

TEXTE

79/2023

Abschlussbericht

Umsetzungskonzept umwelt.info

Anhang B: Agiles Vorgehensmodell

von:

Dr. Matthias Bluhm, Georg Börner, Ruth Busch, Rupert Pfeiffer, Lars Schneider
con terra GmbH, Münster

Herausgeber:

Umweltbundesamt

TEXTE 79/2023

Ressortforschungsplan des Bundesministeriums für
Umwelt, Naturschutz und nukleare Sicherheit

Forschungskennzahl 3720 12 101 0

FB001072

Abschlussbericht

Umsetzungskonzept umwelt.info

Anhang B: Agiles Vorgehensmodell

von

Dr. Matthias Bluhm, Georg Börner, Ruth Busch, Rupert
Pfeiffer, Lars Schneider
con terra GmbH, Münster

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber

Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

[f/umweltbundesamt.de](https://www.facebook.com/umweltbundesamt.de)

[t/umweltbundesamt](https://twitter.com/umweltbundesamt)

Durchführung der Studie:

con terra GmbH
Martin-Luther-King-Weg 20
48155 Münster

Abschlussdatum:

November 2022

Redaktion:

Fachgebiet I 1.7 Umweltinformationssysteme/-dienste, Satellitenfernerkundung,
Dateninfrastruktur
Marco Hohmann

Fachgebiet I 1.5 Nationale und internationale Umweltberichterstattung
Stefan Krämer

Mit Unterstützung von:

Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit Referat Z III 4
Umweltinformationen, Chief Data Officer, Künstliche Intelligenz
Birte Solveig Kulla

Publikationen als pdf:

<http://www.umweltbundesamt.de/publikationen>

ISSN 1862-4804

Dessau-Roßlau, Juni 2023

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen und Autoren.

Inhaltsverzeichnis

Abbildungsverzeichnis.....	7
Tabellenverzeichnis.....	7
Abkürzungsverzeichnis.....	8
1 Einleitung und Zielsetzung.....	9
2 Agiler Prozess.....	10
2.1 Kennzeichen agiler Vorgehensmodelle.....	10
2.2 Mögliche Vorgehensmodelle.....	10
2.2.1 Scrum.....	11
2.2.2 Kanban.....	11
2.2.3 Extreme Programming.....	11
2.2.4 Kombiniertes Vorgehensmodell.....	12
2.3 Bewertung der Vorgehensmodelle aus Sicht von umwelt.info.....	12
2.4 Erforderliche Ergänzungen.....	15
3 Anpassung des agilen Prozesses für die umwelt.info-Realisierung.....	16
3.1 Gesamtteam.....	16
3.1.1 Aufgaben.....	16
3.1.2 Kompetenzen im Team.....	17
3.1.3 Rollen und Kompetenzen.....	19
3.1.4 Organisationseinheiten.....	22
3.1.5 Struktur des Gesamtteams.....	22
3.2 Ereignisse.....	25
3.2.1 Sprintplanung.....	26
3.2.1.1 Vorbereitung der Sprintplanung.....	27
3.2.1.2 Sprintplanung im Gesamtteam.....	27
3.2.1.3 Sprintplanung im Team.....	28
3.2.2 Regelmäßige Abstimmung.....	29
3.2.2.1 Tägliche Abstimmung im Team.....	29
3.2.2.2 Wöchentliche Abstimmung im Gesamtteam.....	30
3.2.2.3 Backlogpflege.....	30
3.2.3 Abschließende Betrachtung.....	32
3.2.3.1 Review im Team.....	32

3.2.3.2	Review im Gesamtteam.....	33
3.2.3.3	Retrospektive im Team.....	34
3.2.3.4	Retrospektive im Gesamtteam.....	34
3.2.4	Sonstige Ereignisse.....	35
3.3	Artefakte	35
3.3.1	Produktvision	36
3.3.2	Systemkontext	36
3.3.3	Roadmap.....	37
3.3.4	Glossar	37
3.3.5	Product-Backlog.....	37
3.3.6	Sprint-Backlog.....	39
3.3.7	Inkrement.....	41
3.4	Ablauf des Projekts	41
3.5	Methoden	43
3.5.1	Anforderungs-Management	43
3.5.2	Planung	44
3.5.3	Coaching, Feedbackschleifen und Meetings.....	45
3.5.4	DevOps & Software Craftsmanship	46
3.6	Werkzeuge	47
3.6.1	Kommunikation.....	47
3.6.2	Backlog Management	48
3.6.3	DevOps & Software Craftsmanship	48
3.6.4	Visualisierung, Modellierung und Dokumentation.....	48
3.7	Eskalationsmechanismen.....	49
3.8	Vertragsgestaltung.....	50
3.8.1	Rahmenvereinbarung	50
3.8.2	Zusammenwirken mit dem Vorgehensmodell.....	50
4	Quellenverzeichnis	52

Abbildungsverzeichnis

Abbildung 1:	Rollen im Gesamtteam und ihre Verteilung auf Organisationseinheiten	25
Abbildung 2:	Schematischer Aufbau des Product-Backlogs	31
Abbildung 3:	Zusammenhang von Product-Backlog und Sprint-Backlogs	39
Abbildung 4:	Ableiten von Einträgen des Sprint-Backlogs.....	40
Abbildung 5:	Ablauf des Gesamtprojekts	41
Abbildung 6:	Verschiedene Zyklen bei der Entwicklung.....	42
Abbildung 7:	Zyklen im Entwicklungsprozess	51

Tabellenverzeichnis

Tabelle 1:	Bewertung der drei Vorgehensmodelle	14
Tabelle 2:	Aufgabenbereiche und Tätigkeiten	16
Tabelle 3:	Zuordnung der Rollen zu Organisationseinheiten.....	23

Abkürzungsverzeichnis

AG	Auftraggeber*in
AN	Auftragnehmer*in
BMUV	Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz
DSGVO	Datenschutz-Grundverordnung
ggf.	gegebenenfalls
PBI	Product-Backlog Item
UBA	Umweltbundesamt
UCD	User Centered Design
XP	Extreme Programming

1 Einleitung und Zielsetzung

Die Ergebnisse der Machbarkeitsstudie [1] zeigen, dass Entwicklung und Betrieb vom "Umwelt- und Naturschutzinformationssystem Deutschland" (umwelt.info) eine außerordentlich hohe Komplexität aufweisen und dass mehrere Organisationen mit Entwicklung und Betrieb beschäftigt sein werden. Des Weiteren ist zu erwarten, dass das Anforderungsprofil noch angepasst und erweitert wird.

Dies lässt ein agiles Vorgehensmodell besonders geeignet erscheinen. Ziel dieses Dokuments ist die Auswahl und bei Bedarf die Anpassung eines geeigneten Vorgehensmodells für umwelt.info.

Dafür werden zunächst agile Prinzipien (siehe Kap. 2.1) und verschiedene gängige Modelle beschrieben (siehe Kap. 2.2) und im Hinblick auf die Anwendung für umwelt.info verglichen und bewertet (siehe Kap. 2.3).

Da von vornherein nicht zu erwarten war, dass ein bestehendes Vorgehensmodell unverändert für die Realisierung eines Projekts der Größenordnung von umwelt.info verwendet werden kann, wurden Anpassungen an die Erfordernisse von umwelt.info erarbeitet (siehe Kap. 3).

Ein agiles Vorgehen bei der Projektumsetzung erfordert die Schaffung geeigneter vertraglicher Grundlagen. Aus diesem Grund sollte eine Rahmenvereinbarung entworfen werden, die es ermöglicht, das beschriebene Vorgehensmodell im Auftrag des Umweltbundesamtes anzuwenden. Aspekte zur Vertragsgestaltung und der Bezug zum Vorgehensmodell sind in Kap. 3.8 beschrieben.

2 Agiler Prozess

Für die Umsetzung von umwelt.info soll ein agiles Vorgehensmodell entworfen werden. Gründe für ein agiles Vorgehensmodell gegenüber einem traditionellen sind die Komplexität des Vorhabens [2], die Beteiligung verschiedener betroffener Stellen sowie die zu erwartende Dynamik in den Anforderungen an umwelt.info.

Agile Vorgehensmodelle zeichnen sich durch eine hohe Transparenz bei der Entwicklung, durch eine hohe Effizienz bei der Reaktion auf geänderte Anforderungen und durch einen geringeren Anteil von Fehlentscheidungen aus [3]. Das führt generell zu einer höheren Akzeptanz bei den Nutzenden, zu einer gesteigerten Qualität des erstellten Produkts und zu insgesamt geringerem Investitionsbedarf [4].

Andererseits muss berücksichtigt werden, dass agile Vorgehensmodelle von allen Beteiligten starkes Engagement und kontinuierliche Beteiligung erfordern. Insbesondere ist es notwendig, dass auf Seiten der auftraggebenden Stelle feste Ansprechpersonen zur Verfügung stehen. Dies muss durch eine entsprechende Kapazitätsplanung sichergestellt werden.

Nachfolgend werden zunächst agile Vorgehensmodelle kurz beschrieben und auf die Verwendbarkeit in umwelt.info überprüft. Anschließend wird ein auf die Bedürfnisse von umwelt.info angepasster agiler Prozess erarbeitet und in Kap. 3 beschrieben.

2.1 Kennzeichen agiler Vorgehensmodelle

Agile Vorgehensmodelle zeichnen sich insbesondere durch inkrementelles Vorgehen in kleinen Schritten aus [5]. Es wird versucht, möglichst schnell eine erste Version des zu erstellenden Produkts zu realisieren und dieses kontinuierlich zu verbessern und zu vervollständigen. Nach jeder Iteration erfolgt neben der Planung der nächsten Phase eine kritische Beurteilung der vorhergehenden Phase. Auf diese Weise werden Optimierungspotenziale im Prozess schnell erkannt und können frühzeitig genutzt werden.

Relativ geringer Zeitaufwand für jede Iteration und häufige Beurteilungszyklen sowie ein transparenter Umgang mit Erkenntnissen begünstigen ein empirisches Vorgehen bei der Suche nach Lösungen für bestimmte Teilprobleme, führen zu Lerneffekten und erleichtern die Reaktion auf Änderungen in den Anforderungen [6].

Das Bestreben, Entscheidungen möglichst spät zu fällen, nämlich wenn sie im Interesse des Arbeitsfortschritts nicht mehr hinausgezögert werden können, führt zu einer breiteren Entscheidungsgrundlage und somit zu einer geringeren Zahl von Fehlentscheidungen im Projektverlauf.

Wichtiges Kennzeichen aller agilen Prozesse ist auch die Anpassbarkeit an sich ändernde Anforderungen und Rahmenbedingungen. In unterschiedlichen Projektphasen sollte der verwendete Prozess kritisch evaluiert und ggf. angepasst oder sogar grundlegend geändert werden. Dies ist z. B. der Fall, wenn sich der Schwerpunkt des Projekts von der Entwicklung hin zum Betrieb verlagert [7].

2.2 Mögliche Vorgehensmodelle

Unter den gängigen agilen Vorgehensmodellen gibt es drei, die als Basis für die Entwicklung von umwelt.info besonders geeignet erscheinen. Diese drei Modelle konzentrieren sich auf den Kern der Realisierung, also die Umsetzung von Entwicklungsaufgaben. Sie besitzen einen hohen

Bekanntheitsgrad, so dass vielfach Kenntnisse bereits vorhanden sind bzw. Informationen über diese Vorgehensmodelle leicht zu beschaffen sind. Sie sind in der Praxis bewährt und an die spezifischen Anforderungen von umwelt.info anpassbar. Diese drei Vorgehensmodelle werden im Folgenden kurz beschrieben: Scrum [8], Kanban [9] und Extreme Programming [10].

2.2.1 Scrum

Das markanteste Kennzeichen des Scrum-Vorgehensmodells ist der feste Iterationstakt in Form von sogenannten Sprints. In einem Sprint wird im Team geplant, welche Arbeitspakete innerhalb des Sprints bearbeitet werden. Nach dem Sprint wird eine neue Version des Produkts ausgeliefert. Außerdem werden nach jedem Sprint die erarbeiteten Inhalte und die Arbeitsweise kritisch beurteilt [8].

In Scrum werden bestimmte Mengen von Arbeitspaketen in Form von Listen verwaltet. Diese sind das Product-Backlog, das alle bekannten Arbeitspakete enthält und das Sprint-Backlog, in dem die aktuell bearbeiteten Positionen enthalten sind. Als Ergebnis jedes Sprints entsteht ein grundsätzlich funktionstüchtiges Produkt, das sogenannte Inkrement [11]. Dieses und die beiden Backlogs werden neben anderen Dokumenten als Artefakte bezeichnet.

Kennzeichnend sind auch Besprechungen mit definiertem Fokus. Dies sind regelmäßige Treffen, um den Arbeitsfortschritt zu diskutieren und um Probleme möglichst frühzeitig zu erkennen (Daily Scrum), Treffen, um einen abgelaufenen Sprint und die bisherige Arbeitsweise zu beurteilen (Review und Retrospektive) sowie Treffen, um die Inhalte des kommenden Sprints festzulegen (Planning). Diese Treffen bezeichnet man im Scrum-Kontext auch als Ereignisse.

Scrum kennt drei verschiedene Rollen: Für die Pflege des Product-Backlogs und für die Priorisierung von Arbeitspaketen den Product Owner, für die Abarbeitung von Arbeitspaketen die Developer und für die Einhaltung des Entwicklungsprozesses und ggf. die Moderation von Treffen den Scrum Master.

2.2.2 Kanban

Im Gegensatz zu Scrum mit seiner festen Taktung durch Sprints zeichnet sich Kanban durch einen stetigen Arbeitsfluss aus (Flow). Einzelne Arbeitspakete werden aus einem Gesamtpool entnommen und durchlaufen den Realisierungsprozess, um sofort nach der Fertigstellung in das Produkt integriert zu werden. Der Arbeitsfluss wird auf einem kontinuierlich gepflegten Tableau visualisiert (Kanban Board). Dieses Tableau besitzt Spalten für die einzelnen Prozessschritte. Jede dieser Spalten ist in Kapazität (Work in Progress Limit) begrenzt, um Staus zu vermeiden [9].

Kanban kennt keine vorgegebenen Rollen. Allerdings werden bei konkreten Anwendungen häufig bereits bestehende Rollen genutzt und fortlaufend weiterentwickelt, z. B. für die Priorisierung von Arbeitspaketen. Die Ereignisse aus Scrum haben in Kanban keine Entsprechung. Jedoch erfordert ein Vorgehen nach Kanban ebenso wie Scrum die intensive Abstimmung innerhalb des Entwicklerteams sowie Feedbackschleifen für eine kontinuierliche Verbesserung des Vorgehens [12].

2.2.3 Extreme Programming

Beim Extreme Programming (XP) [10] steht die eigentliche Codeerstellung im Vordergrund. Die Bereiche des Projekt- und des Anforderungsmanagements bleiben weitgehend unberücksichtigt. XP richtet sich vor Allem an die technisch orientierten Mitglieder des Projektteams und versucht

mit seinen Regeln, eine hohe Codequalität sicherzustellen. Ähnlich zu Kanban sieht XP einen kontinuierlichen Arbeitsfluss mit stetiger Aktualisierung des Produkts vor [13]. Auch hier gibt es keine definierten Rollen und Ereignisse. Dennoch ist ausreichend Kapazität für die Kommunikation innerhalb des Projektteams und mit dem Auftraggeber vorzusehen.

2.2.4 Kombiniertes Vorgehensmodell

Um den speziellen Anforderungen und Rahmenbedingungen eines bestimmten Projekts gerecht zu werden ist es nicht verwerflich, keines der existierenden Vorgehensmodelle unverändert zu übernehmen. Vielmehr ist es sinnvoll, auf der Basis der bewährten Ansätze einen eigenen Prozess zu entwickeln und dabei Elemente der bekannten Modelle zu kombinieren. Gerade bei umfangreichen Projekten werden Elemente verschiedener Vorgehensmodelle kombiniert. Auf diese Weise entsteht ein Vorgehensmodell, das speziell auf ein bestimmtes Projekt angepasst ist und den spezifischen Anforderungen bestmöglich gerecht wird. Dabei ist vor Allem die strategische Entscheidung von Bedeutung, ob man einen stetigen Arbeitsfluss nach Kanban oder XP oder ein getaktetes Vorgehen in festen Iterationen nach Scrum bevorzugt.

2.3 Bewertung der Vorgehensmodelle aus Sicht von umwelt.info

Folgende grundlegende Anforderungen und Rahmenbedingungen sind beim Entwurf des Vorgehensmodells für die Umsetzung von umwelt.info zu berücksichtigen:

- ▶ **Anwendbarkeit auf Arbeitsschritte in den Ausbaustufen 1 – 4:**
Die Anforderungen aus den Ausbaustufen 1-4 haben einen unterschiedlichen Detaillierungsgrad und eine unterschiedliche Spezifikationsreife. Gerade aus den Ausbaustufen 3 und 4 sind auch Anforderungen zu erwarten, die heute noch nicht absehbar sind. Das zu entwickelnde Vorgehensmodell muss die nötige Flexibilität aufweisen, um mit solchen Anforderungen umzugehen.
- ▶ **Umfassende Abdeckung des agilen Prozesses:**
Alle Aspekte des Umsetzungsprojekts (Anforderungsanalyse, Anforderungsmanagement, Budgetplanung, Personalplanung, Realisierung, Test, Betrieb) müssen abgedeckt werden.
- ▶ **Anpassbarkeit und Erweiterbarkeit:**
Neue oder bisher nicht ausreichend spezifizierbare Anforderungen, insbesondere aus den Ausbaustufen 3 und 4 erfordern möglicherweise Anpassungen am Vorgehensmodell. Dies kann die Einführung neuer Rollen, neuer Werkzeuge aber auch die gezielte Neudefinition von Ereignissen oder Dokumenten (z. B. dem Product-Backlog) umfassen.
- ▶ **Skalierbarkeit:**
Die Realisierung von umwelt.info wird nicht von einem Entwicklerteam allein zu leisten sein. Daher wird es verschiedene Teams mit unterschiedlichem Schwerpunkt geben. Die Arbeit dieser Teams muss inhaltlich und zeitlich fortlaufend koordiniert werden. Ebenso ist es notwendig, ihre Entwicklungsergebnisse im Gesamtsystem zu integrieren. Es ist darüber hinaus damit zu rechnen, dass die Zahl der Teams, ihre Zusammensetzung und ihr Tätigkeitsschwerpunkt im Verlauf des Projekts wechseln. Somit ist die Anzahl der Entwicklerteams eine wichtige Skalierungsgröße.
Anzahl und Zuständigkeiten der Teams werden zu Beginn des Realisierungsprojekts definiert. Aussagen, die dazu jetzt schon getroffen werden können, finden sich in Kap. 3.1.5.

