselemente ergeben, die in die Arbeitselementeliste (WIL) aufgenommen werden müssen. Durch eine Änderungsanforderung können mehrere neue zu entwickelnde Storys entstehen um die Änderungsanforderung umzusetzen. Alternativ kann der Plan zur Minderung eines architektonischen Risikos mit der Umsetzung einiger Enabler Storys verbunden sein.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Arbeitselementeliste (WIL)	S	A	C	R	S	S

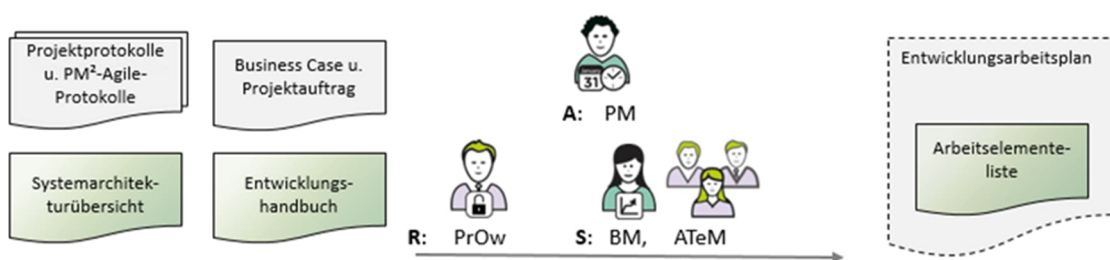


Abbildung 41: Arbeitselementeliste (WIL) – Input und primäre Rollen

Phasen des PM²-Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Arbeitselementeliste (WIL)	–	Erstellt	Aktualisiert	–

7.2.2.2 Releaseplan

Der Releaseplan dokumentiert die übergeordneten UX- und Entwicklungsmeilensteine. Er gibt auf Grundlage der Informationen, die dem Projektkernteam (PCT) zu einem bestimmten Zeitpunkt vorliegen, Auskunft darüber, was wann freigegeben wird. Der Releaseplan resultiert aus der anfänglichen Gesamtprojektplanung während der Initiierungs- und Planungsphase und sollte auf Grundlage des Fortschritts und Feedbacks des agilen Projektkernteam (A-PCT) angepasst werden.

Der Hauptzweck des Releaseplans besteht darin,

- die Freigabe von einsatzfähigen Versionen der Lösung zu planen,
- einen übersichtsartigen Plan mit den zu erreichenden Hauptzielen, einschließlich der zu liefernden wichtigsten Features, zu bieten,
- die Risiken, Probleme, Entscheidungen, Änderungen, Beschränkungen und Annahmen im Zusammenhang mit dem Entwicklungsaspekt wie etwa Abhängigkeiten von anderen Projekten und Informationssystemen sichtbar zu machen,

- den Zeitplan für die Einführungsaktivitäten (z. B. Bereitstellungsstrategie und-aktivitäten) zu unterstützen,
- den Zeitplan für geschäftliche Implementierungsaktivitäten (z. B. Schulungsaktivitäten) zu unterstützen.

Mit Releases wird ein Timebox-Rhythmus für die Konstruktionsarbeit eingeführt, der (ein gewisses Maß an) Vorhersehbarkeit sowohl für das agile Projektkernteam (A-PCT) als auch für die übrigen Stakeholderinnen und Stakeholder bietet.

Der Releaseplan erleichtert die Steuerung etwaiger Abhängigkeiten des in Entwicklung befindlichen Informationssystems mit anderen Projekten, Informationssystemen oder geschäftsbezogenen Aktivitäten.

Für jedes geplante Release müssen Einführungsaktivitäten festgelegt werden und es muss sichergestellt werden, dass die für das Release vorgesehene Umgebung vorbereitet ist und die Endnutzerinnen bzw. Endnutzer für die Nutzung der gelieferten Lösung bereit sind. Diese Aktivitäten sollten dokumentiert und auf den Einführungsplan abgestimmt sein.

Releases, die der gesamten antrag stellenden Organisation zur Verfügung gestellt werden sollen, müssen auch mit den erforderlichen Geschäftsimplementierungsaktivitäten verknüpft werden, um eine erfolgreiche Einführung des Informationssystems zu gewährleisten. Diese Aktivitäten sollten dokumentiert und auf den Geschäftsimplementierungsplan abgestimmt sein.

Wesentliche Beteiligte	Beschreibung
Projektleitung (PM)	Ist für die Abstimmung zwischen der in dem Releaseplan vorgesehenen Lieferstrategie und der übergeordneten Vision des Projekts sowie allen in dem Projektauftrag beschriebenen relevanten Meilensteinen verantwortlich.
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist für die Entwicklung des Releaseplans zuständig und wird dabei von dem Product Owner (PrOw), der Orientierung auf Basis der priorisierten Arbeitselementliste (WIL) gibt, und den übrigen agilen Teammitgliedern (ATeM), der zusätzliche Input zur Durchführbarkeit des Plans geben, unterstützt.
Agiles Teammitglied (ATeM)	Unterstützt die Teamkoordinatorin oder den Teamkoordinator (TeCo) durch die Bereitstellung von Input zur Durchführbarkeit des Plans einschließlich Abhängigkeiten zwischen den verschiedenen Arbeitselementen.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Unterstützt die Teamkoordinatorin oder den Teamkoordinator (TeCo) durch die Bereitstellung von Input zur globalen Architektur der Lösung und etwaigen Bausteinen, die Auswirkungen auf den Releaseplan haben können.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Wird von dem Product Owner (PrOw) über den aktualisierten Releaseplan informiert.
Product Owner (PrOw)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo) durch Bereitstellung der Lieferstrategie des Unternehmens in Form von Meilensteinterminen, Releasezielen und geplanten Features sowie entsprechenden Arbeitselemente.

Handlungsempfehlungen

- **Frühestmögliche Erstellung eines Releaseplans** – Das erste Release Planning-Event zur Erstellung des anfänglichen Releaseplans sollte so früh wie möglich durchgeführt werden. Vorzugsweise sollte dies zu Beginn der Planungsphase erfolgen, da der Releaseplan die wichtigste Grundlage für die Lieferung der Lösung darstellt.
- **Berücksichtigung der Releasestrategie im Releaseplan** – Der Releaseplan ist mehr als nur ein Terminkalender für größere Releases mit den wichtigsten Funktionen. Er muss die übergeordnete (Marketing-, operative, geschäftliche, rechtliche usw.) Strategie für die Freigabe der Lösung an die Nutzergemeinschaft widerspiegeln. Dabei spielt das Release Planning-Event eine entscheidende Rolle.
- **Bereitstellung unterschiedlicher Detailgrade** – Da sich der Releaseplan an verschiedene Arten von Empfängerinnen und Empfängern richtet, müssen unterschiedliche Detailgrade bereitgestellt werden. Beispielsweise ist das Entscheidungsgremium (AGB) an den verschiedenen geplanten Releases und ihren wichtigsten Zielsetzungen interessiert, um zu validieren, ob der Releaseplan auf die übergeordnete Strategie abgestimmt ist. Der Product Owner (PrOw) muss dagegen wissen, welche Arbeitselemente in den einzelnen Releases geliefert werden, da ihr/ihm dies dabei hilft, zu verifizieren, welcher Teil des jeweiligen Features freigegeben wird und wann.
- **Überprüfung des Plans nach jeder Iteration** – Am Ende der einzelnen Iterationen ergeben sich durch einige neue Inputs im Rahmen des Lernprozesses (aus Events wie den Iterationsreview und die Iterationsretrospektive), neue Arbeitselemente für die Arbeitselementeliste (WIL). Beispielsweise müssen die aktualisierten Daten zur Velocity des Teams im Releaseplan berücksichtigt werden, um seine Durchführbarkeit kontinuierlich zu bewerten.
- **Release Planning-Event** – Die Outputs der verschiedenen Schritte des Release Planning-Events fließen in einen aktualisierten Releaseplan ein. Hinzugefügte, aktualisierte oder von der Arbeitselementeliste (WIL) entfernte Arbeitselemente mit den entsprechenden relativen Schätzungen der agilen Teammitglieder (ATeM) müssen in dem Releaseplan berücksichtigt werden, um das aktuelle Szenario wiederzugeben.
- **Durchführbarkeit des Releaseplans** – Während des Projekts kommen ständig neue Fakten, Daten und Entscheidungen hinzu, die sich auf den Umfang der Arbeitselementeliste (WIL) und die Velocity des Teams auswirken. Diese beiden Elemente bestimmen die Durchführbarkeit des Releaseplans und ermöglichen es den geschäftlichen Stakeholderinnen und Stakeholdern, erforderlichenfalls Anpassungen vorzunehmen. Auch die Verwendung verschiedener Datensätze, wie etwa eine optimistische und eine pessimistische Velocity, kann dabei helfen, diverse mögliche Szenarien zu erstellen.
- **Aufstellung eines Bereitstellungsplans** – Die Strategie für die Bereitstellung der Software (und ihrer Updates) in der Produktionsumgebung sollte so früh wie möglich geplant werden, da sie Auswirkungen auf die Arbeitselementeliste (WIL) und den Releaseplan haben kann. Dies ist auch der Fall, wenn spezifische geschäftliche und unterstützende Abteilungen für die Verwaltung des übergeordneten Bereitstellungssystems des Unternehmens zuständig sind. Das Vorhandensein einer Continuous Delivery Pipeline als Teil einer DevSecOps-Initiative kann diesen Prozess optimieren und seine Komplexität drastisch reduzieren.