- ▶ **Stabilität gegenüber personellen Veränderungen:**
Für den prognostizierten Entwicklungszeitraum von 5 Jahren kann nicht von einer unveränderlichen Personalausstattung ausgegangen werden. Auch werden in verschiedenen Phasen des Projektes Kapazitäten in unterschiedlichem Umfang benötigt.
- ▶ **Stabilität gegenüber schwankender Arbeitsintensität:**
Gerade in Projekten mit vielen beteiligten Organisationen und Personen muss damit gerechnet werden, dass nicht alle Ansprechpartner permanent zur Verfügung stehen oder dass einzelne Teams nicht an allen Sprints teilnehmen. Dies ist häufig nicht langfristig vorhersehbar, so dass diesen Problemen kurzfristig begegnet werden muss. Durch entsprechende Verteilung von Rollen und Verantwortlichkeiten sowie durch sinnvolle Stellvertreterregelungen muss dafür gesorgt werden, dass die planmäßigen Ereignisse stattfinden können und die Teams arbeitsfähig bleiben. Diese Regelungen dürfen nicht für die gesamte Projektlaufzeit festgeschrieben sein, sondern müssen bei Bedarf angepasst werden können.
- ▶ **Geringer Einarbeitungsaufwand:**
Es sollen bei den Projektbeteiligten möglichst bereits Kenntnisse über das Vorgehensmodell oder wesentliche Bestandteile davon vorhanden sein. Ist das nicht der Fall, so sollen diese Kenntnisse leicht erworben oder vermittelt werden können.
- ▶ **Unterschiedliche Interessengruppen (Politik, Behörden, Bund/Land/Kommunen):**
Die Vielzahl der potenziellen Nutzerprofile erfordert hohe Flexibilität bei der Spezifikation von Anforderungen und Arbeitspaketen. Änderungen an diesen Spezifikationen sind zu erwarten und müssen schnell und mit geringem Aufwand eingearbeitet werden können.
- ▶ **Vorgegebene Rollen und Zuständigkeiten im Projektkontext:**
Die Einbindung verschiedener Stellen kann es notwendig machen, Entscheidungen durch bestimmte Gremien mit Vertretern dieser Stellen treffen zu lassen. Es ist wichtig, die Entscheidungsfähigkeit frühzeitig zu klären. Welche Rollen mit welchen Zuständigkeiten vorgesehen sind, wird in Kap. 3.1.3 beschrieben.
- ▶ **Berücksichtigung einer gegebenen Ausschreibungs- oder Vergabesituation:**
Wenn Teilprojekte separat vergeben werden, so dürfen die Vergabeprozesse den Fortlauf des Gesamtprojekts nicht wesentlich verzögern. Bei der Bearbeitung anderer Arbeitspakete muss flexibel auf Verzögerungen in bestimmten Projektbereichen reagiert werden können.
- ▶ **Termine:**
Um den Fortlauf des Projekts kontinuierlich überprüfen zu können und um die beteiligten Stellen über Zwischenstände zu informieren sowie um Anforderungen auf Gültigkeit zu überprüfen und ggf. zu detaillieren ist es sinnvoll, Präsentationstermine zu planen. Diese Termine sollten jeweils ein bestimmtes Thema in den Vordergrund stellen. Sie müssen in einem gewissen Rahmen planbar sein, um die Teilnehmer rechtzeitig auswählen und einladen zu können.
- ▶ **Budget:**
Damit die Einhaltung des für das Umsetzungsprojekt vorgesehenen Budgets gesichert ist, muss der benötigte Aufwand für einzelne Teile des Gesamtsystems zumindest grob geschätzt werden. Das verbleibende Budget ist eine wichtige Größe für die Priorisierung der restlichen Aufgaben.

► **Arbeitsorganisation:**

Das Vorgehensmodell muss in hohem Maße geeignet sein, auch ohne gleichzeitige physische Präsenz der Beteiligten am Ort der Leistungserbringung sowohl für das Projektmanagement als auch für den eigentlichen Aufbau und Betrieb von umwelt.info verwendet zu werden (Home-Office-tauglich). Des Weiteren muss berücksichtigt werden, dass sich Teams ggf. aus Mitgliedern zusammensetzen, die unterschiedlichen Organisationen angehören.

Bewertet man die drei zugrunde gelegten Vorgehensmodelle anhand der aufgeführten Anforderungen, so ergibt sich folgendes Bild (vgl. Tabelle 1):

Tabelle 1: Bewertung der drei Vorgehensmodelle

Anforderung	Scrum	Kanban	Extreme Programming
Anwendbarkeit auf Arbeitsschritte in den Ausbaustufen 1-4 der Machbarkeitsstudie [1]	++	++	++
Abdeckung des Prozesses	++	+	-
Anpassbarkeit	++	+	/
Stabilität gegenüber personellen Veränderungen	+	++	+
Einarbeitungsaufwand	+	++	-
Berücksichtigung unterschiedlicher Interessengruppen	++	+	/
Berücksichtigung vorgegebener Rollen und Zuständigkeiten	++	+	/
Berücksichtigung der Ausschreibungs- und Vergabesituation	++	++	/
Planbarkeit von Terminen	++	-	/
Planbarkeit des Budgets	++	-	/

++ gut geeignet, + geeignet, - weniger geeignet, / keine Aussage sinnvoll

Während alle drei betrachteten Vorgehensmodelle gleichermaßen gut für die Umsetzbarkeit der funktionalen Inhalte des Realisierungsprojekts in den vier Ausbaustufen geeignet erscheinen, lässt sich die Erfüllung der meisten Anforderungen durch Extreme Programming nicht beurteilen. Durch die Eigenheit von XP, sich sehr stark auf die Codeerzeugung zu konzentrieren, sind viele der hier vorliegenden Anforderungen nicht abgedeckt. Das bedeutet nicht, dass keine Methoden aus XP im Vorgehensmodell für die Realisierung von umwelt.info berücksichtigt werden sollten. Aber als Grundlage für den Entwurf des gesamten Vorgehensmodells scheint Extreme Programming ungeeignet.

Scrum und Kanban zeigen sich beide als grundsätzlich geeignet, um als Grundlage für ein eigenes Vorgehensmodell verwendet zu werden. Allerdings gibt es bei Kanban Schwächen bezüglich der Planbarkeit von Terminen und des Budgets was hier ausschlaggebend ist.

Somit soll das Vorgehensmodell für die Realisierung von umwelt.info auf Scrum basieren. Welche Änderungen und Ergänzungen dabei noch vorgenommen werden sollen, wird im folgenden Abschnitt erläutert.

2.4 Erforderliche Ergänzungen

Scrum ist ein Vorgehensmodell, das allgemein die Arbeitsweise eines Teams beschreibt und organisiert. Dabei deckt es vor Allem den Kernbereich der Entwicklung ab. Weitere Bereiche wie das Projektmanagement, also die Kontrolle der Verfügbarkeit und des Verbrauchs von Kapazitäten und Budget werden bei der Definition der Rollen und Ereignisse nicht berücksichtigt. Ebenso fehlen Aussagen über den Bereich der Ermittlung neuer Anforderungen.¹

Für die Entwicklung und den späteren Betrieb von umwelt.info ist zu berücksichtigen, dass auf Seiten des Auftraggebers personelle Kapazitäten vorhanden sein müssen, die auch maßgeblich in die Entwicklung eingebunden werden sollen. Generell gilt es, die Arbeit verteilter Teams mit Mitgliedern aus verschiedenen Organisationen zu unterstützen. Auf Grund der Komplexität des Vorhabens wird es wahrscheinlich notwendig sein, zeitweise Experten für bestimmte Detailfragen einzubeziehen oder Anforderungen an das System in Workshops mit potenziellen Anwendern zu konkretisieren.

In Hinblick auf die frühzeitige Erstellung eines lauffähigen Prototyps unter realistischen Betriebsbedingungen sind neben den direkt an der Entwicklung beteiligten Stellen Organisationen für den Betrieb zu berücksichtigen. So sind sie auch in den Entwicklungsprozess einzu beziehen.

Als Vorgehensmodell für die Realisierung von umwelt.info ist Scrum grundsätzlich geeignet, in seiner reinen Form sind einige wichtige Anforderungen jedoch nicht abgedeckt. Die Elemente, die durch Scrum vorgegeben werden (Rollen, Ereignisse, Artefakte) müssen den besonderen Anforderungen dieses Projekts entsprechend angepasst und ggf. durch weitere Bestandteile ergänzt werden.

¹ Das hier beschriebene Vorgehensmodell bezieht sich auf den Kernbereich der Entwicklung. Personelle Kapazitäten für andere Bereiche wie das Projektmanagement müssen ebenfalls berücksichtigt werden (vgl. Kap. 6 des Hauptdokuments).

3 Anpassung des agilen Prozesses für die umwelt.info-Realisierung

Auf der Basis von Scrum wurde ein Vorgehensmodell entwickelt, das für die Realisierung von umwelt.info geeignet ist und das darüber hinaus auch den Übergang von der Entwicklung zum Betrieb berücksichtigt. In den folgenden Kapiteln werden zunächst die drei wesentlichen Elemente der Scrum-Methode (Team, Ereignisse, Artefakte) mit ihren Ergänzungen und Anpassungen beschrieben. Danach werden Empfehlungen für geeignete Werkzeuge und Methoden ausgesprochen. Das geht über das eigentliche Vorgehensmodell hinaus, hilft aber, den Ablauf der Entwicklung von umwelt.info konkreter darzustellen. Die getroffenen Festlegungen können geändert werden, wenn sich Anforderungen oder Rahmenbedingungen ändern. Dies kann z. B. beim Übergang von der Entwicklung zum Produktivbetrieb der Fall sein.

3.1 Gesamtteam

Das Gesamtteam umfasst den gesamten Personenkreis, der mit der Realisierung und später dem Betrieb von umwelt.info betraut ist. Dieses Gesamtteam wird im Folgenden beschrieben. Dazu werden zunächst seine allgemeinen Aufgaben aufgeführt und die dafür benötigten Kompetenzen erläutert. Diese Aufgaben werden anschließend zu Rollen zusammengefasst, die wiederum in einer Teamstruktur angeordnet bzw. den Organisationseinheiten zugeordnet werden.

3.1.1 Aufgaben

Hier werden diejenigen Aufgaben² in Bereiche zusammengefasst und deren Tätigkeiten beschrieben, die bei der Umsetzung von umwelt.info zu berücksichtigen bzw. durch Projektrollen abzudecken sind:

Tabelle 2: Aufgabenbereiche und Tätigkeiten

Aufgabenbereich	Tätigkeiten
Projektvision und -ziele	Formulierung des Projektziels in Form einer Projektvision (vgl. Kap. 3.3.1) Kommunikation des Projektziels und Herstellen der allgemeinen Verständlichkeit darüber
Anforderungsmanagement (Product-Backlog³ Management)	Erhebung und Analyse neuer Anforderungen, Entscheidung, ob Anforderungen mit Blick auf das Projektziel weiterverfolgt werden, Zuweisung von Anforderungen an Teams, Pflege der Product-Backlog Items (PBIs), Priorisierung nach Wert einer Anforderung für Nutzende, Konkretisieren von PBIs, Erstellen, Ändern, Löschen, Aufteilen von PBIs, Herstellen von Transparenz und allgemeinem Verständnis über das Product-Backlog und die PBIs. Dies muss sowohl gegenüber den Stakeholdern in Bezug auf Roadmap und Release Planung erfolgen als auch gegenüber den Entwickelnden.

² Die Aufgaben leiten sich teils aus der Machbarkeitsstudie und teils aus der Definition eines „Scrum Teams“ im Scrum Guide ab [8].

³ Liste der Anforderungen und Besonderheiten bezogen auf ein Produkt zu einem bestimmten Zeitpunkt.

Aufgabenbereich	Tätigkeiten
Entwicklung / Umsetzung von Product-Backlog Items ("Done" Inkremente)	Programmierung (Frontend & Backend, Full-Stack, ...), Softwarearchitektur (im agilen Kontext), Dokumentation, Testen und Sicherstellung der Qualität, User Experience und Design.
Infrastruktur	Bereitstellung, Betrieb, Wartung,
Moderation	Vorbereitung und Durchführung bzw. Sicherstellung von Teamereignissen wie z. B. Retrospektiven, Sprint Reviews, Daily Scrum
Coaching	Unterstützung bei Erlernen und Durchführen des Prozesses und bei der Selbstorganisation des Teams, Unterstützung beim Bestreben nach dem Liefern hoher Qualität und adressieren von Hindernissen oder Störungen adressieren, Unterstützung des Chief Product Owners bzw. der Product Owner bei der Pflege des Product-Backlogs, Abbauen von Barrieren zwischen Scrum Team und Stakeholdern.
Koordination	Release Planung, Koordination mehrerer Teams mit fachlichem Schwerpunkt, Synchronisierung von Sprints verschiedener Teams,
Projektmanagement	Überwachung des Budgets / Controlling, Planung der Kapazitäten, Leitung des Nationalen Zentrums für Umwelt- und Naturschutzinformationen.
Veranstaltungsmanagement	Organisation von Fach- oder Nutzer-Workshops
Redaktionelle Arbeiten	Erstellen von textlichen und/oder graphischen, tabellarischen o.a. Darstellungen zu aktuellen Themen, abgeleitet aus verfügbaren Daten, Nutzergruppenspezifische Aufbereitung von Daten und Informationen.
Komponentenbetreuung	Fachliche Administration, Technische Administration, Operative Zusammenarbeit mit Datenbereitstellern (z. B. zu Metadaten Index oder Daten Aufbereitung) Beispiele: Harvester, Crawler, ...

3.1.2 Kompetenzen im Team

Um die zuvor beschriebenen Aufgabenbereiche abzudecken und um die erforderlichen Tätigkeiten ausführen zu können, werden allgemeine und individuelle Kompetenzen vom umwelt.info-Projektteam gefordert. Die allgemeinen Kompetenzen beziehen sich im Wesentlichen auf den Kontext eines Teams. Die individuellen Kompetenzen werden im darauffolgenden Abschnitt Projektrollen zugeordnet und dort weiter detailliert.

Es ist damit zu rechnen, dass sich die Aufgabenschwerpunkte und damit auch die zentralen Rollen und Kompetenzen während der Gesamtlauzeit des Projektes verschieben, wenn ein fließender Übergang von vornehmlich Entwicklung hin zu Betrieb stattfindet.

Bei der Betrachtung individueller Kompetenzen spielt neben dem Ausgangszustand beim Projektstart auch die Weiterentwicklung eine Rolle. Für die Gestaltung eines kontinuierlichen Ausbaus von Kompetenzen können Kompetenzstufen (z. B. Novize, Praktiker, Experte) ebenso wie individuelle Motivatoren [14] und Bedürfnisse genutzt werden. Zudem stehen zahlreiche Weiterbildungsangebote zur Verfügung, mit denen auch auf eine Zertifizierung (wie sie z. B. die Scrum Alliance [15] oder Scrum.org [16] ausstellen) hingearbeitet werden kann.

Projektziele eines Projekts in der Größenordnung und Komplexität von umwelt.info lassen sich nur mit Hilfe von Teams erreichen, da nur durch die Arbeit im Team erforderliche Kompetenzen gebündelt werden sowie Austausch und Kreativität entstehen können. Grundsätzlich empfiehlt es sich, Teams divers⁴ und cross-funktional⁵ aufzusetzen und die benötigten Team-Kompetenzen „T-förmig“⁶ zu gestalten. Ein weiterer Aspekt der Teamarbeit ist die Verteilung von Wissen auf ausreichend viele Köpfe im Team, so dass beim Ausfall Einzelner oder bei Änderungen der Teamzusammensetzung essenzielles Wissen nicht verloren geht (auch als Truck- oder Bus Faktor bekannt⁷) [4].

Neben den individuellen Kompetenzen für die Ausübung von Aufgaben einer Projektrolle werden von allen Beteiligten Teamfähigkeit und ein grundlegendes Verständnis von agiler Denk- und Arbeitsweise (Agiles Mindset) erwartet. Dazu zählen wir vor allem folgende Aspekte:

- ▶ Gemeinsame Team-Verantwortung (z. B. für die Code-Basis),
- ▶ Fokus auf das Schaffen von Werten und engen Kundenkontakt,
- ▶ Respektieren der agilen/Scrum Werte (Engagement, Fokus, Offenheit, Respekt, Mut),
- ▶ Transparenz und eine offene Fehler- bzw. Lernkultur,
- ▶ Bereitschaft zu einer iterativen und inkrementellen Denk- und Arbeitsweise.

Ein weiterer Block allgemeiner Kompetenzen geht mit dem Agilen Mindset einher bzw. ergänzt dieses hinsichtlich der ganzheitlichen Betrachtung von Software-Entwicklung und deren Betrieb. Dieses sogenannte DevOps Mindset lässt sich wie folgt umreißen:

- ▶ Fokus auf Kommunikation & Feedback, Automation, Metriken & Messbarkeit,
- ▶ Möglichst realitäts- und zeitnah Entwicklung und Betrieb miteinander verbinden,
- ▶ Integration von Personen mit Schwerpunkt Betrieb ins Team,
- ▶ Berücksichtigung des Betriebs bei der Entwicklung.

Über diese allgemeinen (siehe oben) und die rollenbezogenen individuellen Kompetenzen (siehe 3.1.3) hinaus, können sich weitere Anforderungen an Wissen und Fähigkeiten des Teams aus dem konkreten technischen Kontext ergeben. Dieser wird in Kap. 3.2 des Hauptdokuments detaillierter beschrieben.

⁴ Verschiedenste Personen oder Persönlichkeiten fördern konstruktive und kreative Kollaboration.

⁵ Alle zur Erreichung der Ziele benötigten Funktionen werden durch Personen im Team bedient.

⁶ Nicht jeder ist in jedem Fachgebiet Experte und doch ist in einem breiten Bereich bei allen im Team erforderliches Basiswissen in relevanten Themenfeldern vorhanden.

⁷ Der Bus-Faktor beschreibt das Risiko durch nicht geteilte Informationen oder Fähigkeiten innerhalb eines Teams [64].

3.1.3 Rollen und Kompetenzen

Neben den oben erläuterten allgemeinen Kompetenzen mit Fokus auf eine Teamsicht ergeben sich aus den Aufgaben und Tätigkeiten zusammenfassende Rollen, die im Folgenden beschrieben werden. Dabei wird eine Rolle als Menge von Aufgaben oder Verantwortlichkeiten verstanden, die durch eine Person übernommen werden, die diese Rolle dann innehat. Das bedeutet nicht, dass eine Person nur eine Rolle haben kann. Gerade in kleineren Teams werden oft mehrere Rollen durch eine Person ausgefüllt. Auch in diesen Fällen sollte die Zuweisung der Rollen explizit erfolgen, so dass klar ist, bei welcher Person welche Zuständigkeit liegt und welches ihre Aufgaben sind.

Eine tabellarische Übersicht der Rollen findet sich in Kap. 3.1.4 bei ihrer Zuordnung zu Organisationseinheiten.

Projektleiter*in

Diese Rolle ist mit der organisatorischen und wirtschaftlichen Leitung des Projekts betraut, beschäftigt sich aber nicht mit den inhaltlichen Fragestellungen. Ihre Kernaufgaben bestehen in der Beschaffung und Verwaltung von Ressourcen sowie in der Kontrolle des Budgets. Dabei muss sie eng mit dem Chief Product Owner bzw. den Product Owners zusammenarbeiten.

Eine Projektleiter*in muss in jeder beteiligten Organisationseinheit vorhanden sein. Dort ist sie für alle Teams zuständig und dient als zentraler Ansprechpartner für die anderen Organisationseinheiten in organisatorischen und wirtschaftlichen Fragen.

Die Kompetenzen, die von dieser Rolle gefordert werden, sind Erfahrung mit dem agilen bzw. dem DevOps-Mindset, Erfahrung in der Bund-Länder-Zusammenarbeit auf Seiten des Auftraggebers, Erfahrung in der Zusammenarbeit mit Behörden auf Seiten des Auftragnehmers sowie Erfahrung bei der Leitung von komplexen IT-Projekten. Diese Rolle kann durch die Projektleitung umwelt.info ausgefüllt werden.

Chief Product Owner

Der Chief Product Owner steuert gemeinsam mit den Product Owners fachlich den Projektablauf. Dem entsprechend obliegt dieser Person die Verantwortung für das Product-Backlog, in dem alle Anforderungen an das System eingetragen sind. Der Chief Product Owner fällt die Entscheidungen zur Priorisierung von Anforderungen, wenn kein Konsens hergestellt werden kann. Er plant gemeinsam mit den Product Owners die Sprints der Teams und gibt diesen ihre Sprintziele vor. Ebenso plant er den Inhalt und die Inbetriebnahme von Produktversionen.