- Klärung und Sichtbarmachung von Annahmen – Die Velocity des Teams und der vordefinierte Umfang der einzelnen Iterationen sind zwei Beispiele für Elemente, die Auswirkungen auf das Verständnis eines Releaseplans haben. Bei der Definition des Releaseplans müssen diese Elemente klar als Annahmen des aktuellen Releaseplans sichtbar sein und identifiziert werden.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Releaseplan	I	A	R	S	S	S

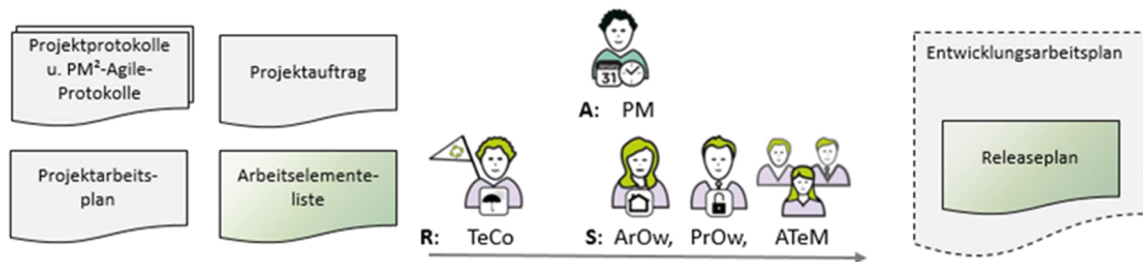


Abbildung 42: Releaseplan– Input und primäre Rollen

Phasen des PM²-Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Releaseplan	–	Erstellt	Aktualisiert	–

7.2.2.3 Iterationsplan

In seiner Minimalform ist der Iterationsplan die Gruppe der Arbeitselemente – eine Teilgruppe der Arbeitselemente-liste (WIL) –, die für die kommende Iteration geplant sind. Er enthält das Start- und Enddatum der Iteration, das Iterationsziel, das von dem Product Owner (PrOw) definiert und vom agilen Projektteam (A-PCT) zugesagt wurde, und die Liste aller Arbeitselemente, die zur Erreichung des Iterationsziels geliefert werden müssen. Darüber hinaus müssen in dem Iterationsplan möglicherweise Synchronisierungspunkte mit anderen Teams erfasst werden. Dies gilt insbesondere wenn diese an unterschiedlichen Projekten arbeiten. Dieses Artefakt dient außerdem dazu, die Probleme, Risiken usw. aufzuzeigen, die während der Iteration beseitigt werden müssen.

Der Iterationsplan hilft dem agilen Projektteam (A-PCT) bei der Überwachung des Fortschritts der Iteration (beispielsweise mithilfe eines Kanban-Boards und von Burndown-Charts der Iteration). Zudem hält er auf Grundlage der Outputs des Iterationsreview die Ergebnisse der Iteration für eine Bewertung fest, die für eine künftige Verbesserung nützlich sein kann.

Der Hauptzweck des Iterationsplans besteht darin, dem agilen Projektkernteam (A-PCT) Folgendes zur Verfügung zu stellen:

- eine zentrale Informationsquelle in Bezug auf das Iterationsziel,
- eine Visualisierung des Umfangs einer Iteration,
- die Ergebnisse der Evaluierung,
- einen Indikator für den aktuellen Fortschritt des Teams und eine Prognose für den Abschluss der Iterationsarbeiten.

Einer Iteration zugewiesene Arbeitselemente haben nicht notwendigerweise dieselbe Priorität. Sobald die Arbeitselemente der Iteration zugewiesen wurden, stellt das Team sicher, dass es alle Arbeiten abschließen kann, unabhängig von den ursprünglichen Prioritäten der Arbeitselemente (mit Ausnahme von Elementen, die blockierende Abhängigkeiten mit Elementen anderer Teams bilden). Die Entscheidung, was in einer Iteration zuerst entwickelt werden soll, variiert je nach Projekt und Iteration.

Arbeitselemente können Teil eines Features (Funktionseinheit) sein, das einen erkennbaren und wertvollen Teil der Lösung darstellen sollte. Die Arbeitselemente werden anschließend in die entsprechenden Aufgaben zerlegt, die für ihren Abschluss erforderlich sind.

Wesentliche Beteiligte	Beschreibung
Projektleitung (PM)	Ist für die Abstimmung des Ziels eines Iterationsplans auf die Strategie des Releaseplans und den Projektauftrag verantwortlich.
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist für die Entwicklung des Iterationsplans zuständig und wird dabei von dem Product Owner (PrOw), von dem das Iterationsziel bereitstellt wird, und den übrigen agilen Teammitgliedern (ATeM) unterstützt, die Informationen darüber liefern, welche Elemente wie implementiert werden sollen.
Agiles Teammitglied (ATeM)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo) durch die Bereitstellung von Input zur Durchführbarkeit des Plans, u. a. zu den Elementen, die während der Iteration implementiert werden können.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Unterstützt die Teamkoordinatorin bzw. den Teamkoordinator (TeCo) durch die Bereitstellung von relevanten Informationen über potenzielle architektonische Risiken/Probleme, die Auswirkungen auf den Iterationsplan haben können.
Product Owner (PrOw)	Wird von der Teamkoordinatorin bzw. dem Teamkoordinator (TeCo) konsultiert, damit er das Iterationsziel und eine priorisierte Arbeitselemente-liste (WIL) bereitstellt – den wichtigsten Input für den Iterationsplan.
Anforderungsmanagerin bzw. Anforderungsmanager (BM)	Wird von dem Product Owner (PrOw) und der Projektleitung (PM) über die Iterationsziele und darüber, was am Ende der Iteration erwartet werden kann, informiert.

Handlungsempfehlungen

- Eine Iteration, ein (Iterations-)Plan – Jede Iteration beginnt mit dem Iterationsplanungs-Event, bei dem eine Strategie zur Erreichung eines Iterationsziels festgelegt wird. Sie endet mit dem Iterationsreview-Event, bei dem das Ergebnis der geleisteten Arbeit im Hinblick auf das festgelegte Iterationsziel bewertet wird. Diese Informationen werden im entsprechenden eindeutigen Iterationsplan festgehalten.
- Beginn mit dem Iterationsplanungs-Event – Die vom agilen Projektkernteam (A-PCT) während des Iterationsplanungs-Events festgelegte Strategie stellt kritische Informationen für den Iterationsplan bereit, wie etwa die Iterationsdauer, das Iterationsziel, die zu implementierenden Arbeitselemente, Abhängigkeiten, damit verbundene Risiken usw. Die anfängliche Version des Burndown-Charts der Iteration wird ebenfalls auf Grundlage der Inputs aus diesem kritischen Event erstellt.
- Lebendiges Artefakt – Der Iterationsplan ist kein statisches Artefakt, da einige seiner Elemente häufig aktualisiert werden. Beispielsweise muss das Burndown-Chart kontinuierlich auf Basis der geplanten und erreichten Arbeitsleistungen an die Situation der Iteration angepasst werden. Der Umfang der Iteration ist ein weiteres Beispiel für ein Element, das überarbeitet werden und verschiedene Änderungen an dem Iterationsplan auslösen kann.
- Abschluss mit dem Iterationsreview – Im Rahmen des Iterationsreview-Events werden abschließende Schlussfolgerungen zu der vom agilen Projektkernteam (A-PCT) während der Iteration geleisteten Arbeit vorgestellt. Wurde das Iterationsziel erreicht? Gab es Arbeitselemente, die nicht erfolgreich geliefert wurden? Wenn ja – was war der Grund? Die bei diesem Event gesammelten wertvollen Informationen werden als Teil des Iterationsplans zur aktuellen und zukünftigen Verwendung erfasst.
- Einsatz von Softwaretools zur Erstellung und Pflege des Iterationsplans – Bevor die verschiedenen in den vorangegangenen Punkten beschriebenen Arten von Informationen in einem konventionellen manuell erstellten Dokument zusammengefasst werden, sollten alternative Softwaretools untersucht werden, die eine deutliche Zeitersparnis ermöglichen können.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Iterationsplan	I	A	R	C	S	S

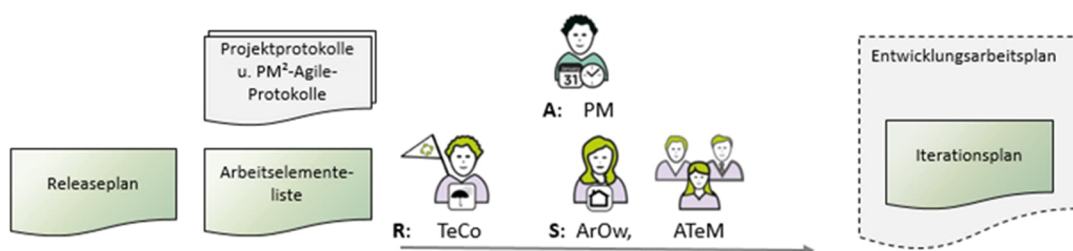


Abbildung 43: Iterationsplan – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Iterationsplan	Erstellt Aktualisiert	Erstellt Aktualisiert	Erstellt Aktualisiert	–

7.2.3 Testplan

In einem Testplan werden die Ziele/der Umfang/die Zielsetzungen von Tests, die zu testenden Elemente, der zu verfolgende Ansatz, die benötigten Ressourcen und die zu produzierenden Liefergegenstände festgelegt. Er spezifiziert das Universum der durchzuführenden Tests (z. B. Komponententests, Regressionstests, Funktionstests, Integrationstests, Nutzerakzeptanztests) sowie die Phasen, in denen die Tests ausgeführt werden (z. B. Entwicklung, Integration, Test, Abnahme). Auch die Testtechniken, Kennzahlen und Kriterien, die zur Bewertung der Testergebnisse und des Testabschlusses herangezogen werden, werden darin detailliert beschrieben.