In dieser zentralen Rolle hat er die Aufgabe, die verschiedenen Teams zu koordinieren und die Sprintziele der Teams aufeinander abzustimmen.

Bei allen Tätigkeiten wird der Chief Product Owner von einem Gremium unterstützt, das aus den Product Owners der einzelnen Teams besteht und bei Bedarf durch weitere Teammitglieder, Fach- oder technische Expert*innen ergänzt wird.

Kompetenzen für diese Rolle sind Erfahrungen mit dem Agile- bzw. dem DevOps-Mindset und Erfahrung in der Bund-Länder-Zusammenarbeit. Der Chief Product Owner muss geübt im agilen Anforderungsmanagement sein. Darüber hinaus muss er teamfähig sowie entscheidungs- und kommunikationsstark sein.

Der Chief Product Owner ist Teil der Organisationsstruktur der Auftraggebenden Stelle. Die Rolle kann durch den*die Projektleiter*in IT von umwelt.info ausgefüllt werden.

Product Owner

Jedes Team hat einen Product Owner. Dieser stellt die Verbindung zum Chief Product Owner her. Er*Sie definiert und kommuniziert seinem*ihrem Team die Sprintziele, die gemeinsam mit den Chief Product Owner formuliert wurden. Dabei stellt er*sie sicher, dass ein gemeinsames Verständnis dieser Ziele vom Chief Product Owner bis hin zu den Developern vorhanden ist.

Während der Chief Product Owner das Product-Backlog als zentrales Anforderungsdokument pflegt, tut der Product Owner dies für das Sprint-Backlog seines Teams. Und ebenso wie der Product Owner bei seinen Tätigkeiten von seinem*ihrem Team unterstützt wird, unterstützt er*sie den Chief Product Owner. Er*Sie ist dafür verantwortlich, dass den Chief Product Owner relevante Informationen aus dem Team erreichen (z. B. Probleme bei der Budgeteinhaltung oder neue Aufgaben). Andererseits muss er*sie auch die Sprintziele für das Team kommunizieren und dafür sorgen, dass dieses aus den Zielen die richtigen Aufgaben ableitet.

Die notwendigen Kompetenzen ähneln denen des Chief Product Owners, allerdings hat der Product Owner keine Koordinationsaufgaben für verschiedene Teams. Die Rolle des Product Owners wird durch ein Teammitglied ausgefüllt.

Developer

Inhaber*innen dieser Rolle leisten die produktive Arbeit, indem sie die Aufgaben aus dem Sprint-Backlog umsetzen und damit neue Inkremente erzeugen. Als Mitglieder eines Teams haben Developer auch Aufgaben in den Bereichen Qualitätssicherung, Dokumentation, Datenpflege usw. Sie nehmen aktiv an der Sprintplanung teil und nehmen so direkt Einfluss auf die Entwicklung des Systems und die Umsetzung der Roadmap.

Die Rolle der Developer ist auch wichtig für die Aufwandsschätzung, die wiederum als Grundlage der Planungen von Product Owner und Chief Product Owner benötigt wird. Als Qualifikation sollte neben den notwendigen fachlichen bzw. technischen Kenntnissen Erfahrung mit agilen Vorgehensweisen und eine starke Teamorientierung vorhanden sein.

Scrum Master

Der Scrum Master begleitet das gesamte Projektteam als Coach, Moderator*in oder Trainer*in. Seine*Ihre Aufgabe ist die Sicherstellung der Durchführung der verschiedenen Ereignisse und bei Bedarf auch die Teilnahme daran. Er*Sie unterstützt methodisch bei der Selbstorganisation der Teams und achtet darauf, dass der Schwerpunkt und das Ziel der Ereignisse beachtet werden. Darüber hinaus unterstützt er*sie die Product Owner bei der Pflege der Backlogs.

Die Rolle des Scrum Masters muss nicht auf eine einzelne Person beschränkt sein. Gerade bei vielen verteilten Teams können auch mehrere Scrum Master eingesetzt werden. Andererseits muss auch nicht jedes Team exklusiv über einen eignen Scrum Master verfügen. So kann eine Person diese Rolle für mehrere Teams wahrnehmen. Dies kann auch in Form einer Dienstleistung für andere Organisationseinheiten geschehen.

Zu den Kompetenzen des Scrum Masters gehört Erfahrung mit agilen Methoden. Auch Erfahrung in der Softwareentwicklung ist notwendig. Der Scrum Master arbeitet autonom und verfügt über die notwendige Autorität, um als Coach oder als Moderator*in zu agieren.

Technischer Betrieb

Diese Rolle ist für den eigentlichen Betrieb der Systeme zuständig. Das betrifft nicht nur die eigentliche Produktionsumgebung, sondern auch Prototypen und Testsysteme in der Organi-

sationseinheit der auftraggebenden Stelle. Ausdrücklich ausgenommen davon ist der fachliche Betrieb.

Inhaber*innen dieser Rolle müssen Erfahrung mit dem Betrieb der verwendeten Basissoftware besitzen. Sie brauchen außerdem Kenntnis der Konzepte von Continuous Integration und DevOps.

Fachlicher Betrieb

Die Rolle des fachlichen Betriebs umfasst die inhaltliche Aktualisierung und Qualitätssicherung der Systeme. Diese Rolle sorgt dafür, dass die Datengrundlagen aktuell und konsistent sind. Sie ist auch in der Lage, neue inhaltliche Anforderungen, z. B. das Einrichten neuer Datendienste durchzuführen. Dazu muss sie operativ sowohl mit den Daten bereitstellenden Organisationen als auch mit dem technischen Betrieb zusammenarbeiten.

Als Qualifikation für diese Rolle sind Erfahrung in der Daten- / und Qualitätskontrolle notwendig. Vor allem inhaltliches, aber auch technisches Verständnis für umwelt.info sind ebenfalls eine Voraussetzung.

Redakteur*in

Die Redakteur*in bereitet Informationen über das Projekt oder Inhalte für die Datenbasis des Systems textuell oder grafisch auf. Inhaber*innen dieser Rolle beschäftigen sich mit Material für die Präsentation des Projekts nach außen und mit der Sicht auf die Daten im System selbst. Sie müssen Erfahrung im Umgang mit Content Management Systemen haben und Kompetenz in der Visualisierung von Sachverhalten besitzen. Außerdem müssen sie versiert im Umgang mit Fachthemen aus dem Umweltbereich und dem entsprechenden gesellschaftlichen Diskurs sein.

Veranstaltungsmanager*in

Diese Rolle ist zuständig für die Veranstaltung von Fach- oder Nutzerworkshops zur Klärung von Fragen im Rahmen der Entwicklung, aber auch für die Durchführung von Veranstaltungen zur Präsentation des Projekts. Dazu gehört die Teilnahme an Messen und Konferenzen aber auch die Durchführung von Presseterminen etc. Diese Rolle erfordert Organisations- und Kommunikationstalent.

Fachexpert*in

Inhaber*innen dieser Rolle arbeiten nicht direkt an der Entwicklung des Projekts mit, unterstützen aber mit ihrem Spezialwissen die Arbeit der Product Owner oder der Developer. Häufig nehmen Mitglieder des fachlichen Betriebs auch die Rolle der Fachexpert*in wahr.

Fachexpert*innen werden auch häufig als potenzielle Nutzende des Systems angesehen und als solche um die Beurteilung bestimmter Sachverhalte gebeten.

Fachexpert*innen benötigen viel Erfahrung in ihrer Fachdomäne und müssen dort auch gut vernetzt sein, um ggf. den Kontakt zu anderen Fachleuten herstellen zu können. Sie müssen ihr Wissen gut vermitteln können, damit die fachlich begründeten Anforderungen auf Seiten der Developer richtig verstanden werden.

Technologieexpert*in

Diese Rolle wirkt ähnlich wie die Fachexpert*in, hat aber den Schwerpunkt auf technologischen Themen. Während Fachexpert*innen häufig aus dem Umfeld des Auftraggebers kommen, werden Technologieexpert*innen oft durch den Auftragnehmer gestellt oder akquiriert. Auch sie müssen in der Lage sein, ihr Wissen effizient an Dritte weiterzugeben.

3.1.4 Organisationseinheiten

Das umwelt.info-Umsetzungsprojekt und sein späterer Betrieb finden im Kontext verschiedener Organisationseinheiten statt. In der einfachsten Betrachtungsweise sind dies der Auftraggeber (AG) und der oder die Auftragnehmer (AN).

Im Detail können weitere (Unter-) Einheiten, u.a. beteiligte Institutionen (z. B. BMUV) angeschlossen sein. Auch auf Seite des Auftragnehmers besteht die Möglichkeit, dass weitere Sub-Unternehmer beteiligt werden.

Für die Betrachtung der Projektrollen und den Überlegungen, in welcher Organisationseinheit diese angesiedelt werden, soll für das Vorgehen die grobe Unterscheidung in AG und AN genügen. Im folgenden Abschnitt wird eine genauere Zuordnung der einzelnen Rollen zu AG oder AN Seite vorgenommen.

Zentral für den Erfolg der umwelt.info-Umsetzungs- und Betriebsphase ist jedoch in jedem Fall eine enge Kooperation zwischen beiden Organisationseinheiten und im Sinne des agilen Manifests eine hohe Transparenz sowie starke Kommunikationskanäle innerhalb des Projektteams. Im Folgenden werden einige Faktoren genannt, die diese Kooperation – auch und insbesondere auf Team Ebene – maßgeblich positiv beeinflussen können und daher bei der Gestaltung des AG-AN Verhältnisses bedacht werden sollten:

- ▶ Synchroner Kommunikation (Teamgefühl, Motivation, Gesamtbild, Planung, Roadmap, ...),
- ▶ Asynchrone Kommunikation (Erreichbarkeit, Information, Unterstützung, schnellere Entscheidungen, ...),
- ▶ Ad-hoc Kommunikation (möglichst niedrige Schwelle durch Teamraum analog oder digital),
- ▶ Zugriff auf alle benötigten Ressourcen und Artefakte über Organisationsgrenzen hinweg,
- ▶ Organisationsübergreifende Teams / Integration von Betrieb / Informationsfluss,
- ▶ Delegation & Widerspruchsfreiheit von Verantwortungsbereichen (Entscheidungen, Teilnahme an Terminen, Aufgaben),
- ▶ Nutzung von Werkzeugen, die für die Arbeit in verteilten Teams geeignet sind und diese unterstützen (vgl. Kap. 3.6).

3.1.5 Struktur des Gesamtteams

Kombiniert man die Definition einzelner Rollen für das Projekt und die beteiligten Organisationseinheiten, so ergibt sich eine übergeordnete Sicht auf die Zuordnung oder Auf- bzw. Verteilung einer oder mehrerer Rollen in Bezug auf Organisationseinheiten. Dabei wird nicht vorab festgelegt, ob eine Person mehrere Rollen wahrnimmt oder eine Rolle auf mehrere Personen aufgeteilt wird.

Tabelle 3: Zuordnung der Rollen zu Organisationseinheiten⁸

Rolle	AN	AG	Hinweise
Projektleiter*in	+	+	Kooperation der Projektleitung auf AG und AN Seite
Chief Product Owner	-	+	Verantwortlich für die starke Vernetzung und Verzahnung auf AG Seite; enge Zusammenarbeit mit den Product Owners, Teams und Experten auf AN Seite
Product Owner	+	-	Enger Austausch und Zusammenarbeit mit dem Chief Product Owner (Arbeiten am Product-Backlog oder an der Roadmap) sowie mit anderen Product Owners (Kooperation mit anderen Teams)
Developer	+	(+)	Enge Zusammenarbeit vor allem mit dem eigenen mit Product Owner, ggf. auch mit Personen aus anderen Teams oder dem Chief Product Owner.
Scrum Master	+	(+)	Rolle nicht auf Organisationseinheiten aufteilbar; arbeitet nicht fachlich; Anzahl der Personen für diese Rolle ggf. an Anzahl der zu betreuenden Teams und an deren Größe ausrichten
Technischer Betrieb	(+)	+	Über AG Seite angebunden; möglichst enge Zusammenarbeit und Austausch mit den Developern und dem (Chief) Product Owner; Expertise zum Betrieb auf AN Seite wünschenswert
Fachlicher Betrieb	-	+	Fachliche Expertise vornehmlich auf AG Seite; enge Kooperation und Abstimmung mit den Product Owners auf AN Seite
Fachexpert*in	(+)	+	Sind nicht kontinuierlich involviert und können vornehmlich von AG Seite nach Bedarf eingebunden werden; fachliche Expertise auf AN Seite wird durch das Team / den Product Owner aufgebaut
Technologieexpert*in	+	+	Eine feste Zuordnung ist nicht erforderlich. Experten können aus dem Gesamtteam hinzugezogen werden (z. B. aus dem Betrieb auf AG Seite oder zu speziellen Themen auf AN Seite).
Redakteur*in	-	+	Enge Vernetzung auf AG Seite; hat als Ansprechpartner den Chief Product Owner aber auch die Product Owner auf AN Seite
Veranstaltungsmanager*in	+	+	Veranstaltungen mit Fokus eher auf der öffentlichkeitswirksamen Seite sind tendenziell eher beim AG angesiedelt; weitere (z. B. fachliche) Ereignisse auch durch AN organisierbar

Die Sicht auf externe Beteiligte (wie zum Beispiel Anwender*in/Nutzer*in) oder externe Dienstleister (wie z. B. ITZ BUND) werden für die Beschreibung der Rollen im Projekt

► ⁸ AN = Auftragnehmer; AG = Auftraggeber; + = verpflichtend; (+) = optional; - = nicht auf dieser Ebene

ausgeklammert und es wird angenommen, dass diese durch AG Prozesse in das Projekt eingebunden werden. Darunter fällt beispielsweise der Kontakt zu den Anwendenden / Endnutzenden des Projektergebnisses, im Wesentlichen durch den Chief Product Owner, z. B. in Form von Nutzerworkshops.

Im Folgenden bezeichnen wir als Team diejenigen Personen, die in den Rollen Developer und Product Owner unmittelbar, fortwährend und kooperativ an der Wertschöpfung und Umsetzung eines Teils des umwelt.info-Projekts beteiligt sind. Die Rollen Scrum Master und Projektleiter*in können ebenfalls integraler Teil des Teams sein, was aber im umwelt.info-Projektcontext nicht der Regel entspricht. Hier arbeiten Scrum Master und Projektleiter*in normalerweise für mehrere Teams einer oder sogar mehrerer Organisationseinheiten.

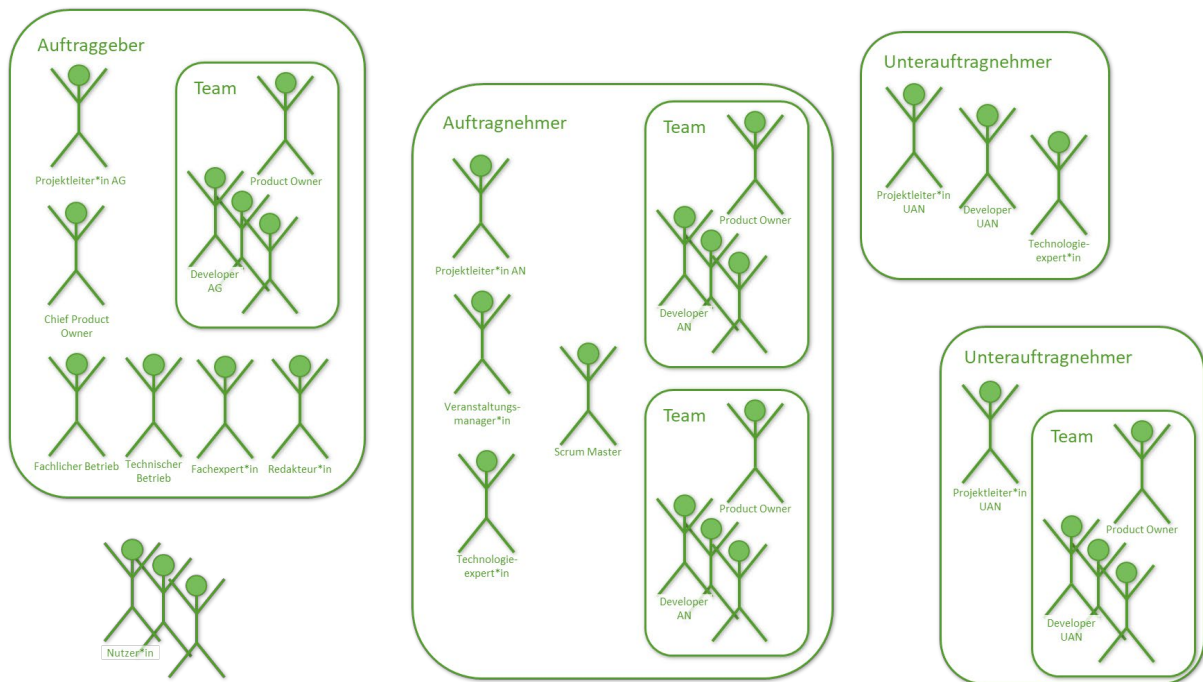
Ein Team sollte idealerweise die Anzahl von 10 Personen nicht übersteigen, damit eine effektive Kommunikation und produktive Arbeitsweise gewährleistet bleiben. Steigt die Anzahl der Personen, die am selben Projektziel arbeiten über 10 hinaus, empfiehlt sich die Reorganisation in mehrere Teams. Diese arbeiten abgegrenzt, mit eigenem fachlichem Ziel und eigenem Product Owner, orientieren sich am gemeinsamen Product-Backlog und kooperieren mit dem Chief Product Owner [8].

Für die Aufteilung der Teams kann die Architektur die Grundlage bilden (vgl. Anhang A2). So könnte jedem Team eine oder ggf. mehrere Komponenten des Systems für die Entwicklung zugeordnet werden. Komponenten, die besonders eng gekoppelt sind, könnten Teams zugewiesen werden, die in engem Kontakt zueinander stehen [17].

Abbildung 1 zeigt beispielhaft den Aufbau des gesamten umwelt.info-Projektteams. Die wesentlichen Organisationseinheiten sind als Auftraggeber und Auftragnehmer bezeichnet. Weitere Unterauftragnehmer sind aufgeführt. In der Organisationseinheit des Auftraggebers sind Projektleiter*in und Chief Product Owner zu finden. Sie haben die organisatorische, wirtschaftliche und inhaltliche Kontrolle über das Gesamtprojekt. Für den Betrieb des Systems, auch der ersten Prototypen in der Entwicklungsphase, stehen Mitarbeitende des fachlichen und technischen Betriebs bereit. Um Hintergrundinformationen für laufende Arbeiten zur Verfügung zu stellen, gibt es Fachexpert*innen. Für begleitende nicht-technisch geprägte Arbeiten stehen Redakteur*innen zur Verfügung. Die Fachexpert*innen stellen gleichzeitig (potenzielle) Nutzende des Systems dar. Weitere Nutzende finden sich außerhalb der Organisationseinheit. Sie können während der Entwicklung im Rahmen von Anforderungsanalysen eine Rolle spielen.

Auf Seiten des Auftragnehmers gibt es ebenfalls eine Projektleiter*in, die das Projekt organisatorisch und wirtschaftlich kontrolliert. Daneben gibt es für die Überwachung des Vorgehensmodells einen Scrum Master, der allen Organisationseinheiten zur Verfügung steht. Technologieexperten, die für bestimmte Aspekte der Entwicklung benötigt werden, sind ebenfalls auf Seiten des Auftragnehmers zu finden. Hier befindet sich auch die Mehrzahl der Teams, in denen die eigentliche Entwicklungsarbeit geleistet wird. Jedes dieser Teams besitzt einen eigenen Product Owner und eine Gruppe von Developern. Die Anzahl der Teams ist eine der Skalierungsgrößen, die im Projektverlauf genutzt werden können.