Der Hauptzweck eines Testplans besteht darin, die Testarbeiten für einen bestimmten Zeitplan zu skizzieren und zu kommunizieren.

Jeder Testplan bildet den Rahmen, in dem die testdurchführenden agilen Teammitglieder (ATeM) für den jeweiligen Zeitplan arbeiten. Indem er die Testarbeiten lenkt, leitet und einschränkt, ermöglicht der Plan eine Konzentration auf die erforderlichen Liefergegenstände und kommuniziert den Zweck der Arbeiten an die Stakeholderinnen und Stakeholder.

Daher sollte ein Testplan keine Einzelheiten enthalten, die von den an den Testaktivitäten beteiligten Personen nicht verstanden oder als irrelevant angesehen würden. Da sich die verschiedenen Testebenen auf unterschiedliche Funktions- und Risikobereiche konzentrieren, können andere Organisationseinheiten verschiedene Zuständigkeiten in Bezug auf die Testarbeiten insgesamt haben. In diesem Fall sind detaillierte Testpläne für verschiedene Organisationseinheiten notwendig, um Tests zu planen, vorzubereiten, durchzuführen und über diese zu berichten, ohne das Risiko zu erhöhen.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Unterstützt die Erstellung des Testplans. Stellt in enger Zusammenarbeit mit der Projektleitung (PM) die Abstimmung auf das Projekthandbuch auf Projektebene sicher.
Agiles Teammitglied (ATeM)	Ist für die Erstellung des Testplans zuständig und leistet technische Beratung zu den Testaktivitäten, dem Umfang und der Art der Tests, den Testtechniken usw.
Systemarchitektin bzw. Systemarchitekt (ArOw)	Leistet Unterstützung, indem zu technischen Fragen beraten wird und technische Beschränkungen aufzeigt, die Auswirkungen auf die Testaktivitäten haben können.
Product Owner (PrOw)	Leistet Unterstützung, indem Erkenntnisse und Informationen über die Realität der Antrag stellenden Organisation, die Auswirkungen auf die Testaktivitäten haben können, bereitgestellt werden.
Projektleitung (PM)	Ist dafür verantwortlich, wie das agile Projektkernteam (A-PCT) mit dem Testaspekt des Projekts umzugehen beabsichtigt. Außerdem wird die Abstimmung der Testarbeiten auf die übergeordnete Steuerung der Abnahme der Liefergegenstände des Projekts sichergestellt.

Handlungsempfehlungen

- Festlegung einer globalen Teststrategie – In einer frühen Phase des Projekts sollte ein Entwurf eines Rahmentestplans erstellt werden. Die Testarbeiten sollten nicht als zusätzlicher Arbeitsaufwand verstanden werden, sondern vielmehr als Routinearbeiten in die Entwicklungszyklen (Iterationen, Release) einbezogen werden.
- Teamleistung – Allein die Tatsache, dass es sich um einen Testplan handelt, bedeutet nicht, dass er in der Zuständigkeit der Testerin bzw. Tester liegt. Der Testauftrag sollte mit dem gesamten agilen Projektteam (A-PCT) erörtert werden. Nach einer anfänglichen Einigung über den Testauftrag, die Ziele und spezifische Zielsetzungen ist der Testansatz zu definieren, indem diese Informationen im Testplan festgehalten werden.
- Abstimmung auf das Entwicklungshandbuch – Bei der Festlegung der globalen Teststrategie ist sicherzustellen, dass sie mit der im Entwicklungshandbuch beschriebenen Entwicklungsstrategie kompatibel ist.
- Fokussierung auf die Kommunikationsaspekte der Testarbeiten – An Tests können viele verschiedene Stakeholderinnen und Stakeholder zu unterschiedlichen Zeitpunkten beteiligt sein. Es ist wichtig, dass alle Beteiligten den Testansatz verstehen, d. h. die erwartete Einbeziehung der Stakeholderinnen und Stakeholder während der Testarbeiten.
- Finden des richtigen Gleichgewichts – Der Testplan und alle dazugehörigen Elemente sollten mit allen Stakeholderinnen und Stakeholdern durchgegangen werden, um eine Einigung zu erzielen. Dabei sollte der Schwerpunkt daraufgelegt werden, basierend auf den identifizierten Risiken und Prioritäten ein Gleichgewicht zwischen den Testressourcen und dem Umfang der aktuellen Testarbeiten zu finden.
- Dokumentation der Testrisiken und Maßnahmen zur Risikominderung im Testplan – Wenn die projektspezifische Risikoplanung Teil eines umfassenderen Risikomanagements ist, müssen die Testrisiken mit den Risikoanalyse-, Konzept- und Risikominderungsprozessen der Projektleitung (PM) koordiniert werden. Es ist sicherzustellen, dass testbezogene Risikoelemente genau zu den Anforderungen und Test-Assets zurückverfolgt werden können.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Testplan	I	A	S	S	S	R

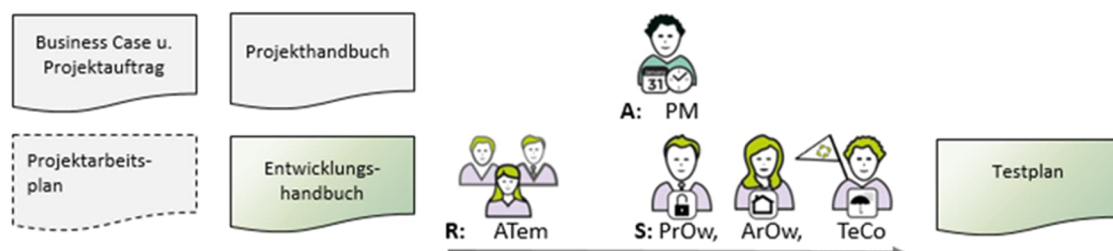


Abbildung 44: Testplan – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Testplan	–	Erstellt	Aktualisiert	–

7.2.4 Bereitstellungsplan

Im Bereitstellungsplan werden die Strategie, die Rollen und Zuständigkeiten sowie die Aufgaben festgelegt, die berücksichtigt werden müssen, damit das Informationssystem von dem für den IT-Betrieb zuständigen Team bereitgestellt, betrieben und verwaltet werden kann. Die Bereitstellung kann sowohl in der IT-Betriebsumgebung als auch für die Endnutzerinnen bzw. Endnutzer des Informationssystems zu zahlreichen Veränderungen und Belastungen führen. Dies gilt insbesondere dann, wenn sie nicht im Rahmen einer gut definierten und strukturierten Bereitstellungsstrategie erfolgt.

Da sich der Bereitstellungsplan auf die technischen Aspekte der Bereitstellung konzentriert, steht er in engem Zusammenhang mit den IT-Betriebsaktivitäten wie Cut-Over, Rollback, Back-Out-Verfahren für Notfälle und Risikomanagement.

Der Einführungsplan und der Geschäftsimplementierungsplan ergänzen den Bereitstellungsplan, indem sie die Strategie zur Sicherstellung, dass die Kundenorganisation für die Nutzung des Systems bereit ist (Schulungen, Support usw.), abdecken.

Der Hauptzweck des Bereitstellungsplans besteht darin,

- die Strategie, die bei den Bereitstellungsarbeiten über den Projektlebenszyklus angewendet werden soll, zu beschreiben und zu kommunizieren;
- Konsens und Bewusstsein im Hinblick auf die mit einer erfolgreichen technischen Bereitstellung der Lösung (IT-Betrieb) verbundenen Arbeiten und Zuständigkeiten zu schaffen.

Im Bereitstellungsplan sollte der iterative und inkrementelle Ansatz von PM²-Agile berücksichtigt werden. Ferner sollten die Bereitstellungsarbeiten zum Gesamtziel der regelmäßigen Freigabe funktionierender Lösungen für die Nutzer des Informationssystems beitragen. Der Bereitstellungsplan muss hinreichend ausführlich sein, um dem für den IT-Betrieb zuständigen Team die Bereitstellung, den Betrieb und die Steuerung des Informationssystems zu ermöglichen.

Der Bereitstellungsplan steht in engem Zusammenhang mit dem Geschäftsmodell, da beide Artefakte wichtige Elemente darstellen, um sicherzustellen, dass das Informationssystem in Produktionsumgebungen bereitgestellt und verwendet werden kann. Der Bereitstellungsplan sollte folgende Abschnitte enthalten:

- Bereitstellung in der Produktionsumgebung – Enthält Einzelheiten zum gesamten Bereitstellungsprozess, einschließlich der voraussichtlich verfügbar werdenden Features, der zu erstellenden Arbeitspakete, der zu verwendenden Bibliotheken, der Umgebungsanforderungen, der erstellten Konfigurationsdaten usw.
- Validierung der Lösung – Beschreibt die eingesetzten Mechanismen, um zu gewährleisten, dass die Bereitstellung ordnungsgemäß erfolgt ist. Anhand von spezifischen Belastungs- und Leistungstests sowie Smoke-Tests mit einigen Schlüssel-Features kann überprüft werden, ob eine Korrektur erforderlich ist, die möglicherweise zu einem vollständigen Rollback der Bereitstellung führt.