Abbildung 1: Rollen im Gesamtteam und ihre Verteilung auf Organisationseinheiten



Quelle: Eigene Darstellung, con terra GmbH

Eine zweite Möglichkeit der Skalierung besteht in der Einbindung von Unterauftragnehmern. Dabei ist nicht festgelegt, wie diese innerhalb ihrer eigenen Organisationseinheit organisiert sind. Dies kann analog zum hier beschriebenen Vorgehensmodell sein oder auch frei, ohne Product Owner. Gemeinsam ist allen Unterauftragnehmern jedoch, dass sie eine*n Projektleiter*in als zentrale*n Ansprechpartner*in benennen.

Eine Besonderheit bei der Organisation des umwelt.info-Gesamtteams ist, dass sich innerhalb der Organisationseinheit des Auftraggebers mindestens ein eigenes Team befindet. Dieses bearbeitet ebenso wie die Teams des Auftragnehmers Arbeitspakete aus dem Product-Backlog. Dabei wird der Schwerpunkt allerdings vermutlich auf inhaltlichen Arbeiten liegen, weniger auf technischen.

Ebenso ist es möglich, gerade vor dem Hintergrund verteilten Arbeitens, ein Team aus Mitgliedern verschiedener Organisationseinheiten zu bilden. So könnte ein Fachexperte des Auftraggebers als ständiges Teammitglied in einem Team des Auftragnehmers mitarbeiten.

3.2 Ereignisse

Eines der Kennzeichen von Scrum sind sogenannte Ereignisse. Dabei handelt es sich um regelmäßige Besprechungen des Teams, in denen ein bestimmter Aspekt des Entwicklungsprozesses gezielt behandelt wird. Im Rahmen der Realisierung von umwelt.info werden dies in der Regel die einzelnen Teams sein, nicht das Gesamtteam. Diese Konzentration auf bestimmte Aspekte stellt sicher, dass alle relevanten Themen behandelt werden und gewährleistet zudem, bei richtiger Durchführung, Effizienz und geringen Zeitbedarf für diese Besprechungen.

Scrum weist mit den Sprints, in denen die Entwicklungsarbeit abläuft, relativ kurze Prozesszyklen auf. Die Ereignisse orientieren sich im Wesentlichen an diesen Zyklen und fokussieren dabei auf folgende Schwerpunkte

- ▶ Planung des nächsten Sprints,
- ▶ Koordination bei der Durchführung und
- ▶ Abschließende Betrachtung.

Darüber hinaus gibt es noch weitere Ereignisse, die nicht dem zyklischen Ablauf folgen, sondern bei Bedarf initiiert werden. Dazu gehören z. B. Workshops zur Lösung bestimmter fachlicher Fragen oder Präsentationstermine für ausgewählte Gruppen oder die Öffentlichkeit.

In Bezug auf die Entwicklung von umwelt.info gibt es bestimmte Rahmenbedingungen zu beachten, die es erfordern, den Scrum-Prozess und damit auch die darin enthaltenen Ereignisse vom Standard abweichend zu gestalten. Diese Besonderheiten wurden bereits in Kap. 2.3 beschrieben. Sie sind der Grund dafür, dass im hier beschriebenen Vorgehensmodell zusätzliche Ereignisse definiert sind, die in Scrum an sich nicht vorgesehen sind.

Generell gilt, dass den einzelnen Teams ein hohes Maß an Selbstorganisation zugestanden wird. Ob ein Team für bestimmte Ereignisse den Scrum Master als Moderator einlädt oder ob es ein Teammitglied mit der Rolle eines eigenen Scrum Masters betraut, bleibt in der Regel dem Team selbst überlassen. Anders ist dies bei Ereignissen, die das Gesamtteam betreffen oder die auf Grund der Anzahl der Teilnehmer oder deren mangelnder Erfahrung mit Scrum-basierten Methoden möglicherweise nicht zielgerichtet durchgeführt werden. Bei solchen Ereignissen sollte der Scrum Master auf jeden Fall teilnehmen, um Effizienz und Zielerreichung der Termine sicher zu stellen.

Im Folgenden werden die Ereignisse im angepassten Scrum-Prozess beschrieben. Dabei sind sie nach den o.g. Schwerpunkten gegliedert.

3.2.1 Sprintplanung

Im herkömmlichen Scrum-Prozess ist die Sprintplanung ein Ereignis, bei dem das Entwickler-team den Schwerpunkt der nächsten Arbeitsphase formuliert und die Inhalte des folgenden Sprints festlegt. Bei der Umsetzung von umwelt.info ist jedoch mit mehreren parallelen Teams zu rechnen. Diese Teams weisen insgesamt eine relativ große Anzahl von Mitgliedern auf. Diese wiederum sind wahrscheinlich räumlich verteilt, verschiedenen Organisationen zugeordnet und mit unterschiedlichen Bereichen des Gesamtprojekts beschäftigt.

Um diese Teams bei der Sprintplanung angemessen zu berücksichtigen und effizient zu koordinieren ist es notwendig, das Ereignis der Planung selbst gut vorzubereiten. Die eigentliche Planung muss dann in zwei Detaillierungsniveaus erfolgen, nämlich auf der Ebene des Gesamtprojekts und auf der Ebene der Teams. Dem entsprechend werden für den Schwerpunkt der Sprintplanung drei Ereignisse vorgesehen.

Es ist wichtig, die Sprintplanung in der Gruppe der betroffenen Personen durchzuführen, weil in ihrem Rahmen sichergestellt werden muss, dass alle Beteiligten das gleiche Verständnis von den Aufgaben und Anforderungen haben. Missverständnisse in der Aufgabenstellung führen ansonsten zu fehlerhaften Entwicklungen und erhöhtem Aufwand. Gibt es zu viele Beteiligte, so ist es ausreichend, wenn nur Vertreter bestimmter Gruppen an der Planung teilnehmen.

Die Unterscheidung zwischen Gesamtteam und Teams im Kontext der umwelt.info-Entwicklung erfordert es, dass zwei Planungsebenen berücksichtigt werden. Auf Ebene des Gesamtteams wird die Sprintplanung durch den Chief Product Owner und die Product Owner der Einzelteams durchgeführt. Diese vertreten dabei jeweils ihre Teams. Auf der Ebene der Einzelteams

geschieht die Sprintplanung durch die Developer jedes Teams und den jeweiligen Product Owner. Durch diese Person wird die Verbindung zwischen den beiden Ebenen hergestellt.

3.2.1.1 Vorbereitung der Sprintplanung

Bei der Vorbereitung der Sprintplanung des Gesamtteams werden Aufgaben aus dem Product-Backlog für bestimmte Teams ausgewählt. Diese Aufgaben werden den Vertretern der Teams dann in der Sprintplanung für das Gesamtteam als Ziele für deren eigene Sprintplanung vorgeschlagen. Die Zuordnung der Aufgaben zu den Teams ist in der Regel einfach und ergibt sich von selbst aus den bearbeiteten Teilsystemen. Schwieriger ist die Priorisierung der Aufgaben und das Festlegen der Entwicklungsreihenfolge. Damit dabei nicht nur fachliche und politische Faktoren eine Rolle spielen, sondern auch technische Einflussgrößen berücksichtigt werden, ist die Teilnahme der Projektleiter*in auf Seiten des AG sowie, in seltenen Fällen, Vertretern einzelner Teams vorzusehen. Dennoch ist der Chief Product Owner die prägende Rolle dieses Ereignisses. Er legt seine Vorstellungen vom nächsten Entwicklungsschritt dar und formuliert die Zielvorstellung für den Sprint. Die anderen Teilnehmenden beraten in Hinblick auf die Realisierbarkeit dieses Ziels und auf die Auswahl und Priorisierung der Aufgaben. Grundlegende Informationen zu den Aufgaben werden gemeinsam gesammelt und zusammengestellt.

Wenn neue Aufgaben erkannt werden, werden sie vom Chief Product Owner in das Product-Backlog eingetragen und somit in den Entwicklungsprozess aufgenommen.

Grundlage für die Vorbereitung der Sprintplanung ist die regelmäßige Pflege des Product-Backlogs (vgl. Kap. 3.2.2.3). Diese erfolgt durch den Chief Product Owner.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Projektleiter*in AG
- ▶ Ggf. Projektleiter*in des AN oder Product Owner bestimmter Teams
- ▶ Ggf. Developer
- ▶ Ggf. Fachexpert*in

Aufgaben

- ▶ Identifizieren der nächsten Aufgaben
- ▶ Grobe Abschätzung des Aufwands
- ▶ Zusammenstellen von Informationen zu diesen Aufgaben
- ▶ Vorschlag für das Ziel des nächsten Sprints formulieren
- ▶ Identifizieren von neuen Aufgaben
- ▶ Pflege des Glossars

3.2.1.2 Sprintplanung im Gesamtteam

Basierend auf der Vorbereitung der Sprintplanung wird bei diesem Ereignis das Sprint-Backlog für das Gesamtteam erstellt. Zunächst wird das Sprintziel für das Gesamtteam formuliert.

Danach werden die Aufgaben ausgesucht, die im Sprint bearbeitet werden sollen. Diese Aufgaben werden dann den entsprechenden Teams als Ziel für ihre eigenen Sprints zugewiesen. Kann die Zuweisung nicht einvernehmlich erfolgen, entscheidet der Chief Product Owner.

Die eigentliche Zielformulierung der untergeordneten Sprints der Einzelteams erfolgt nicht im Rahmen dieses Ereignisses, sondern bei der detaillierten Sprintplanung der jeweiligen Teams. Allerdings werden diese Ziele durch die Auswahl der Aufgaben vorgegeben. Werden neue Aufgaben erkannt, so trägt sie der Chief Product Owner in das Product-Backlog ein.

Basis für die Sprintplanung im Gesamtteam ist die Vorbereitung durch den Chief Product Owner und die Projektleiter*in AG (vgl. Kap. 3.2.1.1).

Da hier die Aufgaben für alle Teams in der nächsten Entwicklungsphase festgelegt werden, sollten an der Besprechung neben dem Chief Product Owner, die Product Owner und ggf. weitere Vertreter der Teams teilnehmen. Dem Scrum Master obliegt die Moderation. Er hat auf die effiziente Durchführung des Ereignisses zu achten.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Product Owner
- ▶ Scrum Master
- ▶ Vertreter der Teams

Aufgaben

- ▶ Formulieren des Sprintziels
- ▶ Bestätigen der aktuellen Aufgaben
- ▶ Abschätzen des Aufwands für die Aufgaben
- ▶ Zuweisen der aktuellen Aufgaben als Ziel an Teams
- ▶ Identifizieren von neuen Aufgaben
- ▶ Pflege des Glossars

3.2.1.3 Sprintplanung im Team

Nach der teamübergreifenden Planung des nächsten Sprints treffen sich die einzelnen Teams und planen ihre eigenen Sprints. Dabei wird zunächst das Sprintziel für die nächste Entwicklungsphase des Teams formuliert. Anschließend werden die Aufgaben durch das Team selbst an die Developer verteilt. Falls im Rahmen der Planung neue Aufgaben identifiziert werden, so werden sie vom Product Owner vermerkt und an den Chief Product Owner weitergeleitet, um in die Pflege des Product-Backlogs einfließen zu können.

Die Länge von Sprints eines Teams muss nicht unbedingt der Sprintlänge für das Gesamtteam entsprechen. Sprints eines Teams können auch kürzer sein, wenn das Team dies als sinnvoll erachtet. So ist es z. B. denkbar, dass ein Sprint des Gesamtteams von zwei Wochen Länge zwei Sprints eines Teams umfasst. Das Team muss in einem solchen Fall die Aufgaben, die ihm im Rahmen der Planung des Gesamtsprints zugewiesen wurden, von vornherein auf zwei Sprints

verteilen. Die Teilnahme des Scrum Masters ist bei diesen Ereignissen nicht zwingend vorgesehen, die Moderation sollte dem Product Owner obliegen.

Teilnehmer

- ▶ Product Owner
- ▶ Developer
- ▶ Ggf. Scrum Master

Aufgaben

- ▶ Formulieren des Sprintziels
- ▶ Zuweisen der aktuellen Aufgaben an Developer
- ▶ Validieren der Aufwandsschätzungen
- ▶ Identifizieren und Dokumentieren von neuen Aufgaben

3.2.2 Regelmäßige Abstimmung

Um die Arbeitsintensität während der Entwicklungsphasen gleichbleibend hochzuhalten und um auftretende Probleme möglichst schnell zu erkennen, gibt es regelmäßige kurze Abstimmungen. Lösungen können so schnell gefunden werden.

Die parallele Arbeit verschiedener Teams bei der Entwicklung von umwelt.info erfordert auch hier Ereignisse auf verschiedenen Ebenen, ebenso wie bereits bei der Sprintplanung.

3.2.2.1 Tägliche Abstimmung im Team

Die Abstimmung im Team dient im Wesentlichen dazu, das Team über die unmittelbaren nächsten Schritte zu informieren. Dazu kann es sinnvoll sein, den Stand der Arbeiten festzustellen. Gefundene Probleme werden nicht eingehend diskutiert. Falls im Team nicht spontan eine Lösung bekannt ist, werden bestimmte Teammitglieder mit einer genaueren Analyse oder Bearbeitung beauftragt. Identifizierte neue Aufgaben werden in der Regel nur aufgenommen und vom Product Owner später an den Chief Product Owner weitergeleitet, der sie ggf. in das Product-Backlog übernimmt. In Ausnahmefällen besonderer Dringlichkeit kann eine neue Aufgabe allerdings auch direkt durch das Team in den laufenden Sprint aufgenommen werden.

Die bei den Sprintplanungen geschätzten Aufwände für die Durchführung der Aufgaben werden überprüft. Kann eine Aufgabe auf Grund eines Fehlers bei der Aufwandsschätzung nicht im aktuellen Sprint beendet werden, so muss dies vom Product Owner dem Chief Product Owner mitgeteilt werden, so dass es bei der weiteren Planung berücksichtigt werden kann.

Teilnehmer

- ▶ Product Owner
- ▶ Developer

Aufgaben

- ▶ Identifizieren von Problemen

- ▶ Finden von vorhandenen Ansätzen zur Problemlösung
- ▶ Feststellen von Verzögerungen und Fehlern bei der Aufwandsschätzung
- ▶ Identifizieren und Dokumentieren von neuen Aufgaben

3.2.2.2 Wöchentliche Abstimmung im Gesamtteam

Um die Arbeiten der Teams zu koordinieren, werden wöchentlich Besprechungen des Gesamtteams durchgeführt. Es wird nicht möglich sein, das gesamte Projektteam zu diesen Ereignissen einzuladen, deshalb senden die Teams Vertreter, in der Regel den Product Owner. Die Besprechung wird von Scrum Master und dem Chief Product Owner geleitet.

Im Ereignis wird der Arbeitsstand der Teams kurz dargelegt. Probleme und Fehler in der Planung werden angesprochen und es wird abgefragt, ob Ideen für eine Problemlösung im Gesamtteam existieren. Muss es zu engen Kooperationen zwischen Teams kommen (z. B. für Installationen, Datenlieferungen, ...), werden die notwendigen Kontakte durch Benennung von Verantwortlichen hergestellt. Werden neue Aufgaben identifiziert, werden sie vom Chief Product Owner aufgenommen, um ins Product-Backlog eingepflegt zu werden.

Aufgabe des Scrum Masters als Moderator*in ist es, die einzelnen Beiträge kurz zu halten und Diskussionen an bestimmte Personen zu verlagern, so dass diese nicht den Ablauf des Ereignisses verzögern.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Product Owner
- ▶ Scrum Master
- ▶ Ggf. weitere Vertreter der Teams

Aufgaben

- ▶ Information über den Arbeitsstand der Teams
- ▶ Identifizieren von Problemen
- ▶ Feststellen von Verzögerungen und Fehlern bei der Aufwandsschätzung
- ▶ Finden von Ansätzen zur Problemlösung
- ▶ Planung und Erläuterung gegenseitiger Zuarbeiten, Installationen etc.
- ▶ Identifizieren und Dokumentieren von neuen Aufgaben
- ▶ Pflege des Glossars

3.2.2.3 Backlogpflege

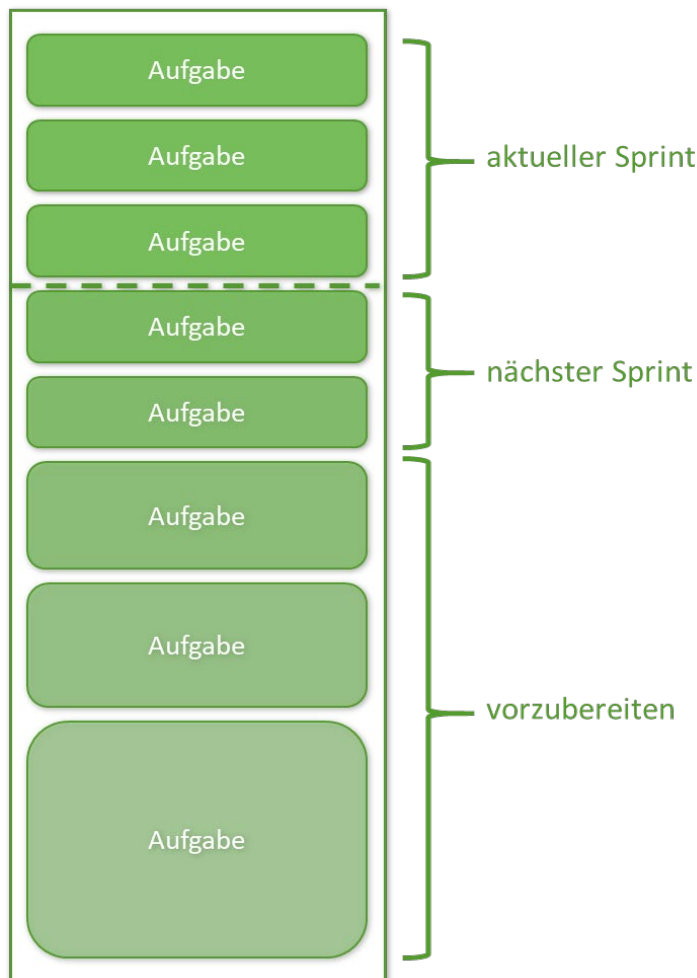
Die Pflege des Product-Backlogs ist ein kontinuierlicher Prozess, bei dem der Chief Product Owner die Aufgaben präzisiert, priorisiert und in kleinere Arbeitsblöcke aufteilt, die in einem Sprint bearbeitet werden können. Grundlage dafür sind die Einordnung einer Aufgabe in der

Roadmap, die Abhängigkeiten zu anderen Aufgaben und die Einschätzung des Realisierungsaufwandes.

Abbildung 2 zeigt den schematischen Aufbau des Product-Backlogs. Oben im Product-Backlog befinden sich die Aufgaben, die im aktuellen Sprint bearbeitet werden. Darunter die Aufgaben, die durch den Chief Product Owner unter Mithilfe der Product Owner der Teams vorbereitet wurden, sodass sie im nächsten Sprint umgesetzt werden können. Beim Eintrag in das Product-Backlog haben die Aufgaben einen sehr unterschiedlichen Detaillierungsgrad und auch eine unterschiedliche Komplexität. Wird zum Beispiel im Projektverlauf klar, dass das System um eine komplexe Funktion für die Druckausgabe von Karten ergänzt werden soll, so muss das beim Eintrag in das Backlog noch nicht genau spezifiziert sein. Auch eine Abschätzung des Entwicklungsaufwandes wird noch nicht unbedingt benötigt. In der Abbildung wird die unklare Spezifikation und die Unsicherheit bezüglich des Realisierungsaufwands durch unterschiedliche Größen der Aufgaben und durch eine blässere Farbe ausgedrückt.