- Überwachung und Meldung von Problemen sowie entsprechende Reaktion – Beschreibt die vorgesehenen Mechanismen zur Unterstützung der Erkennung, Meldung und Lösung verschiedener Probleme. Ziel ist es, potenzielle Probleme zu identifizieren, bevor sie zu echten Problemen werden, die die ordnungsgemäße Bereitstellung und den ordnungsgemäßen Betrieb des Systems gefährden.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist in enger Zusammenarbeit mit der Projektleitung (PM) für die Erstellung des auf den Entwicklungsarbeitsplan (Releasezeitplan) und den Einführungsplan auf Projektebene abgestimmten Bereitstellungsplans zuständig.
Agiles Teammitglied (ATeM)	Unterstützt durch technische Beratung zu den Bereitstellungsarbeiten (Einzelheiten zu Datenmigration, Notfallmaßnahmen oder Continuous Delivery Pipeline).
Systemarchitektin bzw. Systemarchitekt (ArOw)	Leistet Unterstützung, indem zu technischen Fragen beraten wird und technische Beschränkungen aufgezeigt werden, die Auswirkungen auf den zu definierenden Bereitstellungsansatz haben können.
Product Owner (PrOw)	Leistet Unterstützung, indem Erkenntnisse und Informationen über die Realität der Antrag stellenden Organisation bereitgestellt werden, die Auswirkungen auf den zu definierenden Bereitstellungsansatz haben können.
Projektleitung (PM)	Ist für den Bereitstellungsansatz des agilen Projektkernteams (A-PCT) und die Abstimmung der Bereitstellungsarbeiten auf die Einführungsarbeiten insgesamt verantwortlich.
Vertreterinnen und Vertreter der maßgeblichen Geschäftseinheit	Leistet einen Beitrag durch die Bereitstellung von Informationen über die betrieblichen Prozesse und die Infrastruktur sowie durch die Prüfung der Durchführbarkeit, der Wirksamkeit und der Effizienz des Bereitstellungsplans und der damit verbundenen Risiken.

Handlungsempfehlungen

- Kompatibilität mit dem Entwicklungshandbuch – Wenn das Entwicklungshandbuch die inkrementelle und iterative Entwicklungsmethode von PM²-Agile vorsieht, muss der Bereitstellungsplan eine klare Strategie vorsehen, um diesen Ansatz zu ermöglichen. Die Einrichtung einer Continuous-Delivery-Pipeline ist eine strategische Entscheidung, die in dem Plan beschrieben und validiert werden muss.
- Entwicklung und Betrieb – Das DevSecOps-Mindset muss in eine Reihe von Praktiken übersetzt werden, die Softwareentwicklung und Betrieb umfassen. Es bringt diese Ökosysteme zur Erstellung des Bereitstellungsplans zusammen. Wenn die Organisation siloartig mit einer unabhängigen IT-Betriebsabteilung strukturiert ist, die von den Entwicklungsteams getrennt ist, sollte der IT-Betrieb in seiner Gesamtheit einbezogen werden, um betriebliche Beschränkungen zu berücksichtigen.
- Lebendiges Artefakt – Der Bereitstellungsplan muss in dem Maße überprüft werden, in dem sich das Projekt weiterentwickelt und Feedback aus früheren Bereitstellungsarbeiten gesammelt wird.
- Abstimmung auf den Releaseplan – Es muss sichergestellt werden, dass die Bereitstellungsarbeiten unter Berücksichtigung des Releaseplans geplant werden. Falls eine spezifische IT-Betriebsabteilung Bereitstellungen in der Produktionsumgebung ohne vorhandene DevSecOps-Praktiken leitet, wird diese Abstimmung entscheidend.

- Test im Vorfeld – Es muss sichergestellt werden, dass das Betriebsteam den Bereitstellungsplan verstehen und die Lösung in einer Vorproduktionsumgebung erfolgreich bereitstellen, betreiben und verwalten kann. Es sollten mehrere Probeläufe durchgeführt werden, um eine reibungslose Einführung in der Produktionsumgebung zu gewährleisten.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Bereitstellungsplan	I	A	R	S	S	S

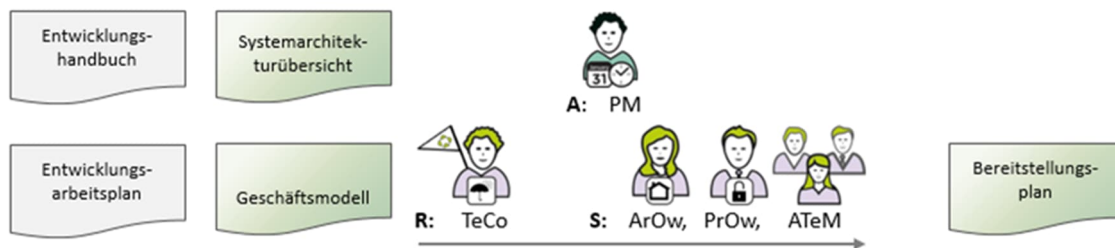


Abbildung 45: Bereitstellungsplan – Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Bereitstellungsplan	–	Erstellt	Aktualisiert	–

7.3 Koordinierungs- und Berichtsartefakte

7.3.1 Entwicklungsstatusbericht

Der Entwicklungsstatusbericht dokumentiert und vermittelt den Stand des agilen Projektkernteams (A-PCT) bei der Erreichung der Entwicklungsziele und ist in das Berichtswesen auf Projektebene integriert.

Ähnlich wie der Projektstatusbericht enthält der Entwicklungsstatusbericht Informationen zur Nachverfolgung von Kosten, Zeitplan, Umfang, Risiken, Problemen, Änderungen sowie Prognosen für die nächsten Schritte des Projekts. Je nach den für das Projekt festgelegten Berichtsanforderungen kann der Entwicklungsstatusbericht nach jeder Iteration und jedem Release erstellt werden.

Ein wesentliches Element des Berichts – sowohl auf Iterations- als auch Releaseebene – ist das Burndown-Chart. Das Release Burndown-Chart zeigt den geschätzten Funktionsumfang, der für das aktuelle Release noch fertigzustellen ist, und beantwortet die folgenden Fragen:

- Wann kann ein Release auf Basis der Leistung (Velocity) des Teams fertiggestellt werden?
- Wie weit ist die Arbeit bisher fortgeschritten?
- Reicht die Velocity des Teams aus, um das Release fristgerecht fertigzustellen?

Das Release Burndown-Chart¹⁴ bietet einen allgemeinen Überblick über den Fortschritt eines Release. Es wird in jeder Iteration mindestens einmal überprüft und gibt Aufschluss darüber, ob das Team die Funktionalität in der vorgesehenen Zeit liefert. Es kann auch Projekte hervorheben, deren Umfang außer Kontrolle geraten ist und die daher den erwarteten Fortschritt beeinträchtigen. Das auf einem Trend oder einer Prognose basierende Diagramm ermöglicht dem Team, Maßnahmen zur besseren Kontrolle des Umfangs zu ergreifen oder Ressourcen, Budget, Zeitrahmen, Qualitätsniveau oder Zusagen anzupassen.

Das Burndown-Chart der Iteration ist das primäre Tool, um sich über den Stand der aktuellen Iteration zu informieren. Es zeigt die aktuelle Teamleistung auf Basis des geschätzten noch verbleibenden Aufwands in Gegenüberstellung zur optimalen Teamleistung auf Basis des erwarteten verbleibenden Aufwands, der die Erfüllung des Iterationsziels gewährleistet.

Das Burndown-Chart der Iteration sollte häufig, am besten täglich, aktualisiert werden. Tägliche oder noch häufigere Aktualisierungen ermöglichen dem Team, auf Änderungen zu reagieren. Änderungen können beispielsweise darin bestehen, den Projektumfang durch Entfernen von Arbeitselementen aus der Iteration oder von der Arbeitselementliste zu reduzieren, das für ein Arbeitselement festgelegte Anspruchsniveau zu verringern oder bessere Herangehensweisen für die Bearbeitung von Arbeitselementen zu finden.

Im Gegensatz zum Burndown-Chart der Iteration, das den verbleibenden Aufwand in Points oder Stunden darstellt, verfolgt das Release Burndown-Chart das Ziel, festzustellen, ob das Team bei jeder Iteration eine funktionierende Software liefert. Das resultierende Release Burndown-Chart hilft, die vom Team tatsächlich gelieferte Funktionalität mit dem von ihm erwarteten Lieferumfang zu vergleichen.

Wesentliche Beteiligte	Beschreibung
Teamkoordinatorin bzw. Teamkoordinator (TeCo)	Ist für die Erstellung des Entwicklungsstatusberichts zuständig und hat in enger Absprache mit der Projektleitung (PM) sicherzustellen, dass die Informationen über den entwicklungsbezogenen Fortschritt mit dem Projektstatusbericht auf Projektebene übereinstimmen.
Agiles Teammitglied (ATeM)	Unterstützt die Erstellung des Entwicklungsstatusberichts durch die Bereitstellung von Informationen über die täglich bearbeiteten Aufgaben.
Projektleitung (PM)	Die Projektleitung (PM) als für den Entwicklungsstatusbericht verantwortliche Instanz muss Kenntnis von den Fortschritten des agilen Projektkernteam (A-PCT) haben. Sie stellt außerdem sicher, dass relevante Informationen aus dem Entwicklungsstatusbericht in den Projektstatusbericht aufgenommen werden.
Product Owner (PrOw)	Wird über den Fortschritt des Teams informiert.

¹⁴ Weitere Informationen zum Burndown-Chart können dem Tool/der Technik „Burndown-Charts“ entnommen werden.