Abbildung 2: Schematischer Aufbau des Product-Backlogs

Je komplexer eine Aufgabe ist, desto größer ist sie dargestellt. Je unklarer Spezifikation und Aufwandsschätzung, desto blässer die Farbe.



Quelle: Eigene Darstellung, con terra GmbH

Die Vorbereitung der einzelnen Aufgaben besteht nun darin, die Spezifikation zu schärfen, die Aufwandsschätzung einzuholen und ggf. eine Aufgabe in Teilaufgaben zu zerteilen, also sie in mehrere Backloginträge zu zerlegen. Es kann dabei durchaus notwendig sein allein für die Vorbereitung eigene Aufgaben zu definieren und an Teams zu vergeben. Ist eine Aufgabe hinreichend genau spezifiziert und ist der Aufwand für ihre Umsetzung hinreichend genau bekannt, so gilt sie als bereit für die Umsetzung (Sprint-ready) und kann in einen der nächsten Sprints aufgenommen werden. Die Arbeit mit dem Product-Backlog und sein Zusammenwirken mit den Sprint-Backlogs einzelner Teams wird in Kap. 3.3.5 genauer beschrieben.

Neben der aufgabenspezifischen Sicht auf das Product-Backlog ist bei dessen Pflege auch die übergreifende Sicht für die Priorisierung der Aufgaben und deren Zuteilung zu den verschiedenen Teams bedeutsam.

Die Erfahrung hat gezeigt, dass es sinnvoll ist, mit den Product Owners der an der Entwicklung beteiligten Teams regelmäßige Besprechungen durchzuführen, um den Chief Product Owner bei der Pflege des Product-Backlogs zu unterstützen. Dies ist im Fall der umwelt.info-Realisierung besonders wichtig, weil die Product Owner die Sichten ihrer Teams dem Chief Product Owner mitteilen müssen.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Product Owner
- ▶ Ggf. weitere Vertreter der Teams

Aufgaben

- ▶ Priorisieren der Aufgaben für die nächsten Sprints
- ▶ Detaillieren von bisher nur grob formulierten Aufgaben
- ▶ Schätzen des Aufwands für Aufgaben oder initiieren einer Aufwandsschätzung
- ▶ Aufteilen von komplexen Aufgaben in mehrere Arbeitspakete
- ▶ Aufnehmen von neuen Aufgaben aus den Planungsmeetings in das Product-Backlog
- ▶ Pflege des Glossars

3.2.3 Abschließende Betrachtung

Die abschließende Betrachtung jeder einzelnen Entwicklungsphase wird durch zwei Ereignisse gekennzeichnet, bei denen getrennt inhaltliche und methodische bzw. organisatorische Aspekte in den Mittelpunkt rücken. In Hinblick auf die Rahmenbedingungen bei der Entwicklung von umwelt.info ist hier sicherzustellen, dass Erkenntnisse, die in Reviews und Retrospektiven der einzelnen Teams erarbeitet werden, auf die Ebene des Gesamtteams transportiert werden, damit diese dort teamübergreifend berücksichtigt werden können.

3.2.3.1 Review im Team

In der engen Scrum-Definition dient das Review am Ende eines Sprints der Präsentation der Sprintergebnisse für die Auftraggeber*in oder andere am Projekt interessierte Personen [8]. Im

hier beschriebenen Ansatz mit den verschiedenen Ebenen des Teams und des Gesamtteams wird von diesem Ziel abgewichen. Das Review im Team ist ein teaminternes Ereignis, bei dem im Wesentlichen die vollständige Abarbeitung der Aufgaben und die Erreichung des Sprintziels geprüft werden. Der Product Owner des Teams als Teilnehmende*r am Review des Gesamtteams wird hier mit den Informationen versorgt, die er*sie dort benötigt.

Teilnehmer

- ▶ Product Owner
- ▶ Developer
- ▶ Ggf. Scrum Master

Aufgaben

- ▶ Vorstellung der durchgeführten Arbeiten
- ▶ Prüfen der Fertigstellung der einzelnen Aufgaben
- ▶ Prüfen der Erreichung des Sprintziels
- ▶ Zusammenstellen der Informationen für das Review des Gesamtteams

3.2.3.2 Review im Gesamtteam

Basierend auf den Ergebnissen der Reviews der einzelnen Teams wird das Review im Gesamtteam durchgeführt und vom Scrum Master moderiert. Im Review werden dem*der Projektleiter*in, dem Chief Product Owner und den anderen Teilnehmenden die Ergebnisse des Sprints präsentiert. Dazu können auch potenzielle Nutzende oder Fachexpert*innen eingeladen werden, deren Beurteilung für weitere Projektentscheidungen wichtig ist [18].

Beim Review des Gesamtteams muss berücksichtigt werden, dass einzelne Teams möglicherweise während des Gesamtsprints mehr als einen eigenen Sprint durchgeführt haben. Die Präsentation der Arbeiten erfolgt im Regelfall weniger detailliert als im Review des Teams.

Beim Review im Gesamtteam muss auch entschieden werden, was mit Aufgaben geschieht, die im Sprint nicht abgearbeitet werden konnten. Solche Aufgaben werden entweder im Product-Backlog für den nächsten Sprint vorgesehen, modifiziert, aufgeschoben oder sogar gestrichen.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Projektleiter*in AG
- ▶ Product Owner
- ▶ Scrum Master
- ▶ Ggf. weitere Vertreter der Teams
- ▶ Ausgewähltes Fachpersonal, Nutzende

Aufgaben

- ▶ Vorstellung der durchgeführten Arbeiten
- ▶ Prüfen der Fertigstellung der einzelnen Aufgaben
- ▶ Prüfen der Erreichung des Sprintziels
- ▶ Feststellen der Auswirkungen verfehlter Ziele auf den nächsten Sprint
- ▶ Abfragen der Meinung von Fachexperten und Nutzer*innen zu den erarbeiteten Lösungen

3.2.3.3 Retrospektive im Team

Im Gegensatz zum Review mit fachlich/inhaltlichem Schwerpunkt wird bei der Retrospektive Gewicht auf die Methodik und die Organisation während des letzten Sprints gelegt. Bei eingespielten Teams mit gut eingeführtem Vorgehensmodell sind Retrospektiven kurz. 90 Minuten sollten aber in der Regel nicht unterschritten werden [19].

Zu Beginn eines Projekts sind Retrospektiven eine gute Gelegenheit, die Methodik, die Rollen-zuweisung und die Kommunikationskanäle zu optimieren. Gerade bei den ersten Retrospektiven kann es sinnvoll sein, einen Scrum Master als Moderator hinzuzuziehen.

Ebenso wie beim Review macht es der komplexe Aufbau des umwelt.info-Gesamtteams notwendig, dass die Ergebnisse der Retrospektiven im Team durch ihre Vertretungen auf die Ebene des Gesamtteams transportiert werden. Innerhalb des Teams sind also nicht nur Fragen der Selbstorganisation zu klären, sondern auch Aspekte der Kommunikation mit anderen Teams oder dem Product Owner.

Teilnehmer

- ▶ Product Owner
- ▶ Scrum Master
- ▶ Developer
- ▶ Ggf. Projektleiter*in AN/UAN

Aufgaben

- ▶ Kritisches Betrachten der Zusammenarbeit
- ▶ Kritisches Betrachten der Kommunikation nach Außen
- ▶ Validieren des Vorgehensmodells
- ▶ Zusammenstellen der Informationen für die Retrospektive des Gesamtteams

3.2.3.4 Retrospektive im Gesamtteam

Viele Ergebnisse der Retrospektiven der einzelnen Teams sind nur innerhalb dieser Teams relevant. Andere aber haben Bedeutung für die Kooperation mit anderen Teams, dem Chief Product Owner oder dem Generalauftragnehmer. Diese Ergebnisse sollen durch die Product Owner der Teams in der Retrospektive des Gesamtteams eingebracht werden. Dort können

dann übergreifende Aspekte wie die Synchronisierung der Arbeiten in verschiedenen Teams oder die Teamgröße diskutiert und angepasst werden.

Teilnehmer

- ▶ Chief Product Owner
- ▶ Ggf. Projektleiter*in AG
- ▶ Product Owner
- ▶ Scrum Master
- ▶ Ggf. Projektleiter*in AN
- ▶ Ggf. Vertretungen der Teams

Aufgaben

- ▶ Kritisches Betrachten der Zusammenarbeit
- ▶ Kritisches Betrachten der Kommunikation zwischen den Teams
- ▶ Kritisches Betrachten der Kommunikation zwischen Chief Product Owner und Auftragnehmer
- ▶ Validieren des Vorgehensmodells

3.2.4 Sonstige Ereignisse

Neben den Ereignissen, die direkt aus der Scrum-Methode abgeleitet werden, gibt es weitere Veranstaltungen, die aus der Notwendigkeit entstehen für Spezifikationen die Meinung von Fachexpert*innen oder ausgewählten Nutzer*innen einzuholen.

Es handelt sich dabei um Workshops und Interviews, die unter Leitung von Vertretungen der entsprechenden Teams mit Personen durchgeführt werden, von denen man sich entweder detaillierte Informationen zu fachlich komplexen Fragestellungen oder Hinweise auf die optimale Benutzbarkeit des Systems bzw. auf die Ausgestaltung bestimmter Anwendungsfälle erwartet. Die Entscheidung über Zusammensetzung und Fragestellung obliegt dem Product Owner, Vorschläge dazu werden von den Teams erarbeitet.

Ein weiterer Typ von Ereignissen dient der Präsentation des Projekts selbst gegenüber einem Fachpublikum oder der Öffentlichkeit. Hier sind bestimmte Teams nur eingebunden, um Teile der Präsentation beizusteuern. Im Wesentlichen obliegt die Organisation und Durchführung aber dem Auftraggeber, insbesondere dem Chief Product Owner.

3.3 Artefakte

Bestimmte Dokumente oder Produkte, die im Verlauf des Entwicklungsprozesses eine definierte Rolle spielen, werden als Artefakte bezeichnet. Im Folgenden werden die verschiedenen Artefakte des umwelt.info-Vorgehensmodells beschrieben.

3.3.1 Produktvision

Die Produktvision ist eine sehr kurze, schlagwortartige Beschreibung des Produkts. Sie besteht im Idealfall nur aus ein oder zwei Sätzen, mit denen die Kernfunktionen und die wesentlichen Gruppen von Nutzenden des Systems aufgeführt werden. Sie klärt somit die Fragen, was das System grundsätzlich tut und für wen es gedacht ist.

Die Produktvision ist so allgemein gehalten, dass sie in der Regel im Verlauf des Projekts nicht geändert wird. Jede Änderung der Produktvision würde auch den Schwerpunkt des Projekts verschieben. Sie dient als Leitfaden und übergreifendes Motto. Bei jeder funktionalen Erweiterung sollte hinterfragt werden, ob die Produktvision durch sie unterstützt oder verfälscht wird [20].

Beim Erstellen der Produktvision müssen die Kernfunktionen des Systems zu benannt und festgelegt werden. Ebenso müssen die wesentlichen Nutzergruppen aufgeführt werden, an denen sich die Entwicklung zu orientieren hat.

Der Zwang, einen sehr kurzen Text zu finden, der dennoch die wichtigsten Aspekte des Systems enthält, lässt Produktvisionen oft sprachlich spröde und sperrig werden. Eine gute Produktvision zu formulieren ist schwierig und die Aufgabe darf nicht unterschätzt werden. Dennoch zeigt sich gerade bei umfangreichen Projekten mit langer Laufzeit, dass die wiederholte Rückbesinnung auf die Produktvision wichtig ist, um das Ziel des Projekts nicht unbewusst schleichend zu verschieben.

Die Verantwortung für die Erstellung der Produktvision liegt beim Chief Product Owner und der Projektleiter*in. Unterstützt werden sie dabei durch die Product Owner und ggf. durch weitere Experten.

3.3.2 Systemkontext

Im Zusammenhang mit der Produktvision wird durch dasselbe Gremium auch der Systemkontext erstellt. Er enthält Informationen dazu, mit welchen anderen Systemen das zu erstellende System interagiert und welche Schnittstellen dazu notwendig sind. Dadurch legt der Systemkontext aber die Abgrenzung des Systems fest (Systemgrenze). Gibt es Festlegungen, dass bestimmte Systeme bewusst nicht in den Systemkontext aufgenommen werden sollen, so können sie auch explizit ausgeschlossen werden, indem man sie außerhalb des Systemkontextes platziert [21].

Der Systemkontext muss nicht nur durch technische Systeme gebildet werden, er kann auch Regeln, Richtlinien oder Gesetze enthalten, die zu berücksichtigen oder ausdrücklich nicht zu berücksichtigen sind. Die Entscheidung darüber sollte der Chief Product Owner gemeinsam mit den Product Owners fällen und sich dabei bewusst machen, dass der wesentliche Sinn des Systemkontextes ist, die Entwickelnden zu unterstützen. Er ist nicht dazu gedacht, eine vollständige Dokumentation der geltenden Rahmenbedingungen zu liefern.

Im Lauf des Projekts kann der Systemkontext angepasst werden. Insbesondere kann es sein, dass externe Systeme nachträglich in den Systemkontext aufgenommen werden oder dass die Entscheidung für eines von mehreren vergleichbaren externen Systemen erst nachträglich gefällt werden muss.

Der Systemkontext für umwelt.info ist im Hauptdokument dargestellt.

3.3.3 Roadmap

Die Roadmap skizziert den groben Projektablauf und bestimmt die wechselnden Schwerpunkte der Sprintabfolge. Aus ihren einzelnen Elementen leiten sich die Sprintziele des Gesamtteams und in der Folge auch diejenigen der einzelnen Teams ab. Für den*die Projektleiter*in bietet die Roadmap eine gute Übersicht und die Möglichkeit, den Projektablauf darzustellen und zu kontrollieren. In ihrer Kompaktheit ist sie gut für die Präsentation des Projekts nach außen geeignet [20].

Die Roadmap wird im Wesentlichen durch den Chief Product Owner erstellt, die Product Owner verschiedener Teams können dabei unterstützen, der*die Projektleiter*in AG sollte hinzugezogen werden. Die Roadmap kann im Projektverlauf angepasst werden. Dies bedeutet aber einen schwerwiegenden Eingriff in den Projektablauf und sollte gut begründet sein.

Ein beispielhafter Entwurf für die Roadmap der ersten Phase der umwelt.info-Realisierung befindet sich in Kap. 5.4 des Hauptdokuments.

3.3.4 Glossar

Das Glossar dient dazu, den Gebrauch von Begriffen im Gesamtteam zu vereinheitlichen und so ein einheitliches Vokabular für den Projektkontext zu schaffen. Im Glossar sollten nur Begriffe aufgenommen werden, die im Projektkontext eine besondere Bedeutung haben. Allgemeine Begriffsdefinitionen sind überflüssig. Das Glossar legt somit fest, welche projektspezifischen Begriffe verwendet werden. Es wird zu Beginn des Projekts initial vom Chief Product Owner angelegt und im Verlauf des Projekts weiter gepflegt. Diese kontinuierliche Pflege erfolgt durch die Product Owner und den Chief Product Owner selbst. Mit welchem Maß an Selbstständigkeit das geschieht, muss bei der Initialisierung des Projekts festgelegt werden

Das Glossar muss am Ende des Projekts nicht in ganzer Tiefe fertiggestellt sein. Es handelt sich um ein Artefakt, das die laufende Arbeit erleichtern soll, nicht um ein abnahmerelevantes Projektergebnis. So können verschiedene Begriffsdefinitionen unvollständig sein oder ganz fehlen. Wichtige Begriffe sollten aber zumindest aufgeführt sein. Trifft man im Zuge der Projektarbeit auf neue Begriffe, die noch nicht im Glossar eingetragen sind, so sollte dies möglichst sofort erfolgen. Damit ist der Begriff in das Projektvokabular aufgenommen und die Grundlage für eine einheitliche Verwendung und Schreibweise des Begriffs gegeben [21].

3.3.5 Product-Backlog

Das Product-Backlog ist die zentrale Liste der zu erledigenden Aufgaben im Projekt. Aufgrund der verteilten, skalierbaren Struktur des Gesamtteams für die umwelt.info-Entwicklung, ist in diesem Kontext der Detaillierungsgrad des Product-Backlogs nicht sehr hoch. Das schließt nicht aus, dass Einträge des Product-Backlogs sehr detaillierte Informationen für die umsetzenden Teams beinhalten, dies sollte aber eher die Ausnahme als die Regel sein.

Die verstärkte Detaillierung der einzelnen Einträge und die Konkretisierung des hohen Abstraktionsniveaus im Product-Backlog erfolgt auf der Ebene der Sprint-Backlogs der einzelnen Teams (vgl. Abbildung 3). Einzelne Einträge des Product-Backlogs werden als Ziele für einen oder mehrere Sprints eines Teams aufgefasst. Aus dem Ziel entsteht dann bei der teaminternen Planung ein Sprint-Backlog mit konkreteren Aufgaben, die entsprechend genau beschrieben sind. Dies erfolgt bei der Sprintplanung im Team (vgl. Kap. 3.2.1.3).

Die Einträge im Product-Backlog umfassen nicht nur Aufgaben für die Realisierung von Funktionalität. Es kann sich auch um konzeptionelle oder organisatorische Aufgaben handeln.

Das Product-Backlog kann als lineare, senkrechte Liste von Einträgen aufgefasst werden. Dabei haben die obersten Einträge die höchste Priorität. Einträge am oberen Ende werden als nächstes realisiert. Dazu werden sie als Ziele an Teams übergeben. Die Voraussetzungen für die Übergabe eines Eintrags an ein Team ist eine ausreichende Konkretisierung, eine belastbare Abschätzung des Umsetzungsaufwands und ggf. eine Aufteilung des Eintrags in Teilaufgaben, die innerhalb eines Sprints gelöst werden können. Auch sollte der Eintrag bereits im Vorfeld mit dem Team besprochen worden sein, das mit der Realisierung betraut wird. Ist das nicht möglich, so muss zumindest der Product Owner dieses Teams mit dem Eintrag vertraut sein.

Die Schaffung der Voraussetzungen und ihre Überprüfung erfolgt bei der regelmäßigen Backlogpflege (vgl. Kap. 3.2.2.3) und der Sprintplanung im Gesamtteam (vgl. Kap. 3.2.1.2).

Idealerweise sind im Product-Backlog die Aufgaben zu erkennen, die aktuell bearbeitet werden, diejenigen, die in den nächsten Sprints an die Teams übergeben werden und die keine weitere Vorbereitung benötigen (Sprint ready) und alle anderen entsprechend ihrer Priorisierung. Diese Priorisierung bezieht sich dabei nicht unbedingt auf die Bedeutung einer Aufgabe für das Gesamtprojekt. Vielmehr kann die Priorisierung durch vorliegende bzw. fehlende Informationen, vorhandene Kapazitäten und andere Gegebenheiten beeinflusst werden. Insbesondere muss die Priorisierung der Einträge, die sich in ihrer Reihenfolge im Product-Backlog ausdrückt, permanent überprüft und ggf. angepasst werden. Eine solche Anpassung ist dabei nicht als Mangel anzusehen, sondern als schnelle Reaktion auf geänderte Gegebenheiten [4].