Handlungsempfehlungen

- Entwicklungsarbeitsplan hinzuziehen – Die Iterations- und Releasepläne enthalten wesentliche, transparente Daten für einen präzisen und objektiven Entwicklungsstatusbericht. Einige verfügbare Elemente sind die Velocity des Teams, Burndown-Charts, die Arbeitselementeliste und eine aktualisierte Releasestrategie.
- Projektmanagement-Protokolle sind ebenfalls wichtig – Probleme, Entscheidungen und Risiken gehören zum Alltag eines Projekts und müssen engmaschig gesteuert werden. Durch die laufende Aktualität der Projektmanagement-Protokolle wird sichergestellt, dass alle Punkte behandelt und gelöst werden.
- Fortschrittsbesprechungen an den Iterations- und Releasezyklen ausrichten – Da jede Iteration objektive Informationen über den tatsächlichen Fortschritt des agilen Projektkernteams (A-PCT) liefert, ist die Planung einer Fortschrittsbesprechung am Ende jeder Iteration ein wirksames Mittel, um über den Stand des Entwicklungsaufwands zu informieren.
- Transparenz – Die Statusinformationen sind für die Stakeholderinnen und Stakeholder und das Projektteam stets einsehbar in einem Projektarbeitsbereich (Projektwand oder automatisiertes Tool), den die Stakeholderinnen und Stakeholder aufsuchen und in dem sie sich selbst über den Fortschritt des Teams informieren können.

RAM (RASCI)	BM	PM	TeCo	PrOw	ArOw	ATeM
Entwicklungsstatusbericht	I	A	R	I	I	S

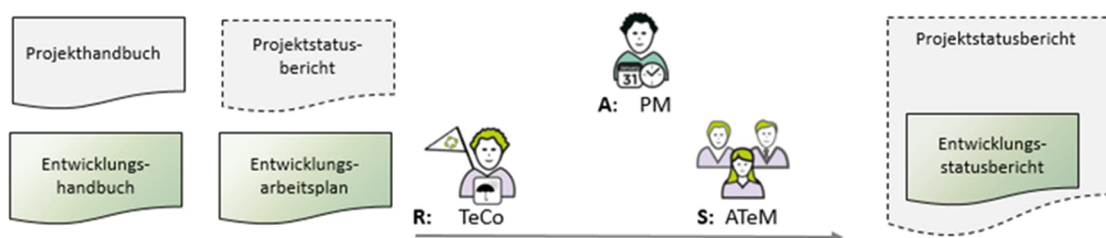


Abbildung 46: Entwicklungsstatusbericht: Input und primäre Rollen

Phasen des PM ² -Lebenszyklus:	Initiierung	Planung	Durchführung	Abschluss
Entwicklungsstatusbericht	–	Erstellt	Erstellt	–

7.3.2 Projektprotokolle

Um die allgemeine Überwachung der Projektleistung in der Durchführungsphase zu ermöglichen, sollte das agile Projektkernteam (A-PCT) auch bereichsspezifische Risiken, Probleme, Entscheidungen und Änderungen dokumentieren. Bei der Verwaltung dieser Informationen kann das Projektkernteam (PCT) Elemente identifizieren, die möglicherweise die Gesamtleistung beeinflussen und innerhalb der allgemeinen Projektorganisation zu behandeln sind.

Bei der Führung dieser Protokolle kann das Projektkernteam (PCT) die Projektleitung (PM) dabei unterstützen, den Stakeholderinnen und Stakeholdern über die Projektentwicklung zu berichten und diesbezügliche Prognosen abzugeben. Die Projektprotokolle werden von der Projektleitung auch als Koordinierungstools eingesetzt, um dem agilen Projektkernteam (A-PCT) Elemente zur weiteren Evaluierung und Steuerung zuzuweisen.

Als Ausgangspunkt können die Managementpläne und die Artefakte verwendet werden. So wird sichergestellt, dass Informationen über bereichsspezifische Risiken, Probleme, Entscheidungen und Änderungen richtig verwaltet werden.

Die nachfolgende Tabelle enthält die Managementpläne und die jeweiligen Protokolle für die einzelnen Elemente.

	Referenz
Risiken	Risikomanagementplan, Risikoprotokoll
Probleme	Problemmanagementplan, Problemprotokoll
Entscheidungen	Entscheidungsprotokoll
Änderungen	Änderungsmanagementplan, Änderungsprotokoll und Änderungsantrag
Testprotokoll	Testpläne und Liefergegenstandsabnahmeplan
Protokoll der Retrospektive	Projektabschlussbericht

7.3.3 Projektberichte

Der Entwicklungsaspekt des Projekts sollte zusammen mit bereichsspezifischen Informationen über den Entwicklungsstand und Fortschritt in die Projektberichte einfließen. Ziel ist es, die bereichsspezifischen Informationen und Metriken zur Überwachung des Entwicklungsfortschritts zu übersetzen und in die Statusberichte auf Projektebene und Projektberichte aufzunehmen.

Der Projektstatusbericht bringt im Abschnitt über den Entwicklungsstatusbericht das Feedback von der Entwicklungsseite des Projekts in die Verwaltungsebene ein.

Bei Projekten, die dem PM²-Agile-Ansatz folgen, ist der Projektfortschrittsbericht ein häufig verwendetes Reporting-Tool (z. B. alle zwei Monate oder vierteljährlich im Gegensatz zum jährlichen Turnus bei mehrjährigen Projekten), mit dem das Projektkernteam (PCT) und das Gesamtprojekt bedeutende Umfangsänderungen besser erfassen und steuern können.

Wie bei der PM²-Methodik bietet der Projektfortschrittsbericht einen groben Überblick über das Gesamtprojekt und seinen aktuellen Status.

Der Bericht enthält eine Projektübersicht (Stakeholderinnen und Stakeholder, Meilensteine und Liefergegenstände, Projektplan, Budget und Kosten) und weitere Angaben zum Projekt (Umfangsänderungen, wesentliche Risiken/Probleme und getroffene Maßnahmen, Erfolge).

Mit diesem groben Überblick über das Gesamtprojekt, einschließlich bereichsspezifischer Besonderheiten, und den Aktualisierungen des ursprünglichen Projektauftrags kann allen Stakeholderinnen und Stakeholdern vermittelt werden, wie das Projekt vorankommt, was sich seit dem ersten Projektauftrag geändert hat und welche Gründe zu diesen Änderungen geführt haben. Je nach Komplexität des Projekts und den Auswirkungen von Umfangsänderungen während der Entwicklung der Lösung ist unter Umständen eine Ad-hoc-Berichterstattung über den Projektstatus (einschließlich eines aktualisierten Projektauftrags) notwendig.

8. Tools und Techniken

In diesem Abschnitt werden einführend und allgemein einige der meistverwendeten Tools und Techniken in PM²-Agile zur Verbesserung der Qualität und Steigerung der Agilität in der Produktentwicklung vorgestellt.

Eine ausführlichere Beschreibung dieser Tools und Techniken ist in den spezifischen Wissensbereichen und Anwendergemeinschaften (Communities of Practice) zu finden.

8.1 Bewertung der Teamarbeit

Ein effektives Team muss mehrere Hindernisse überwinden, wie Patrick Lencioni in seinem Bestseller „Die fünf Dysfunktionen eines Teams“ anschaulich darlegt. Er beschreibt diesen Prozess als eine Abfolge von aufeinander aufbauenden Schritten, mit denen eine Gruppe von Personen zu einem effektiven, reifen und leistungsfähigen Team wird.

Mit dieser Bewertung können Führungskräfte die Fallstricke, die Teams auf Abwege geraten lassen, untersuchen und überwinden. Sie gibt den Teammitgliedern ein Gefühl für ihre besonderen Stärken und identifiziert Verbesserungspotenzial in jedem der fünf Grundbausteine für den Aufbau eines kohärenten, produktiven Teams: Vertrauen, Konfliktfähigkeit, Commitment, Rechenschaftspflicht und Ergebnisse.

8.2 Selbst organisierte Teams

Selbst organisierte Teams sind in der Lage und dazu befugt, Entscheidungen zu treffen und sich schnell an geänderte Anforderungen anzupassen. Aus Sicht sozialer Systeme bedeutet dies, dass das Team neue Ansätze entwickeln und sich an neue Herausforderungen in seinem Umfeld anpassen kann. Selbstorganisation ist ein Prozess und kein einmaliger Vorgang.

Die Förderung von selbst organisierten Teams in einer Organisation ist eine strategische Entscheidung. Führungskräfte müssen die Rahmenbedingungen schaffen, in denen Teams erfolgreich arbeiten und sich kontinuierlich selbst organisieren können. Sie müssen sich darauf einlassen, die Entwicklung von Verhaltensweisen zu lenken, die sich aus der Interaktion unabhängiger Akteure ergeben, anstatt im Voraus festzulegen, was effektives Verhalten ist.

Es bedarf der gemeinsamen Verpflichtung und Anstrengung vieler verschiedener Ebenen der Organisation (oberste Führungsebene, Projektleitung und Entwicklungs-/Teamleitung), damit sich eine Gruppe zu einem selbst organisierten, leistungsfähigen Team entwickelt.

8.3 Design Blocks

Design Blocks sind Teil einer Iteration und bilden einen Ausgangspunkt zur kritischen Überprüfung der Anforderungen, die als Annahmen für das jeweilige Feature betrachtet werden. Design Blocks beruhen auf den Techniken des Design Thinking, die sich drauf konzentrieren, die in der nächsten Iteration zu liefernden Arbeitselemente zu konkretisieren.

Aus praktischer Sicht handelt es sich bei Design Blocks um eine Reihe von Workshops, in denen das agile Projektkernteam (A-PCT) und andere Stakeholderinnen und Stakeholder zusammenkommen, sich austauschen und die Annahmen für ein Feature oder die Arbeitselemente für die nächste Iteration formulieren.

Die Person, die den Workshop moderiert und koordiniert (UX-Spezialistin oder UX-Spezialist) lässt alle Teilnehmenden einen Low-Fidelity-Prototyp erstellen, der durch ihren gemeinsamen Entwurf des

künftigen Features entsteht. Die UX-Spezialistin oder der UX-Spezialist stellt sicher, dass Design, Interaktion, Verhalten, Anforderungen und technische Aspekte gründlich erörtert und von allen verstanden und vereinbart werden.

Wenn das agile Projektkernteam (A-PCT) von der konkreten Low-Fidelity-Version des Features überzeugt ist, erstellen die Designerinnen und Designer der Benutzungsoberfläche (UI) einen High-Fidelity-Prototyp unter Anwendung der UI-Standards aus vorhandenen Designbibliotheken. Dieser High-Fidelity-Prototyp bestimmt dann die Codeerstellung für das Feature im Rahmen der täglichen Arbeit des Teams.

8.4 Features und Storys

Features und Storys sind eine Möglichkeit auszudrücken, welche Anforderungen eine bestimmte Lösung erfüllen sollte. Im Allgemeinen wird eine neue Funktionalität (oder Aktualisierungen einer vorhandenen Funktionalität) zuerst grob beschrieben und anschließend in kleinere Elemente zerlegt, wenn mehr Details benötigt werden und größere Klarheit über diese Details besteht.

Features sind Dienste/Funktionen, die durch eine Lösung zur Erfüllung der Stakeholderbedürfnisse bereitgestellt werden, während Storys kleine Segmente sind, die vertikalen Scheiben (Slices) der gewünschten Funktionalität des Systems entsprechen. Die Umsetzung eines Features ist komplexer und dauert länger als die Umsetzung einer Story. Das liegt daran, dass ein Feature alle relevanten Szenarien berücksichtigt, sodass es sehr unwahrscheinlich ist, dass es innerhalb des Umfangs einer einzelnen Iteration umgesetzt werden kann.

Da das agile Projektkernteam (A-PCT) die iterative Entwicklung als Methode für eine wirksame inkrementelle Entwicklung einsetzt, muss jedes Feature in eine entsprechende Anzahl von Storys zerlegt werden, die in die Arbeitselementeliste (WIL) aufgenommen werden können.

Storys sind äußerst praktisch, weil sie von ihrer Konzeption her in kurzer Zeit implementiert werden können und einen Mehrwert für die Stakeholderinnen und Stakeholder generieren. Storys sind das ideale Tool für agile Teams, weil sie ihnen ermöglichen, die Lösung „Stück für Stück“ zu entwickeln und dabei Feedback in kurzen Iterationen zu sammeln, zu analysieren und zu integrieren. Auf diese Weise lässt sich gut validieren, ob das Team auf dem richtigen Weg ist.

Ein weiterer Vorteil von Storys liegt darin, dass sie aufgrund ihrer einfachen Denkweise und Struktur einem Team und anderen Stakeholderinnen und Stakeholdern ein effizientes Management des Projektumfangs ermöglichen. Mithilfe von Storys ist es einfach zu verstehen, was umgesetzt und geliefert wurde, und die noch verbleibende Arbeit zu planen.

8.5 Zerlegen von User Storys

User Storys können während des Lebenszyklus eines Projekts unterschiedlich groß sein. Eine umzusetzende User Story sollte die richtige Größe aufweisen, damit Schätzung, Priorisierung und Entwicklung innerhalb eines angemessenen Zeitfensters (einer Iteration) möglich sind. Das ist jedoch nur der eine Aspekt. Es gibt noch verschiedene andere gute Gründe, warum die Beherrschung dieses Tools so wichtig ist.

- Umsetzung nach dem Bausteinkonzept – Das Design wird besser, wenn man mit kleineren User Storys arbeitet. Ein besseres Design trägt zu einer höheren Wartbarkeit bei und erleichtert die Skalierbarkeit.

- Einfachere Reaktion auf Änderungen – Kleinere User Storys sind gleichbedeutend mit kleineren Änderungen, die weniger Auswirkungen haben und einfacher umzusetzen und damit auch einfacher zu integrieren sind.
- Risikoreduzierung – Da kleinere Storys einfacher zu beschreiben und zu demonstrieren sind, tragen sie zur Verringerung eines der größten Risiken bei einem Produkt bei: der Unsicherheit.
- Geringere Variabilität – Da kleinere Storys einfacher und durch geringere Variabilität gekennzeichnet sind, verzögern sie häufig nicht die Lieferung und durchlaufen den Implementierungsprozess (das System) schneller, sodass sie die Warteschleife verkürzen.

8.6 Definition of Done

In einer iterativen und inkrementellen Entwicklungsumgebung muss sich das Entwicklungsteam darauf einigen, welche Voraussetzungen erfüllt sein müssen, damit ein bestimmtes Arbeitselement als fertiggestellt gilt. Eine vereinbarte „Definition of Done“ (DoD) besteht aus einer Checkliste mit Kriterien, anhand derer die agilen Teammitglieder (ATeM) prüfen, ob ein umgesetztes Arbeitselement den Qualitätsstandards des Teams entspricht.

Die DoD sollte das Ergebnis der Gespräche und Vereinbarung des Entwicklungsteams darüber sein, was „Fertig“ im Kontext eines Projekts bedeutet. In einigen Organisationen gibt es möglicherweise bereits eine einheitliche „Definition of Done“, die von den verschiedenen Entwicklungsteams verwendet werden kann, sodass projektübergreifend ein gemeinsames Verständnis entsteht, was „Fertig“ für die Organisation bedeutet.

Ein agiles Projektkernteam (A-PCT) profitiert von einer DoD für seine Arbeitselemente, weil dadurch auch der Qualitätsaspekt der Outputs in den Entwicklungszyklus einfließt und die Wahrscheinlichkeit abnimmt, dass in einer späteren Phase des Informationssystem-Lebenszyklus Probleme (z. B. Fehler) auftreten.

8.7 Planning Poker

Planning Poker ist eine konsensorientierte Technik zur Schätzung des relativen Umfangs der Entwicklungsziele. Sie ermöglicht dem Team, effizient eine Einigung erzielen, ohne zu viel Zeit mit einem bestimmten Thema zu verbringen, und gibt den Teammitgliedern Raum, ihre Meinungen, Gedanken und Bedenken zu äußern.

Diese Technik wird idealerweise in einer Workshop-Umgebung eingesetzt, um Arbeitselemente mit höherer Priorität zu verfeinern und den damit verbundenen Aufwand zu schätzen.

8.8 Kanban-Methode

Kanban ist eine Technik aus der Lean-Fertigung. Kanban bedeutet auf Japanisch „Schild“ oder „Tafel“ und ist ein wesentliches Tool des visuellen Managements. Die Kanban-Methode wurde erstmals von Toyota in seinem Produktionssystem für Autos eingesetzt und ermöglichte den vorgelagerten Prozessen, ein Teil erst dann zu produzieren, wenn es von den nachgelagerten Prozessen angefordert wurde. Dadurch wurde der Bestand an unfertigen Erzeugnissen reduziert und die Prozesseffizienz verbessert.

Kanban visualisiert den Prozess in Form von Karten auf einer Tafel und beschreibt, was wann zu produzieren bzw. welches Feature zu welchem Zeitpunkt im Entwicklungsprozess hinzuzufügen ist. Bei den herkömmlichen Push-Methoden baut sich im Laufe des Produktionsprozesses Verschwendung im Bestand auf, wenn in der Produktionskette Engpässe auftreten. Bei der Pull-Methode gibt der letzte Schritt in der Produktionskette dem vorgelagerten Schritt Bescheid, wann er für die Produktion bereit

ist, und erhält dann das Ergebnis des vorherigen Arbeitsschritts. Dieses „Pull“-Prinzip setzt sich rückwärts Schritt für Schritt bis zum allerersten Arbeitsschritt fort.

Die Verwendung der Pull-Methode für die Entwicklung und der Kanban-Methode zur Kontrolle und Visualisierung der Produktionskette hat mehrere Vorteile. Deutlich weniger Fehler, schnellere Fertigstellung, geringeres Risiko, höhere Sichtbarkeit und Transparenz und weniger Bestände in Warteschleife sind nur einige davon.

Diese Methoden erfordern jedoch eine strenge Organisation und ein sorgfältiges agiles Projektteam (A-PCT).

8.9 Burndown-Charts

Burndown-Charts sind grafische Tools zur Nachverfolgung der unterschiedlichen Dimensionen von Fortschritt in einem agilen Projekt. Die Diagramme zeigen, wie ein geplantes und messbares Element (z. B. Stundenaufwand oder Story Points) auf null (keine zu bearbeitenden Aufgaben mehr) „heruntergebrannt“ wird.

- Burndown-Charts verfolgen den Fortschritt von Iterationen (Stundenaufwand oder Story Points über mehrere Tage hinweg) und den Fortschritt von Releases (Story Points über mehrere Iterationen hinweg) und zeigen Hindernisse auf, die die Teamleistung beeinträchtigen könnten.
- Burndown-Charts sind insofern dynamisch, als sich die geplante Arbeit infolge des bei der Entwicklung erhaltenen Feedbacks ändern kann.
- Burndown-Charts sollten gut sichtbar sein, damit die Mitglieder des Projektkernteam (PCT) und die allgemeinere Projektorganisation den aktuellen Fortschritt und die Prognose völlig transparent einsehen können.

8.10 Testgetriebene Entwicklung

Testgetriebene Entwicklung bezeichnet die Praktik, Entwicklertests und Implementierungscode gleichzeitig mit hoher Granularität zu erstellen.