Neue Einträge werden ans Ende des Product-Backlogs eingefügt. Dies erfolgt so früh wie möglich. Das bedeutet, dass bei der Definition des Eintrags nicht alle Informationen vorliegen müssen, die für die Umsetzung notwendig sind. Dafür ist die Aufgabe von nun an im Gesamtteam bekannt und es existiert mit dem Eintrag ein Ort, an dem weitere Informationen gesammelt werden können. Im Rahmen der Backlogpflege wird der Eintrag mit weiteren Informationen versehen und ggf. in mehrere Einzeleinträge aufgeteilt, bis er entweder die Voraussetzungen für die Umsetzung erfüllt und einem Team zur Realisierung übergeben wird oder bis entschieden wird, dass er im Rahmen des Projekts nicht realisiert werden soll.

Die Verantwortung für die Pflege des Product-Backlogs liegt beim Chief Product Owner. Bei der Pflege wird er durch die Product Owner der einzelnen Teams und ggf. durch weitere Experten oder Developer unterstützt. Das Gesamtteam hat jederzeit Einblick in des Product-Backlog [8].

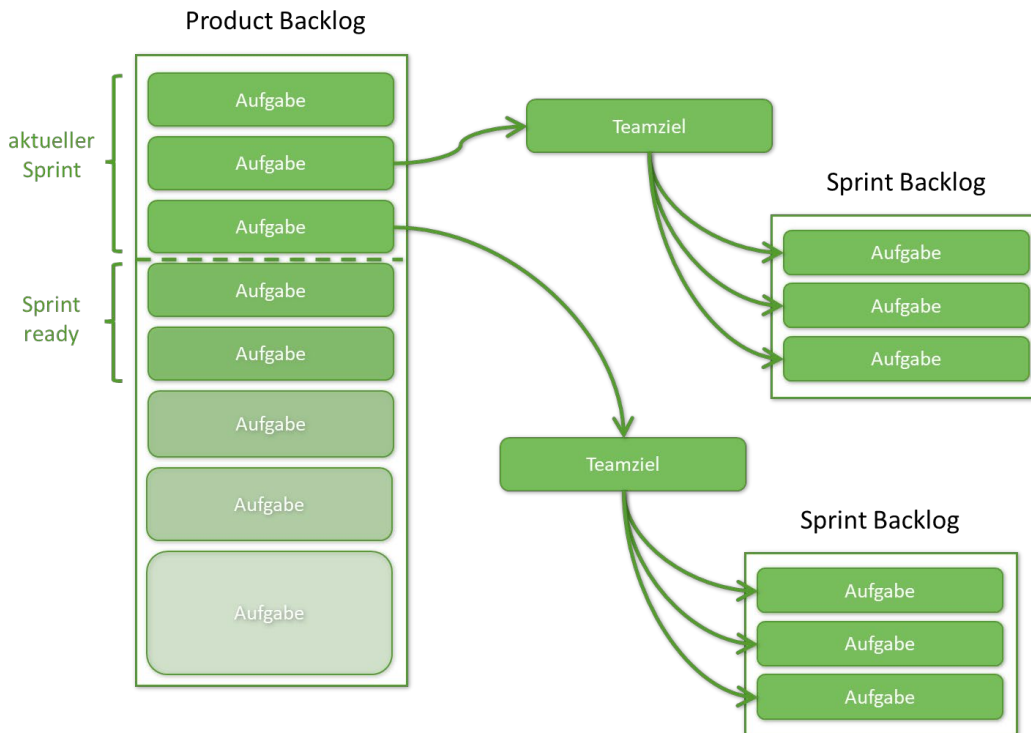
Das Product-Backlog spielt bei verschiedenen Ereignissen eine zentrale Rolle. Dies ist zunächst die Vorbereitung der Sprintplanung (vgl. Kap. 3.2.1.1). Hier wird anhand des Backlogs festgestellt, welche Voraussetzungen noch nicht erfüllt sind um bestimmte Einträge als umsetzungsreif (sprint ready) zu erklären. Außerdem wird mit Hilfe des Backlogs die Priorisierung der Einträge geprüft und ggf. überarbeitet und der Vorschlag für die Sprintplanung im Gesamtteam (vgl. Kap. 3.2.1.2) erstellt.

Im Rahmen dieses Planungsschrittes wird entschieden, welche Einträge des Product-Backlogs als nächstes bearbeitet werden. Diese werden entsprechend gekennzeichnet und den Teams zugewiesen. Damit ergeben sich auch die Sprintziele für die einzelnen Teams.

Bei der regelmäßigen Backlogpflege (vgl. Kap. 3.2.2.3) steht das Product-Backlog ohnehin im Mittelpunkt. Bei diesem Ereignis werden Informationen zu einzelnen Einträgen ergänzt und neue Einträge dem Backlog hinzugefügt.

Schließlich ist das Product-Backlog Grundlage für die Reviews im Gesamtteam (vgl. Kap. 3.2.3.2). Anhand der Einträge, die dem betrachteten Sprint zugewiesen sind, wird geprüft, inwieweit die Sprintziele erreicht wurden.

Abbildung 3: Zusammenhang von Product-Backlog und Sprint-Backlogs



Aufbau des Product-Backlogs und Übergabe von Teamzielen in Sprint-Backlogs
Quelle: Eigene Darstellung, con terra GmbH

3.3.6 Sprint-Backlog

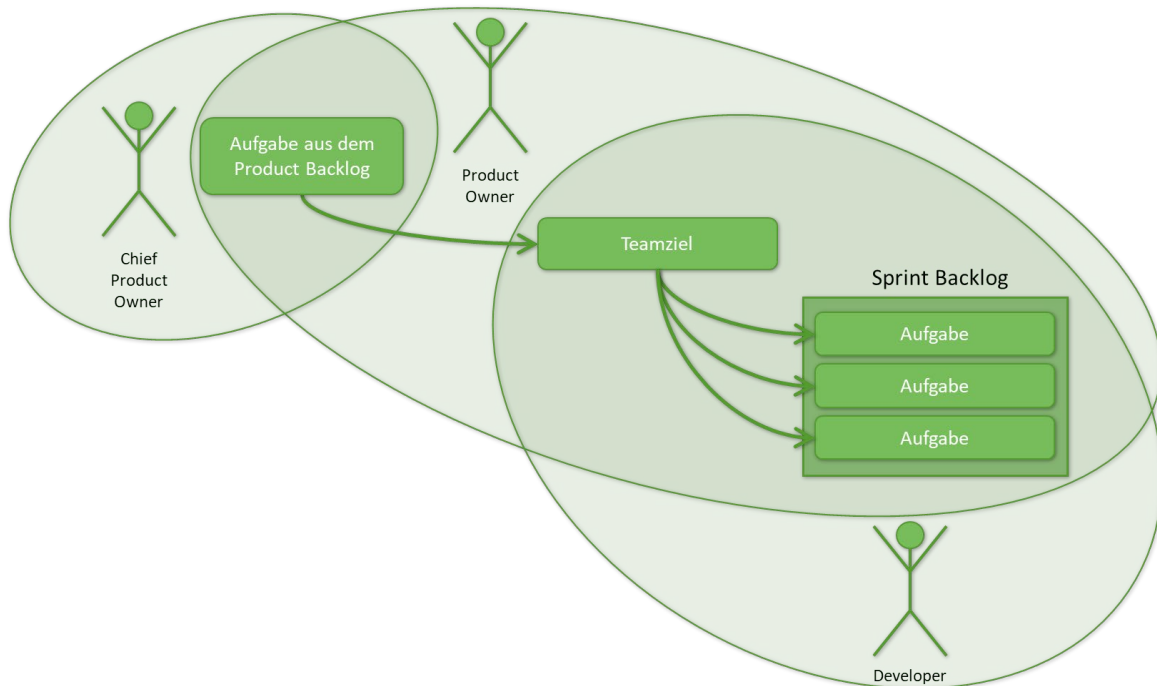
Während das Product-Backlog die zentrale Aufgabenliste für das Gesamtprojekt darstellt, ist das Sprint-Backlog die Aufgabenliste für ein bestimmtes Team und für einen bestimmten Sprint. Können künftige Aufgaben für dieses Team bereits detailliert beschrieben werden, so kann das Artefakt auch Einträge enthalten, die erst in zukünftigen Sprints umgesetzt werden sollen. In solchen Fällen ist aber darauf zu achten, dass keine Inkonsistenzen mit dem Product-Backlog entstehen. Dies sicherzustellen ist Aufgabe des Product Owners [8].

Grundsätzlich ist es aber so, dass das Sprint-Backlog einen inhaltlichen und zeitlichen Ausschnitt der Gesamtaufgabenmenge darstellt. Dafür sind die Einträge des Sprint-Backlogs detaillierter als diejenigen im Product-Backlog. Außerdem entstehen beim Übergang von Aufgaben vom Product-Backlog in das Sprint-Backlog in der Regel mehrere Teilaufgaben, damit die komplexe Gesamtaufgabe auf mehrere Developer aufgeteilt werden kann.

Abbildung 4 verdeutlicht das Füllen des Sprint-Backlogs mit Einträgen aus dem Product-Backlog. Der Chief Product Owner und der Product Owner geben einen Eintrag aus dem Product-Backlog als Sprintziel für ein Team vor. Daraus werden im Rahmen der Sprintplanung des Teams (vgl. Kap. 3.2.1.3) die Aufgaben für den anstehenden Sprint abgeleitet und im Sprint-Backlog eingetragen. Während der Chief Product Owner im Wesentlichen die Sicht auf die

Einträge des Product-Backlogs hat und die Developer im Team auf ihr Sprint-Backlog und die Sprintziele, ist es der Product Owner, der die Verbindung zwischen diesen Akteuren herstellt.

Abbildung 4: Ableiten von Einträgen des Sprint-Backlogs



Ableitung von Aufgaben im Sprint-Backlog aus dem Product-Backlog

Quelle: Eigene Darstellung, con terra GmbH

Beim Erzeugen von Aufgaben als Einträge im Sprint-Backlog müssen verschiedene Informationen für diese Einträge ermittelt oder festgelegt werden. Insbesondere muss für jeden einzelnen Eintrag ein erwarteter Realisierungsaufwand vermerkt werden. Übersteigt der Gesamtaufwand für alle Einträge den zuvor geschätzten Aufwand für das Sprintziel, so muss der Product Owner den Chief Product Owner hiervon in Kenntnis setzen und mit ihm gemeinsam eine Lösung finden. Solche Lösungen können aus Änderungen der funktionalen Spezifikation oder aus der Aufteilung des Sprintziels und der Bearbeitung über einen längeren Zeitraum in mehreren Sprints bestehen. Damit solche Konflikte möglichst vermieden werden, sollten Einträge aus dem Product-Backlog vor der Übergabe an ein Team bestimmte Kriterien erfüllen. So müssen sie bereits im Vorfeld mit dem Team besprochen worden sein und über eine belastbare Schätzung für den Gesamtaufwand verfügen.

Für jeden Eintrag müssen auch die Abnahmekriterien definiert werden ('Definition of Done'). Diese werden beim Review im Team (vgl. Kap. 3.2.3.1) benötigt, um zu entscheiden, ob ein Eintrag fertig bearbeitet, ist [22].

Die laufende Pflege des Sprint-Backlogs während des Sprints erfolgt durch den Product Owner. Dabei wird er durch die Developer unterstützt.

Steht das Product-Backlog auf der Ebene des Gesamtteams bei verschiedenen Ereignissen im Zentrum (vgl. Kap. 3.3.5) so ist das Sprint-Backlog eines Teams die Grundlage für verschiedene Ereignisse auf der Ebene des Einzelteams. Diese Ereignisse sind die Sprintplanung im Team (vgl. Kap. 3.2.1.3), bei der die Einträge im Sprint-Backlog auf der Basis des vorgegebenen Teamziels definiert werden, die tägliche Abstimmung im Team (vgl. Kap. 3.2.2.1) für die Kontrolle des

Entwicklungsfortschritts und die Planung der unmittelbaren nächsten Schritte sowie das Sprintreview im Team (vgl. Kap. 3.2.3.1) für die Feststellung des Grades der Fertigstellung der Einträge im Sprint-Backlog.

3.3.7 Inkrement

Es wird angestrebt, nach jedem Sprint eine lauffähige und präsentierbare Version des Produkts zu erstellen. Sie umfasst die Neuerungen, die durch die Umsetzung der Sprintziele realisiert wurden. Diese neue Version wird als Inkrement bezeichnet. Jedes Inkrement soll das Produkt näher an die Produktvision heranführen und gleichzeitig das Erreichen des Sprintziels demonstrieren. Dabei soll sich der Fortschritt der Inkremente an der Roadmap orientieren [8].

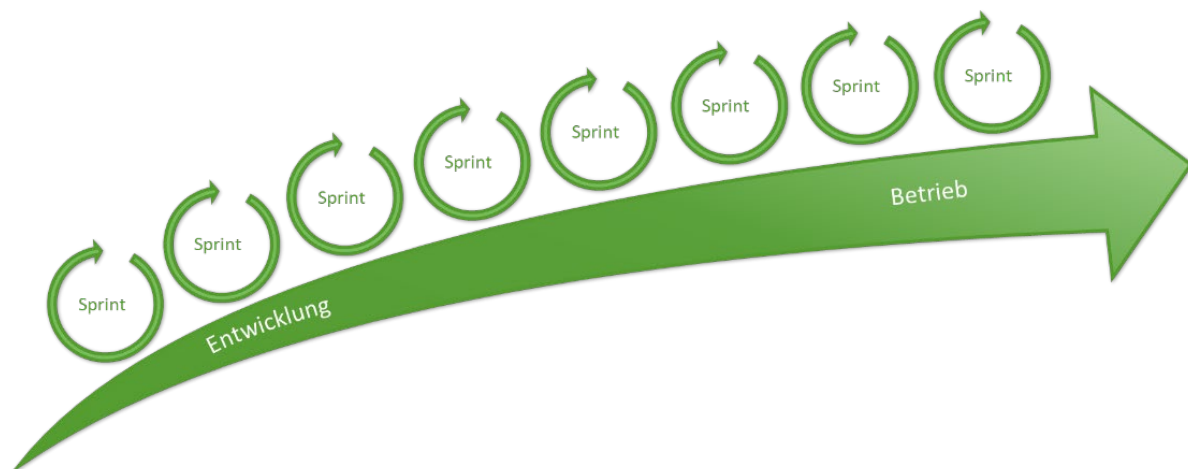
In Bezug auf umwelt.info muss zwischen Inkrementen auf Gesamtteamebene und auf Teamebene unterschieden werden. Die Inkremente der Gesamtteamebene setzen sich dabei aus den Einzelinkrementen auf Teamebene zusammen. Durch die strenge Kopplung an einen Sprint kann für jedes Inkrement nachvollzogen werden, welche Fortschritte es umfasst, diese entsprechen nämlich den Einträgen des Sprint-Backlogs. Um Inkremente aber noch besser unterscheidbar zu machen, sollten sie mit einer Versionsnummer, einer expliziten Teamzuordnung und ggf. einem Namen versehen werden.

Generell sollte versucht werden, möglichst früh eine grundsätzlich lauffähige Version zu erstellen, die dann schrittweise verbessert und funktional komplettiert wird. Dieses System sollte nach Möglichkeit den Kern aller am Produkt beteiligten Komponenten umfassen, um bereits die endgültige System-Infrastruktur widerzuspiegeln und um früh realitätsnahe Tests zu ermöglichen. Diese erste Version wird als Minimum Viable Product bezeichnet. Dieses kann in der Regel noch nicht nach dem ersten Sprint erstellt werden. Zunächst sind konzeptionelle Aufgaben zu lösen und die Projekt-Infrastruktur zu schaffen. Dennoch sollte der Chief Product Owner auf eine möglichst frühe Erstellung drängen.

3.4 Ablauf des Projekts

Entsprechend dem iterativen Ansatz des Scrum-Modells besteht der Ablauf des umwelt.info-Projekts aus einer Folge von Entwicklungszyklen, den Sprints (vgl. Abbildung 5).

Abbildung 5: Ablauf des Gesamtprojekts



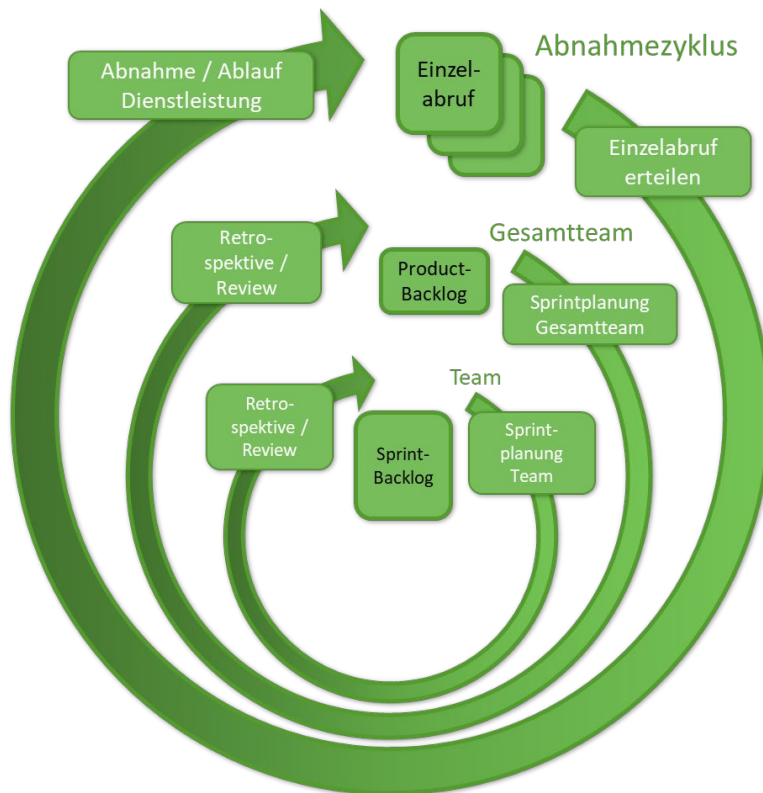
Projektablauf als Abfolge von Sprints mit fließendem Übergang von Entwicklung zu Betrieb

Quelle: Eigene Darstellung, con terra GmbH

Diese Sprints bringen das Produkt immer näher an den einsatzfähigen Zustand und verbessern es danach kontinuierlich weiter. Dabei geht das Produkt fließend von der Entwicklungs- in die Betriebsphase über. Dort endet es jedoch nicht, sondern wird laufend verbessert oder an sich ändernde Anforderungen angepasst.

Diese Betrachtung des Gesamtprojekts lässt allerdings die Besonderheit außer Acht, dass umwelt.info von mehreren verteilten Teams in verschiedenen gleichzeitig laufenden Zyklen auf unterschiedlichen Ebenen entwickelt wird. Abbildung 6 verdeutlicht das.

Abbildung 6: Verschiedene Zyklen bei der Entwicklung



Quelle: Eigene Darstellung, con terra GmbH

Das Entwicklungsvorhaben wird im Wesentlichen durch den Sprintzyklus des Gesamtteams geprägt. Er bildet den mittleren Zyklus der drei Zyklen im Vorgehensmodell. Anfang und Ende dieses Zyklus werden durch die Sprintplanung im Gesamtteam (vgl. Kap. 3.2.1.2) und die abschließende Betrachtung im Gesamtteam (vgl. Kap. 3.2.3.2 und 3.2.3.4) gebildet. Das maßgebliche Dokument ist das Product-Backlog (vgl. Kap. 3.3.5).

Die eigentliche Entwicklungsarbeit wird in Sprints der Einzelteams geleistet. Auch sie werden durch Ereignisse für Planung (vgl. Kap. 3.2.1.3) und Rückschau (vgl. Kap. 3.2.3.1 und 3.2.3.3) eingerahmt. Für diese Sprints ist das wesentliche Artefakt das Sprint-Backlog des jeweiligen Teams (vgl. Kap. 3.3.6).