Bei der testgetriebenen Entwicklung erstellen die Entwicklerinnen und Entwickler zuerst einen kleinen Testfall, um ein Stück Code zu validieren, und führen den Test anschließend aus, um zu bestätigen, dass er fehlschlägt (Plausibilitätsprüfung, sog. „Sanity Check“). Anschließend programmieren sie gerade so viel Implementierungscode, wie nötig ist, damit der Test fehlerfrei ausgeführt werden kann. Dieser Zyklus ist kurz und dauert selten länger als zehn Minuten.

Jeder Zyklus beginnt mit den Tests. Nach Abschluss eines Tests fahren die Entwicklerinnen und Entwickler mit dem nächsten Test fort, bis für die Implementierung des gerade entwickelten Arbeitselements keine Tests mehr erstellt werden müssen. Dadurch wird auch sichergestellt, dass Entwicklerinnen und Entwickler nur den notwendigen Code zur Implementierung des jeweiligen Arbeitselements programmieren, da sie sich darauf konzentrieren, den Code zu schreiben, mit dem die erstellten Tests validiert werden.

Ein weiterer wesentlicher Vorteil der testgetriebenen Entwicklung liegt darin, dass sie kleine Schritte bei der Softwareprogrammierung ermöglicht, was sicherer und weitaus produktiver ist, als den Code in großen Inkrementen zu programmieren.

8.11 Pairing

Pairing ist eine Technik, bei der zwei agile Teammitglieder (ATeM) gemeinsam an einer bestimmten Aktivität arbeiten, zum Beispiel an der Ausgestaltung einer Teilmenge der Lösung, der Codeerstellung oder dem Testen einer Funktion. Geht es bei der Zweierarbeit um die Programmierung eines Teils des Systems (z. B. einer bestimmten User Story) spricht man von „Pair Programming“. In diesem Fall konzentrieren sich beide Entwickler auf den erstellten Code.

Pair Programming beinhaltet keine Coaching- oder Schulungsbeziehung, da beide als Kolleginnen und Kollegen am selben Projekt arbeiten. Besteht ein Paar aus einem erfahrenen und einem weniger erfahrenen Teammitglied, findet natürlich ein gewisses „Training-on-the-Job“ statt, da die Entwicklerin oder der Entwickler mit mehr Erfahrung den anderen unterstützt, hochwertige Software zu erstellen.

Pairing (und insbesondere Pair Programming) ermöglicht eine sofortige Überprüfung der Arbeit in Echtzeit. Sie verbessert die Qualität der Liefergegenstände, da zwei Kolleginnen und Kollegen besprechen und untersuchen, wie die anstehende Aufgabe am besten umzusetzen ist.

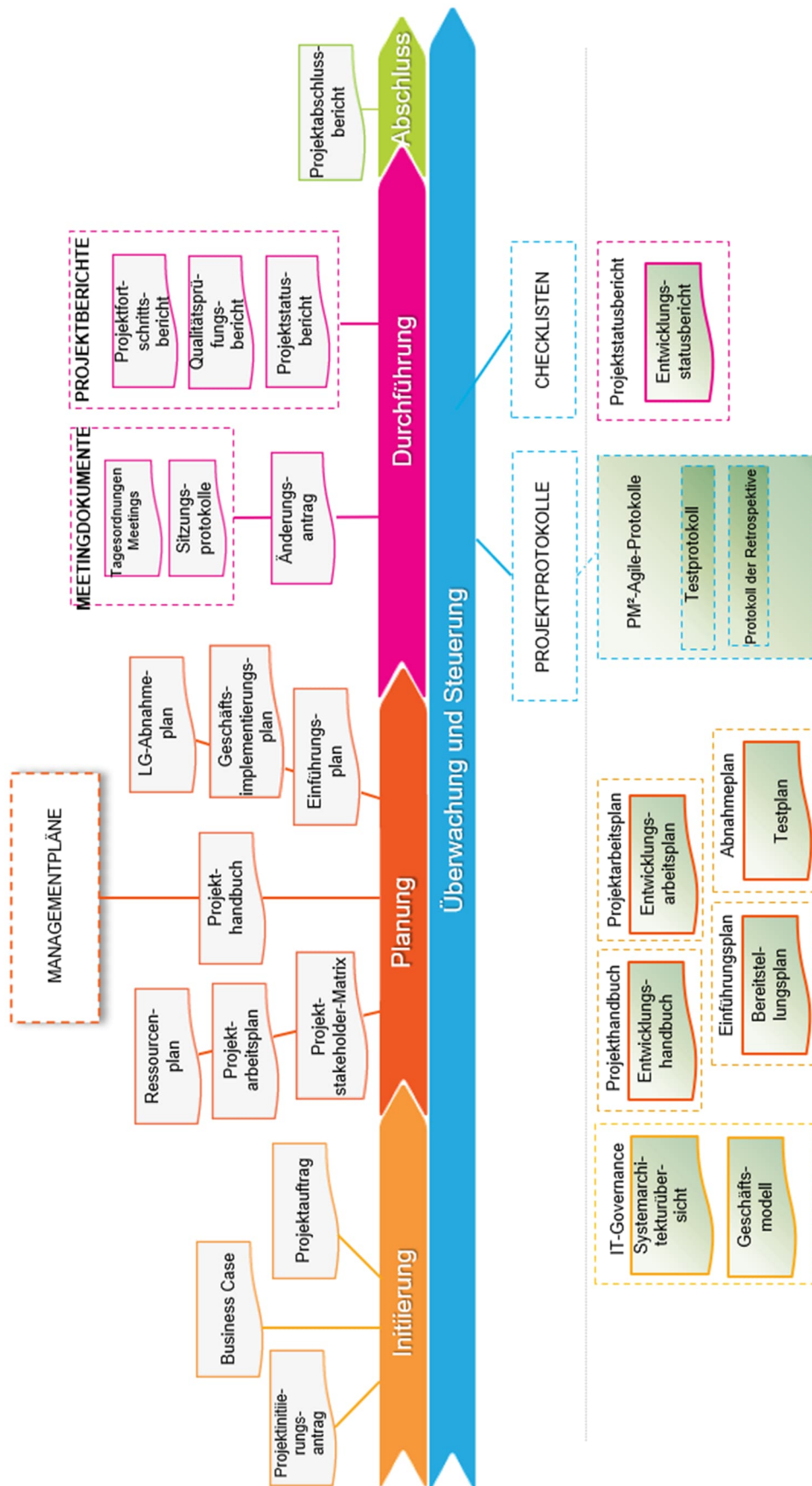
ANHANG

1. Tabellarische Übersichten und Diagramme der agilen Artefakte und Events

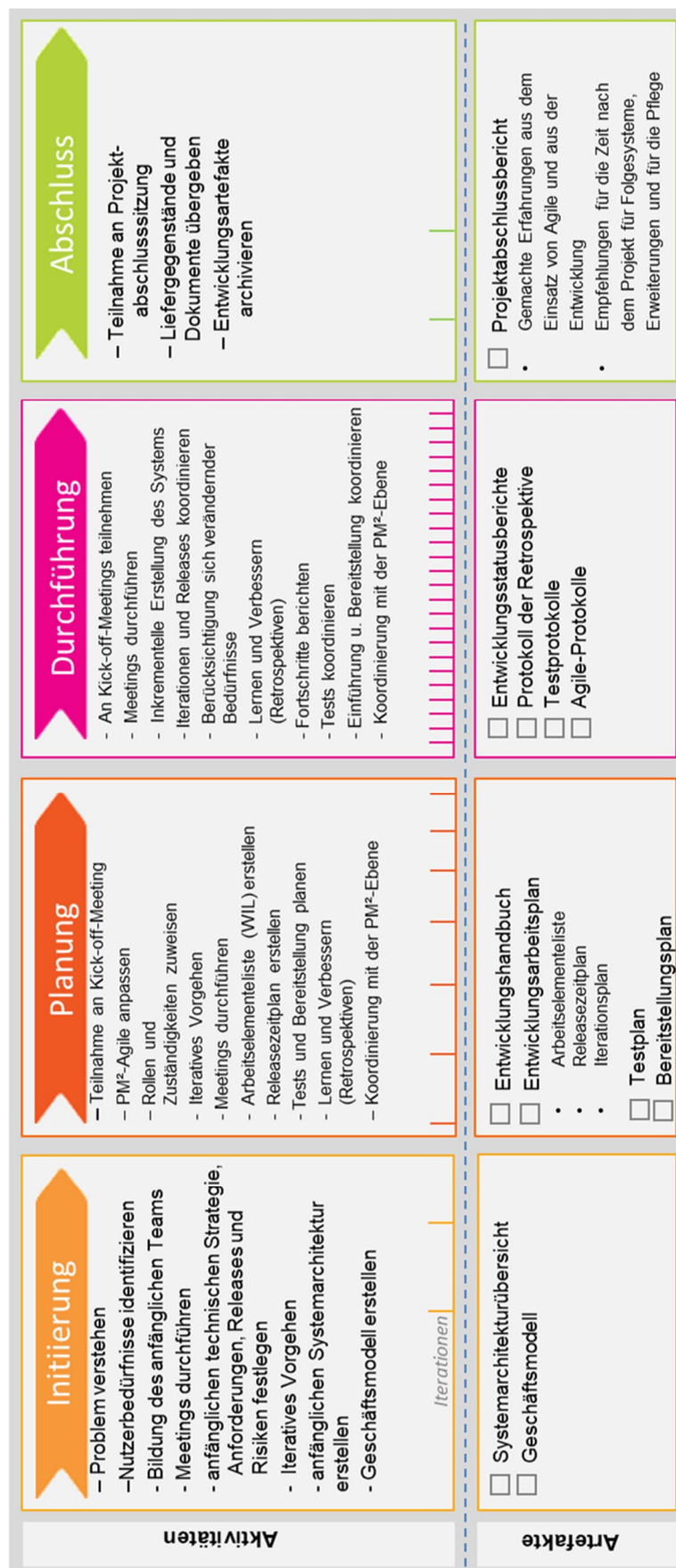
RAM (RASCI) – Responsible, Accountable, Supports, Consulted, Informed

Initiierung	BM	PM	TeCo	PrOw	ArOw	ATeM
Business Case	R	S	–	–	–	–
Projektauftrag	S	R	–	–	–	–
Systemarchitekturübersicht	I	A	S	I	R	S
Geschäftsmodell	I	A	S	I	R	S
Planung	BM	PM	TeCo	PrOw	ArOw	ATeM
Entwicklungshandbuch	I	A	R	S	S	S
Entwicklungsarbeitsplan	I	A	R	S	S	S
• Arbeitselementliste (WIL)	S	A	C	R	S	S
• Releaseplan	I	A	R	S	S	S
• Iterationsplan	I	A	R	C	S	S
Testplan	I	A	S	S	S	R
Bereitstellungsplan	I	A	R	S	S	S
Iterationsplanungsmeeting	I	I	S	S/A	S	R
Durchführung	BM	PM	TeCo	PrOw	ArOw	ATeM
Entwicklungsstatusbericht	I	A	R	I	I	S
Protokoll der Retrospektive	I	A	R	I	I	S
Testprotokolle	I	A	R	I	S	S
Daily Stand-up	–	I	S	I	R	R/A
Iterationsreview-Meeting	C	I	S	S/A	R	R
Iterationsretrospektive-Meeting	I	I	S	I	R	R/A
Überwachung und Steuerung	BM	PM	TeCo	PrOw	ArOw	ATeM
Release Planning						
• Pflege der Arbeitselementliste	S	I	S	R/A	I	I/S
• Ausgestaltung der Arbeitselementliste	I	I	S	R	I	S/A
• Priorisierung der Arbeitselementliste	S	I	S	R/A	C	C
Abschluss	BM	PM	TeCo	PrOw	ArOw	ATeM
Projektabschlussbericht	S	R	S	I	I	I

2. Artefaktlandschaft der PM²-Methodik und PM²-Agile



3. Überblick über die PM²-Agile-Aktivitäten und -Artefakte



4. Begriffsverzeichnis

Das nachfolgende Verzeichnis dient als Nachschlagewerk für die Begrifflichkeiten, die aus dem PM²-Guide „PM²-Agile“ (Guide 3.0.1) aus dem Englischen ins Deutsche übersetzt wurden.

Original (EN)	Übersetzung (DE)
Agile Logs	<i>Agile-Protokolle</i>
Agile Project Core Team (A-PCT)	<i>agiles Projektkernteam (A-PCT)</i>
Agile Team Members (ATeM)	<i>agile Teammitglieder (ATeM)</i>
Architecture Overview	<i>Systemarchitekturübersicht</i>
Architecture Owner (ArOw)	<i>Systemarchitekt, Systemarchitektin (ArOw)</i>
Backlog	<i>Backlog</i>
Burndown Chart	<i>Burndown-Chart</i>
Burnup Chart	<i>Burnup-Chart</i>
Ceremonies	<i>Events</i>
Collocation	<i>Kollokation</i>
Commitment Agreement	<i>Verpflichtungserklärung</i>
Concurrent Testing	<i>Concurrent Testing</i>
Daily Stand-up (meeting)	<i>Daily Stand-up</i>
Definition of Done (DoD)	<i>Definition of Done (DoD)</i>
Deployment Plan	<i>Bereitstellungsplan</i>
Development Handbook	<i>Entwicklungshandbuch</i>
Development Status Report	<i>Entwicklungsstatusbericht</i>
Development Work Plan	<i>Entwicklungsarbeitsplan</i>
Domain Expert	<i>fachkundige Person</i>
Enabler Items	<i>befähigende Elemente</i>
Iteration Burndown Chart	<i>Burndown-Chart der Iteration</i>
Iteration Goal	<i>Iterationsziel</i>
Iteration Plan	<i>Iterationsplan</i>
Iteration Planning	<i>Iterationsplanung</i>
Iteration Planning meeting	<i>Iterationsplanungsmeeting</i>
Iteration Retrospective	<i>Iterationsretrospektive</i>
Iteration Review	<i>Iterationsreview</i>
Minimum viable product (MVP)	<i>Minimum Viable Product (MVP)</i>
Operational Model	<i>Geschäftsmodell</i>
Pair Programming	<i>Pair Programming</i>
Planning	<i>Planning</i>
Planning Poker	<i>Planning Poker</i>
Product Owner (PrOw)	<i>Product Owner (PrOw)</i>
Release Burndown Chart	<i>Release Burndown-Chart</i>
Release Plan	<i>Releaseplan</i>
Release Planning	<i>Release Planning</i>
Retrospectives Log	<i>Protokoll der Retrospektive</i>
Software Configuration Management	<i>Software-Konfigurationsmanagement</i>
Story Points	<i>Story Points</i>
Team Coordinator (TeCo)	<i>Teamkoordinator, Teamkoordinatorin (TeCo)</i>
Test Management	<i>Testmanagement</i>
Test Plan	<i>Testplan</i>

Original (EN)	Übersetzung (DE)
Test-Driven Development (TDD)	<i>Testgetriebene Entwicklung</i>
Testing Logs	<i>Testprotokolle</i>
User Story	<i>User Story</i>
Vote of Confidence	<i>Vertrauensvotum</i>
Work Items	<i>Arbeitselemente</i>
Work Items List (WIL)	<i>Arbeitselementeliste (WIL)</i>

5. Abbildungsverzeichnis

Abbildung 1: PMflex – Das ganzheitliche Projektmanagementsystem.....	1
Abbildung 2: Vision und Synergien von PM ²	3
Abbildung 3: Herausforderungen von Agilität in der öffentlichen Verwaltung	5
Abbildung 4: Werte und Prinzipien des agilen Manifests	7
Abbildung 5: Agiler Iterationszyklus im PNR-Projekt im BVA	10
Abbildung 6: Rollen des Programms Digitale Verwaltung Niedersachsen	12
Abbildung 7: Minimierung der Gesamtzeit durch die Schleife	24
Abbildung 8: Was ein MVP ist und was es nicht ist	25
Abbildung 9: Das richtige MVP erfüllt alle Elemente.....	25
Abbildung 10: Elemente von PM ² -Agile.....	28
Abbildung 11: Elemente von PM ² -Agile im Kontext von der PM ² -Methodik.....	31
Abbildung 12: Einordnung von PM ² -Agile in die Organisationsumgebung.....	32
Abbildung 13: Agile Teams im Kontext der PM ² -Projektorganisation	33
Abbildung 14: Projektphasen von PM ²	33
Abbildung 15: PM ² -Projektlebenszyklus: Überschneidung phasenspezifischer Aktivitäten	34
Abbildung 16: CIR-Rhythmus von PM ² -Agile.....	34
Abbildung 17: PM ² -Projektlebenszyklus mit Iterationen der Aktivitäten: starke Überschneidung phasenspezifischer Aktivitäten.....	35
Abbildung 18: Übersicht über die Aktivitäten und Artefakte von PM ² -Agile in den einzelnen Phasen.....	36
Abbildung 19: Von den Projektphasen zu den täglichen Zyklen.....	36
Abbildung 20: Verfügbarmachen von Arbeitselementen.....	38
Abbildung 21: Lösung „wächst“ zu einem Release.....	39
Abbildung 22: (Kombinierte) Artefaktlandschaft von PM ² -Agile.....	44
Abbildung 23: Lean UX-Prozess von PM ² -Agile.....	48
Abbildung 24: Kommunikation.....	52
Abbildung 25: Beziehung zwischen Features und Storys.....	54
Abbildung 26: Schritte des Software-Konfigurationsmanagements.....	65
Abbildung 27: Bestandteile einer Iterationsplanung.....	70
Abbildung 28: Vereinbarung des Iterationsziels.....	71
Abbildung 29: Die drei wichtigsten Schritte des Release Planning	79
Abbildung 30: Struktur eines Iterationsreviews	86
Abbildung 31: PM ² -Projektorganisation.....	98
Abbildung 32: Primäre Rollen des agilen Projektkernteams (A-PCT).....	99
Abbildung 33: (Kombinierte) Artefaktlandschaft der PM ² -Methode und PM ² -Agile	108
Abbildung 34: Business Case — Input und primäre Rollen.....	109

Abbildung 35: Projektauftrag – Input und primäre Rollen.....	109
Abbildung 36: Systemarchitekturübersicht – Input und primäre Rollen.....	112
Abbildung 37: Geschäftsmodell – Input und primäre Rollen.....	114
Abbildung 38: Entwicklungshandbuch: Input und primäre Rollen.....	116
Abbildung 39: Entwicklungsarbeitsplan – Input und primäre Rollen.....	118
Abbildung 40: Entwicklungsarbeitsplan.....	119
Abbildung 41: Arbeitselementeliste (WIL) – Input und primäre Rollen.....	121
Abbildung 42: Releaseplan– Input und primäre Rollen.....	124
Abbildung 43: Iterationsplan – Input und primäre Rollen.....	126
Abbildung 44: Testplan – Input und primäre Rollen.....	128
Abbildung 45: Bereitstellungsplan – Input und primäre Rollen.....	131
Abbildung 46: Entwicklungsstatusbericht: Input und primäre Rollen.....	133