Die Abbildung suggeriert, dass die Sprintlängen für das Gesamtteam und die Teams gleich sind. Das ist möglich aber nicht zwingend vorgegeben. Vielmehr werden die Sprints der Einzelteams und die des Gesamtteams in der Regel nicht die gleiche Dauer haben. Die Entwicklerteams arbeiten inhaltlich und räumlich eng zusammen, während das Gesamtteam durch die Gruppe der verschiedenen Product Owner, die aus unterschiedlichen Organisationen kommen, gesteuert

wird. Auch sind die Sprintziele im Gesamtteam weitreichender formuliert. Somit wird ein Sprint im Gesamtteam wahrscheinlich zwei bis vier Sprints eines einzelnen Entwicklungsteams umfassen. Die Product Owner sind aber dafür verantwortlich, dass die Synchronisierung zwischen den Zyklen funktioniert. So müssen z. B. die Ergebnisse der Retrospektiven auf der Teamebene in die Retrospektive auf der Ebene des Gesamtteams einfließen.

Auf der obersten Ebene befindet sich der Abnahmezyklus, der den organisatorischen Rahmen bildet. Hier werden die vertraglichen Grundlagen für das Entwicklungsprojekt geschaffen, indem Einzelabrufe über Werke oder Dienstleistungen formuliert werden. Ein Abrufzyklus erstreckt sich selten über einen, im Regelfall über mehrere Sprints des Gesamtteams. Während dieser Sprints wird die vereinbarte Leistung erbracht. Am Ende eines Abnahmezyklus steht die Abnahme des Werkes bzw. der Ablauf der Dienstleistung. Einzelheiten der vertraglichen Regelungen werden in Kap. 3.8 beschrieben. Mit Hilfe dieser Einzelabrufe kann die Projektleitung Meilensteine für die Projektentwicklung definieren, ohne zu sehr in die direkte Steuerung der Entwicklung einzugreifen. Diese erfolgt im Zyklus des Gesamtteams durch den Chief Product Owner.

3.5 Methoden

Im Folgenden werden einige zentrale Tätigkeitsfelder benannt, für die Methoden und (technische) Werkzeuge bewusst ausgewählt werden müssen. Exemplarisch wird in diesem Abschnitt eine Auswahl an Methoden und Werkzeugen vorgestellt.

Bei der späteren Umsetzung sollte darauf geachtet werden, welche Tätigkeitsfelder über mehrere Organisationseinheiten oder Teams hinweg greifen und ob daher einheitliche Methoden und Werkzeuge gewählt werden müssen. Andernfalls können Entscheidungen darüber auch innerhalb einzelner Teams getroffen werden.

3.5.1 Anforderungs-Management

Nicht alle Elemente eines Product- oder Sprint-Backlogs müssen dieselbe Qualität oder Gestalt haben. Manche Aufgaben sind eher technischer oder organisatorischer Natur wie zum Beispiel Maßnahmen aus einer Retrospektive. Andere Backlog-Items beschreiben jedoch, basierend auf den Gesamtzielen des Projektes, die Anforderungen der Nutzer*innen.

Die Perspektive der Nutzer*innen möglichst konkret und unverfälscht einzubringen, dient dazu, die Hypothesen über die für die Nutzenden zu lösenden Probleme und den bisher entwickelten Lösungsweg (die fertigen Inkremente) zu verifizieren und somit stets nah an dem zu bleiben, was für den Nutzenden des Systems von größtem Wert ist.

Ein zentrales Anliegen für Anforderungsmanagement im agilen Kontext ist es, zwischen allen Beteiligten ein gemeinsames Verständnis über die zu lösenden Probleme herzustellen und kollaborativ an deren Lösung zu arbeiten. Dabei das für die Situation passende Gleichgewicht zwischen Planung bzw. Definition eines Vorgehens und der Konversation über mögliche Lösungswege und das Einbeziehen neuer Erkenntnisse zu finden, ist eine der wesentlichen Herausforderungen.

Um die oben genannten Aspekte zu adressieren, schlagen wir User Stories als Methode zur Beschreibung von Anforderungen vor [23]. User Stories beschreiben im Wesentlichen, für welche Art von Nutzenden welches Problem gelöst werden soll und vor allem auch, warum. User

Stories folgen häufig diesem Aufbau: „Als ein <Typ eines Nutzenden> möchte ich ein <Ziel erreichen / Problem lösen>, damit <Motivation / Grund> erfüllt ist.“ [23].

Eine solche Story ist bewusst nicht bis ins letzte Detail vollständig beschrieben, damit Diskussion und Weiterentwicklung von Anforderungen zum richtigen Zeitpunkt ermöglicht werden – nämlich kurz vor der Umsetzung, wenn möglichst alle nötigen Erkenntnisse aus bisherigen Inkrementen mit einfließen können. Die Konversation im Team über die in der Story beschriebenen Aspekte steht also im Mittelpunkt [23].

Im Lebenszyklus einer User Story werden üblicherweise gemachte Erkenntnisse oder getroffene Entscheidungen festgehalten oder es wird auf weitere Informationen verwiesen. So werden Stories beispielsweise oft durch weitere funktionale Anforderungen (häufig durch Akzeptanzkriterien oder Akzeptanztests) ergänzt oder enthalten Verweise auf Nutzerstudien, Prototypen, Mockups, Architektur- oder sonstige relevante Dokumente.

Der Umgang und das Formulieren von User Stories müssen im Gesamtteam eingeübt werden und können durch Team Retrospektiven auch Gegenstand von laufender Verbesserung sein. Prinzipien und Leitsätze für den *Umgang* (Card, Conversation, Confirmation [13]), das *Schreiben* (“Independent, Negotiable, Valuable, Estimable, Small, Testable“ [24] & Specific, Measurable, Achievable, Relevant, Time-boxed [24]) oder das *Verfeinern* (Spikes, Paths, Interfaces, Data, Rules [25]) können hier Anleitung geben.

Im beschriebenen Prozess können User Stories an allen Stellen eine Bedeutung haben, der Fokus liegt aber auf dem Sprint und dem Team. Alle Mitglieder eines Teams können bei der Erfassung oder Bearbeitung unterstützen, allerdings bleibt es ausschließlich dem (Chief) Product Owner vorbehalten, Stories ins Backlog aufzunehmen bzw. deren Reihenfolge, basierend auf dem Wert für die Nutzenden, festzulegen. User Stories können auch außerhalb der Ereignisse Backlogpflege und Sprintplanung entstehen und sich auch während eines Sprints verändern, sofern dies mit dem entsprechenden Product Owner abgestimmt ist.

3.5.2 Planung

Zu den Aufgaben der Product Owner gehört die Priorisierung des Product-Backlogs und damit auch die Planung von Releases (mit frühen und häufigen Veröffentlichungen an die Nutzenden können Annahmen über das zu erzeugende Produkt früh validiert werden). Wesentliche Kriterien, die dabei hinzugezogen werden können, sind der Aufwand für die Umsetzung und der Wert für den Systembetreiber bzw. die Nutzenden. Auf diese Weise können wertvolle jedoch weniger aufwändige Anforderungen früh umgesetzt werden.

Weitere Ziele der (Chief) Product Owner sind das Synchronisieren von Zielen und Aufgaben für mehrere Teams sowie das Planen von Teilabnahmen bzw. dafür passenden Zeitpunkten und das Berücksichtigen von (technischen oder organisatorischen) Abhängigkeiten wie sie beispielsweise in der Roadmap sichtbar werden. Konkret müssen also Fragen auf verschiedenen Ebenen beantwortet werden: Von der Frage nach den Inhalten für den nächsten Sprint bis zur Frage nach Zeitpunkten eines anstehenden Releases oder was der Inhalt davon sein wird.

Diese Ziele lassen sich mit den hier vorgeschlagenen Methoden erreichen, jedoch bleiben bestimmte Herausforderungen dabei grundsätzlich bestehen. So liegt eine Unschärfe in der Vorhersagbarkeit von Zeit oder Umfang von Meilensteinen in der Natur der Sache komplexer Softwareentwicklung. Zu dieser Komplexität gehören auch Abhängigkeiten, die – sofern bekannt – über die Roadmap abgebildet werden können. Hierin steckt zudem die Herausforderung, im Verlauf nicht der hierarchischen Struktur von Softwarekomponenten einer Architektur zu

folgen, sondern einen querschnittlichen Durchstich anzustreben, der früh ein nutzbares Inkrement erzeugt. Ein Team für sich und erst recht die Kooperation mehrerer Teams bedeutet ein hohes Maß an Komplexität. Der „wirkliche“ Businesswert oder Nutzen von Funktionen bei den Nutzenden ist schwer zu messen. Die agilen Metriken, die ein Team über die Dauer der Zusammenarbeit erzeugt, können leicht als Managementinstrument missbraucht werden. Dabei sollten sie nur als Datengrundlage für die Verbesserung von Inkrementen und die Reflexion des eigenen Vorgehens genutzt werden.

Um beispielsweise die o.g. Unschärfe und Komplexität in bzw. zwischen Teams quantifizierbar zu machen, bietet sich eine Aufwandsschätzung mit abstraktem Schätzmaß – oft Story Points [26]– an. Diese können z. B. mit der Methode *Planning Poker* [27] bei der Backlogpflege oder Sprintplanung genutzt werden. In diese Schätzung gehen die *Menge zu erledigender Arbeit*, *Risiko und Unsicherheiten* sowie die *Komplexität* ein. Es gibt neben *Planning Poker* weitere Schätzverfahren, die je nach Situation weniger Austausch zwischen den Beteiligten bieten, aber schneller sind bzw. mit größeren Mengen an Anforderungen umgehen können wie z. B. *Bucket* [28] oder *Magic Estimation* [29].

Ähnliche Methoden kann man für die Bestimmung von Businesswert verwenden, so dass die (Chief) Product Owner in die Lage versetzt werden, eine wertbasierte Priorisierung (auch unter Berücksichtigung der zuvor erwähnten Aufwandsschätzung) zu pflegen. Üblich sind hier Verfahren, die gemeinsam mit den wesentlichen Stakeholdern (gemeinhin Vertreter von Nutzendengruppen) durchgeführt werden. Dazu zählen *Planning Poker* [27] oder auch *Buy a Feature* [30]. Um die Bedeutung bestimmter Fachlichkeit für die Nutzenden zu ermitteln, können auch Methoden aus dem *User Centered Design* (UCD) [31] genutzt werden. Darüber hinaus empfiehlt es sich die Hypothesen über den *Return on Investment* [32] und den Mehrwert für die tatsächlichen Nutzenden z. B. durch weitere Evaluationsmaßnahmen zu verifizieren.

Mit Hilfe der User Story Mapping [33] Methode kann man erreichen, das Backlog von einer eindimensionalen Reihenfolge über Priorität in eine mehrdimensionale Sicht zu erweitern, die es ermöglicht, Inkremente als Durchstich entlang von Nutzerszenarien zu planen.

Mit einer zielorientierten Produkt Roadmap [34] können Product Owner Releases für die entsprechenden (Sub-)Systeme planen. Eine transparente und gemeinsam erarbeitete Roadmap ist nicht nur ein Kommunikationsmittel für die Projektziele, sondern dient auch dazu, Abhängigkeiten und Unsicherheiten für alle Organisationseinheiten sichtbar zu machen. Eine solche Roadmap wird regelmäßig aktualisiert und als Entscheidungsgrundlage für öffentlichkeitsrelevante Releases genutzt. Als weitere Darstellungsform für den Fortschritt eines Releases kann man *Release Burndown-Charts* [35] verwenden. Bei diesen wird der verbleibende Entwicklungsaufwand im Sprint gegen die verbleibende Zeit aufgetragen, was Rückschlüsse auf die Qualität der Aufwandsschätzung zulässt.

Diese und weitere Planungsmethoden finden bei der Sprintplanung, Backlogpflege oder in Abstimmungstreffen der Product Owner und Stakeholder Anwendung. Als Artefakte werden im Wesentlichen die Roadmap und das Product-Backlog manipuliert. Im Sinne eines gemeinsamen Verständnisses im Gesamtteam ist es aber ratsam, die Artefakte transparent im Team zu machen und Teammitglieder wo sinnvoll mit einzubinden.

3.5.3 Coaching, Feedbackschleifen und Meetings

Ziel der agilen Vorgehensweise ist die evidenzbasierte und datengestützte Verifizierung von Hypothesen im Problem- und Lösungsraum [36]. Dieser empirische Ansatz setzt auf

Transparenz, Analyse und Anpassung des eigenen Vorgehens, der getroffenen Annahmen und der Artefakte, die ausgeliefert werden [36]. Die hier erwähnten Methoden und Techniken sollen also dabei helfen, diese drei Fragen zu beantworten:

1. Bearbeiten wir das richtige Problem bzw. haben es richtig verstanden?
2. Liefern wir dafür die richtige Lösung und hilft diese auch, das Problem zu lösen?
3. Haben wir dafür ein motiviertes Team und den passenden Prozess?

Die Herausforderungen in diesem Kontext bestehen darin, den Kontakt zu den "echten" Nutzenden herzustellen, oder zumindest die Barrieren zwischen Entwicklung und Nutzenden zu minimieren. Außerdem sollte man auch diejenigen Daten und Metriken erheben, in deren Kontext man optimieren möchte, sowie Veränderungen an bereits Erstelltem ermöglichen. Um Prozessverbesserungen zu erkennen und das Teamwork zu fördern, ist oftmals eine Strategie notwendig, Regeln im Team explizit zu machen und ganz grundsätzlich ein Team durch Coaching, Training oder ähnliche Angebote zu begleiten (oft durch den Scrum Master durchgeführt).

Dafür sind Retrospektiven unverzichtbar [19], die ein Team als geschützten Raum nutzen kann, um offen über Verbesserungen zu reflektieren und diese z. B. für eine kommende Iteration einzuplanen. Maßnahmen und Experimente, die ein Team auf diese Weise plant, sollten stets in einer folgenden Retrospektive verifiziert und schließlich etabliert, angepasst oder verworfen werden. Innerhalb eines Teams sollten die Aktionen und Experimente einer Retrospektive sichtbar und zugänglich sein - beispielsweise als Teil des Sprint-Backlogs [37].

Das Sprint-Review ist die bereits erwähnte Methode, um Hypothesen im Lösungsraum zu verifizieren. Hier muss das Team zum Artefakt und Kontext passende Methoden finden, um zu ermitteln, wie gut das aktuelle Inkrement als Lösung für die Nutzer*innen funktioniert. Manchmal geschieht dies stellvertretend durch den Product Owner, aber direkte Methoden wie eine Präsentation für Nutzende oder andere Formen der Evaluation sind vorstellbar.

Auf Ebene des Problemraums finden häufig Methoden des UCD, Nutzertests, Workshops oder andere Product-Discovery-Methoden [38] statt. Auf diese Weise kann Unsicherheit über Erkenntnisse der für die Nutzenden zu lösenden Probleme minimiert werden [39].

Coaching durch Scrum Master oder andere Teammitglieder können zu fokussierten Meetings unterstützend beitragen – z. B. durch ein klares Ziel, klar formulierte Ergebnisse, eine Timebox und eine klare Teilnahmeregelung. Außerdem sollten während und nach eines Meetings Arbeitsergebnisse visualisiert werden.

Feedbackschleifen und Reflexion kann auf allen Ebenen des Gesamtteams stattfinden. Im Mittelpunkt stehen Retrospektiven und die Nachverfolgung von Maßnahmen daraus (z. B. im Sprint-Backlog).

3.5.4 DevOps & Software Craftsmanship

DevOps [40] und *Software Craftsmanship* [41] sind Paradigmen, die das Selbstverständnis professioneller Softwareentwickler*innen in einem motivierten Team und die enge Kooperation untereinander (Entwicklung und Betrieb) und mit dem Kunden in den Mittelpunkt stellen, um dessen Ziele zu erreichen und dabei professionell und mit höchster Qualität im Blick vorzugehen [5].

DevOps umfasst dabei verschiedene Methoden, mit denen ein enges Zusammenwirken von Softwareentwicklung, Betrieb und Nutzung erreicht werden kann. Software Craftmanship soll

durch die Stärkung der Eigenverantwortung von Entwickelnden und durch Betonung von handwerklichen Aspekten der Softwareentwicklung nachhaltig die Qualität von Software verbessern.

Um die Ideen von DevOps und Software Craftmanship zu verwirklichen, werden Barrieren zwischen Betriebs- und Entwicklungsteams abgebaut, indem durch Metriken und Feedbackschleifen gestützt ein lernendes, cross-funktionales und integriertes Team entsteht. Dieses wirkt u.a. durch Veränderbarkeit von Code und Architektur darauf hin, dass durch schnelles Deployment auch schnelles Feedback der Nutzer*innen oder Stakeholder in die Entwicklung zurückfließen kann [42].

Eine besondere Herausforderung dabei ist es, trotz teamdynamischer Prozesse und mobiler, verteilter Arbeit über die Grenzen von Organisationseinheiten hinweg ein gemeinsames Verständnis von Zielen und Umsetzung des Systems zu erreichen. Dazu ist Zeit für den Austausch, das Lernen und die Wissensverteilung nötig, so dass Teams sich auf Qualität, Einfachheit und Veränderbarkeit von Code konzentrieren können.

Auf der technischen bzw. Code-Ebene tragen Prinzipien, wie Automatisierung durch *Continuous Integration / Continuous Delivery Pipelines* (CI/CD) [43], *Test Driven Development* [44] (oder auch *Domain Driven Design* [45]) oder *Clean Code* [46] bei. Auf der Team-Ebene können Methoden wie *Pair Programming* oder *Coaching* (z. B. durch Feedback) zum Erreichen der oben genannten Ziele beitragen.

Praktiken und Paradigmen aus diesem Methodenkomplex [11] sollten im Gesamtteam verstanden sein, betreffen dennoch eher die Arbeit der Entwicklungs- und Betriebsteams (oder eines DevOps-Teams) und finden Anwendung bei der Umsetzung von Backlog-Items innerhalb eines Sprints. Retrospektiven können ebenso für Verbesserung der Team Zusammenarbeit wie zur Definition von Schritten hin zu mehr Automatisierung, Code Qualität und Wissensaustausch genutzt werden.

3.6 Werkzeuge

Die Wahl der einzusetzenden Werkzeuge orientiert sich an den bei der Umsetzung abzudeckenden Funktionsbereiche, wie Kommunikation, Dokumentenablage, Product-Backlog Management, Codeverwaltung und CI/CD.

Die Auswahl technischer Werkzeuge wird von organisatorischen Rahmenbedingungen beeinflusst (Lizenzen, Verbot oder Bevorzugung bestimmter Produkte oder Produkttypen, ...), die eine verbindliche Festlegung erst unmittelbar vor Projektstart sinnvoll machen. Vor diesem Hintergrund sind die hier aufgeführten Werkzeuge als beispielhafte Empfehlungen zu verstehen.

Die folgenden Abschnitte sind nach Funktionsbereichen gegliedert, für die die Teams Werkzeuge für die Umsetzungsphase finden und abstimmen müssen. Bei der späteren Auswahl von konkreten Werkzeugen muss insbesondere die Architekturrichtlinie des Bundes [47] im dann gültigen Stand berücksichtigt werden. Weiter sind Rahmenbedingungen des Bundesamts für Sicherheit in der Informationstechnik, des BMUV oder des Umweltbundesamts (UBA) oder die Datenschutz-Grundverordnung (DSGVO) bei der Werkzeugauswahl zu berücksichtigen.

3.6.1 Kommunikation

Werkzeuge für die Kommunikation sind an allen Stellen des Prozesses und für alle Beteiligten relevant. Die Interoperabilität zwischen Organisationseinheiten muss gewährleistet sein, damit

alle im Gesamtteam jederzeit ohne Hindernisse synchron oder asynchron und zielgerichtet kommunizieren können – insbesondere in Zeiten von Mobilem Arbeiten und Homeoffice.

Zu berücksichtigen ist, dass auf AG-Seite und ggf. bei zukünftigen Nutzenden nur browser-basierte Anwendungen zur Verfügung stehen könnten (z. B. *Microsoft Teams* [48] oder *WebEx* [49]). Werkzeuge, mit denen sich unkompliziert Feedback einholen lässt (z. B. *slido* [50] und *mentimeter* [51]), sind ebenfalls in Erwägung zu ziehen.

3.6.2 Backlog Management

Das Product-Backlog muss für das Gesamtteam zugreifbar und transparent sein. Denn obwohl im Wesentlichen die (Chief) Product Owner das Management übernehmen, muss das Product-Backlog auch von jeder Person im Gesamtteam editierbar sein. Daher bietet sich hier ein für verteilte Teams und asynchronen Zugriff geeignetes Werkzeug an.

Der Funktionsumfang hierbei ist nachrangig einzuordnen, da es im Wesentlichen darum geht, im Gesamtteam ein gemeinsames Verständnis von Zielen und Aufgaben herzustellen und die Elemente des Backlog in eine eindeutige Reihenfolge zu bringen.

Diese grundlegenden Aspekte von Transparenz und Zugreifbarkeit sollten auch auf die Sprint-Backlogs übertragen werden. Gängige Werkzeuge vereinen oft beide Sichten auf das Backlog, wie z. B. *GitLab Issues* [52], *Microsoft Planner* [53] und *Atlassian JIRA* [54].

3.6.3 DevOps & Software Craftmanship

Für verschiedene Bereiche von der Entwicklungsumgebung über die Codeverwaltung, das Artefakt- und Dependency-Management, das Security-/Threat-Management bis hin zur CI/CD Pipeline sind eine Vielzahl von Entscheidungen bezüglich der im Projekt eingesetzten Werkzeuge zu treffen. Einige dieser Entscheidungen können von den Umsetzungsteams zu einem späteren Zeitpunkt getroffen werden, so dass die gewählten Werkzeuge möglichst nah an den Bedürfnissen der Teams und denen der Architektur sind.

Wie Betrieb und Entwicklung durch Werkzeuge zusätzlich so verzahnt werden können, dass durch Daten, Metriken und Feedbackschleifen für echte Zusammenarbeit, Produktqualität und schnelle Auslieferung gesorgt werden kann, sollte jedoch teamübergreifend überlegt und entschieden werden.

Ein zusätzlicher Bereich betrifft das visuelle Software-Engineering sowie das Erstellen von Artefakten für Prototypen, Tests oder UI-Entwürfe.

Bei allen Themenbereichen ist abzuwägen, wo Entscheidungen für verwendete Werkzeuge nur einzelne Teams betreffen (z. B. bei der verwendeten DIE) und wo Einheitlichkeit zwischen verschiedenen Teams und Organisationseinheiten hergestellt werden muss (Codeverwaltung). Beispiele dafür sind *GitLab* [52] oder *Jenkins* [55].

In Bezug auf Software Craftmanship muss von Seiten der Projektleitung entschieden werden, ob einheitliche Regeln wie z. B. ein für alle Teams verbindlicher Styleguide eingeführt werden sollen.

3.6.4 Visualisierung, Modellierung und Dokumentation

Eine große Bedeutung bei der Kommunikation komplexer Sachverhalte und dem Festhalten von Ergebnissen aus Meetings kommt der Visualisierung bzw. Modellierung zu [56].

Insbesondere für verteilte, Remote-Work-Teams ist eine visuelle Repräsentation von Meeting- oder Diskussionsständen essenziell. Je nach Darstellungsschwerpunkt (Prozesse, Diskussionen, Softwarekomponenten, etc.) können unterschiedliche Werkzeuge von Nutzen sein.

Dabei ist grundsätzlich abzuwägen, ob man ein eher spezialisiertes Werkzeug verwenden möchte (z. B. was ausschließlich für die Durchführung von Retrospektiven geeignet ist, wie Retrium [57] oder eher generalistische Werkzeuge bevorzugt (wie z. B. Online-Whiteboards, wie Miro [58] oder Collaboard [59]), die zwar oft komplizierter in ihrer Anwendung sind aber mehr verschiedene Anwendungsgebiete unterstützen.

Da insbesondere in diesem Umfeld viele Anbieter mit proprietären Formaten arbeiten, ist besonders darauf zu achten, dass zumindest ein offenes Format für den Austausch als Backup oder Export zur Verfügung steht. Auch beim Erzeugen und Austausch von Dokumenten (z. B. erstellt mit Microsoft Office [60] oder Libre Office [61]) ist auf offene Austauschformate zu achten. Für Ablage und Austausch von Dokumenten muss insbesondere auf DSGVO Konformität geachtet werden, weswegen sich u. A. eine Ablage von Dokumenten auf Seite des AG anbietet. Entsprechende Werkzeuge und Plattformen sind hier bereits im Einsatz (z. B. BSCW [62]).

3.7 Eskalationsmechanismen

Gerade in umfangreichen und komplexen Projekten kommt es häufig zu Konflikten, die einer Eskalation und der Klärung durch übergeordnete Stellen bedürfen. Um solche Konflikte möglichst schnell und effizient zu lösen, sollten Eskalationsmechanismen definiert sein, die in typischen Konfliktsituationen greifen.

Im Folgenden werden Lösungsansätze für Problemsituationen aufgeführt, die in größeren Softwareentwicklungsprojekten typischerweise vorkommen:

- ▶ **Uneinheitliches Verständnis eines Teamziels:**
Haben Entwickelnder und Product Owner ein unterschiedliches Verständnis eines Teamziels, so sollte der Chief Product Owner als klärende Stelle herangezogen werden. Der Chief Product Owner ist in inhaltlichen Fragen des Projekts die maßgebliche Entscheidungsstelle und Auslöser für die Definition des Teamziels.
- ▶ **Uneinigkeit über Sprintziel des Gesamtteams:**
Kann im Gremium der Product Owner mit dem Chief Product Owner keine Einigkeit über ein Sprintziel für das Gesamtteam erreicht werden, so sollte die Projektaufsicht konsultiert werden. Im Fall der Änderung eines bereits formulierten Sprintziels oder seiner Neuformulierung, sind die Auswirkungen auf Aufwand und Kapazitätsbedarf zu untersuchen. Dafür müssen die Projektleitenden auf Seiten des Auftraggebers und der betroffenen Auftragnehmer hinzugezogen werden.
- ▶ **Teamziel nicht erreicht, Aufwandsschätzung falsch:**
Wird ein Teamziel nicht erreicht oder stellt sich eine Aufwandsschätzung als falsch heraus, so ist das zunächst zwischen dem Product Owner des betreffenden Teams und dem Chief Product Owner zu diskutieren. Sind Auswirkungen auf die Arbeit anderer Teams oder auf das Budget zu erwarten, sind die Projektleitenden auf AG- und AN-Seite hinzuzuziehen.

Generell gilt, dass die Projektleitung zu konsultieren ist, wenn Konflikte Auswirkungen auf andere Teams oder das Budget haben. Bei inhaltlichen Fragestellungen besteht folgende

Eskalationskette: Entwickelnder -> Product Owner -> Chief Product Owner -> Lenkungsausschuss.

3.8 Vertragsgestaltung

In diesem Kapitel wird ein Vorschlag für die Vertragsgestaltung inklusive Preisgestaltung erarbeitet, der die oben beschriebene agile Vorgehensweise berücksichtigt. Die agile Vorgehensweise ist in Hinblick auf die Vertragsgestaltung insbesondere dadurch gekennzeichnet, dass noch keine vollumfängliche Beschreibung des Endergebnisses vorliegt, sondern diese erst im Verlaufe des Projektes konkretisiert wird.

Einen recht guten Überblick der Problematik findet sich in [63]. Danach sind in der Vertragsgestaltung u. a. zu beachten, dass die Vision zu Beginn des Projektes klar beschrieben ist (vgl. Kap. 3.3.1). Außerdem soll festgelegt sein, wer die Hauptverantwortung für den Entwicklungsprozess trägt (vgl. Kap. 3.1, insbesondere Ausführungen zu Chief Product Owner und Product Owner). Darüber hinaus sind Dauer und Turnus der Sprints festzulegen (vgl. Kap. 3.2.1). Von besonderer Bedeutung ist es auch, im Vertrag die Aufgaben und Rollen des Auftraggebers klar zu definieren, um spätere Differenzen hinsichtlich der Verantwortlichkeiten und der Eskalationsmechanismen zu vermeiden.

Insbesondere im Hinblick auf die Systempflege kommt der Dokumentation eine bedeutende Rolle zu. Daher sollten bereits im Vertrag Detaillierungsgrad und Verantwortung für die Erstellung und Pflege geregelt sein.

Weitere wichtige Inhalte des Vertrages sind Kündigungsrechte, Details zu Abnahme und Gewährleistung sowie zu Nutzungsrechten an der zu erstellenden Software bzw. an definierten Teilen davon.

3.8.1 Rahmenvereinbarung

In Abstimmung mit dem UBA wurde ein Vertragsmuster für eine Rahmenvereinbarung entworfen. Dieses liegt beim UBA vor und kann dort angefragt werden. Darin werden alle Regelungen getroffen, die über das gesamte Projekt gelten. Über die Rahmenvereinbarung werden Einzelabrufe vereinbart, für die jeweils ein EVB-IT-Vertrag abgeschlossen wird. Bei den EVB-IT-Verträgen sind unterschiedliche Typen möglich (z. B. EVB-IT-Erstellung, EVB-IT-Dienstleistung, EVB-IT-Überlassung). Einzelabrufe werden entweder zum Festpreis oder nach Aufwand (mit Obergrenze) abgeschlossen. Die Projektleitung des UBA muss im Blick haben, dass die Summe der Einzelabrufe das Jahresbudget nicht überschreitet. Dieses Jahresbudget wird vom UBA auf Basis der Schätzungen aus der Machbarkeitsstudie sowie der Fortschreibung im Projektverlauf festgelegt. Die Einzelabrufe sind eigenständige Verträge und, im Falle von Werkverträgen, kann es Abnahmen geben. Zahlungsmodalitäten werden je Einzelabruf geregelt. Ein Einzelabruf kann einen oder mehrere Sprints umfassen.

3.8.2 Zusammenwirken mit dem Vorgehensmodell

Bezüglich der Koordinierung der Sprintzyklen mit den Einzelabrufen der Rahmenvereinbarung gilt, dass diese Einzelabrufe mit den Sprintzyklen synchronisiert sein sollten. Die Einzelabrufe sind dabei sehr gut geeignet, um Projektmeilensteine zu formulieren und bieten die Gelegenheit für Teilabnahmen, Teilzahlungen oder Präsentationen.

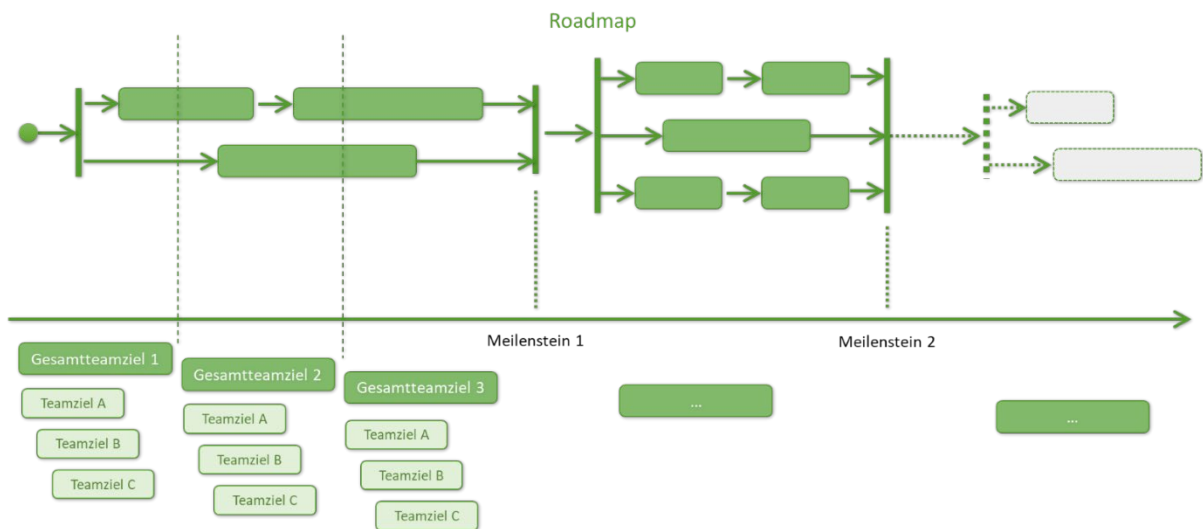
Insgesamt ist die Frequenz des Abrufzyklus geringer als die des Sprintzyklus, so dass es weniger Einzelabrufe/Abnahmen/Meilensteine gibt als Sprints auf der Ebene des Gesamtteams durchgeführt werden.

Die Abfolge der Einzelabrufe/Abnahmen/Meilensteine kann auch auf der Roadmap abgebildet und bereits zu Beginn des Vorhabens festgelegt werden. Dies kann zunächst auch geschehen, ohne für alle Meilensteine feste Termine vorzusehen. Diese können flexibel im Lauf des Projekts festgelegt werden.

Der Zusammenhang zwischen Abnahmezyklus und Sprintzyklus verhindert nicht, dass mehrere Einzelabrufe gleichzeitig bearbeitet werden können. Es ist nur darauf zu achten, dass sie durch jeweils eine eigene Abfolge von Sprintzyklen umgesetzt werden.

Das Zusammenwirken der einzelnen Meilensteine aus der Roadmap mit den Sprints des Gesamtteams und der einzelnen Entwicklungsteams zeigt Abbildung 7. In der Roadmap werden Schritte gesucht, die sich gut als Meilenstein eignen und für diese wird ein Einzelabruf formuliert und beauftragt. In einem oder mehreren Sprints wird der Einzelabruf abgearbeitet und abgenommen. Mit der Abnahme ist dann der Meilenstein erreicht.

Abbildung 7: Zyklen im Entwicklungsprozess



Quelle: Eigene Darstellung, con terra GmbH

4 Quellenverzeichnis

- [1] G. Börner, M. Bluhm, T. Fechner, R. Illes, B. Lubahn, M. Ostkamp, S. Richter, M. Schromm, U. Voges, J. von Zadelhoff, H. Rudolf, L. Hantsche, E. M. Lütkemeyer, M. Zschiesche, M. Niebuhr und R. Nöske, „Umwelt- und Naturschutzinformationssystem UNIS-D - Machbarkeitsstudie,“ Umweltbundesamt, Dessau-Roßlau, 2021.
- [2] J. Appelo, Management 3.0: Leading Agile Developers, Developing Agile Leaders, Boston: Pearson Education, Inc., 2011.
- [3] „Agile Manifesto,“ 2001. [Online]. Available: <https://agilemanifesto.org/iso/de/manifesto.html>. [Zugriff am 08 09 2021].
- [4] M. Cohn, Succeeding with Agile: Software Development Using Scrum, Upper Saddle River, NJ: Addison-Wesley, 2010.
- [5] R. C. Martin, Clean Agile: Back to Basics, Pearson, 2019.
- [6] D. F. Termer und T. Zehler, Agilität in Organisationen - Ein Leitfaden für die Praxis, Bitkom Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V., 2020.
- [7] S. Ockerman und S. Reindl, Mastering Professional Scrum: A Practitioner’s Guide to Overcoming Challenges and Maximizing the Benefits of Agility, Addison-Wesley Professional, 2019.
- [8] „The 2020 Scrum Guide,“ 2020. [Online]. Available: <https://scrumguides.org/scrum-guide.html>. [Zugriff am 08 09 2021].
- [9] M. Burrows, Kanban: Verstehen, einführen und anwenden, Heidelberg: dpunkt.verlag GmbH, 2015.
- [10] K. Beck und C. Andres, Extreme Programming Explained: Embrace Change, Addison-Wesley Professional, 2004.
- [11] R. Pichler, S. Roock und A. Beppe, Agile Entwicklungspraktiken mit Scrum, Heidelberg: dpunkt-Verlag, 2011.
- [12] „Kanban University Materials,“ Kanban University, [Online]. Available: <https://resources.kanban.university/materials/>. [Zugriff am 26 10 2021].
- [13] R. Jeffries, „Essential XP: Card, Conversation, Confirmation,“ 30 08 2001. [Online]. Available: <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>. [Zugriff am 08 09 2021].
- [14] Management 3.0, „Moving Motivators,“ [Online]. Available: <https://management30.com/practice/moving-motivators/>. [Zugriff am 08 09 2021].
- [15] „Scrum Alliance,“ [Online]. Available: <https://www.scrumalliance.org/>. [Zugriff am 08 09 2021].
- [16] „Scrum.org,“ [Online]. Available: <https://www.scrum.org/>. [Zugriff am 08 09 2021].
- [17] M. Skelton, M. Pais und R. Malan, Team Topologies: Organizing Business and Technology Teams for Fast Flow, IT Revolution Press, 2019.

- [18] C. Verwijs und D. Russo, „A Theory of Scrum Team Effectiveness,“ *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 26 05 2021.
- [19] E. Derby und D. Larsen, *Agile Retrospectives: Making Good Teams Great*, Raleigh, NC: Pragmatic Bookshelf, 2006.
- [20] D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Boston: Pearson Education, Inc., 2011.
- [21] S. Zörner, *Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten*, München: Hanser, 2015.
- [22] H. Kniberg, *Scrum and XP from the Trenches: How We Do Scrum*, Lulu.com, 2007.
- [23] M. Cohn, *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional, 2004.
- [24] B. Wake, „INVEST in Good Stories, and SMART Tasks,“ XP123, 17 August 2003. [Online]. Available: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>. [Zugriff am 14 11 2022].
- [25] M. Cohn, „Split Stories Using SPIDR,“ [Online]. Available: <https://www.mountaingoatsoftware.com/exclusive/spidr-poster-download>. [Zugriff am 08 09 2021].
- [26] M. Cohn, *Agile Estimating and Planning*, Prentice Hall Professional Technical Reference, 2006.
- [27] I. Lemberger, „Planning Poker – Die agile Aufwandsschätzung,“ ADITO, 14 Juli 2021. [Online]. Available: <https://www.adito.de/knowhow/blog/planning-poker>. [Zugriff am 04 Oktober 2022].
- [28] M. Holmgren, „Bucket Estimation – How to estimate a really large backlog,“ crisp., 03 06 2018. [Online]. Available: <https://blog.crisp.se/2018/06/03/mathiasholmgren/bucket-estimation-how-to-estimate-a-really-large-backlog>. [Zugriff am 06 10 2022].
- [29] C. Dibbern, „Magic Estimation: Magische mittelfristige Planung,“ teamprove, 12 12 2017. [Online]. Available: <https://www.teamprove.de/blog/magic-estimation-die-magische-methode-fuer-mittelfristige-planungen/>. [Zugriff am 06 10 2022].
- [30] L. Hohmann, „What is buy-a-feature?,“ Lucidspark, 2022. [Online]. Available: <https://lucidspark.com/templates/buy-a-feature>. [Zugriff am 4 Oktober 2022].
- [31] D. A. Norman und S. W. Draper, *User Centered System Design: New Perspectives on Human-computer Interaction*, London: Taylor & Francis, 1986.
- [32] A. G. Coenenberg, T. M. Fischer und T. Günther, *Kostenrechnung und Kostenanalyse 6. Auflage*, Stuttgart: Schäffer-Pöschel, 2007.
- [33] J. Patton, *User Story Mapping: Discover the Whole Story, Build the Right Product*, O'Reilly Media, Inc., 2014.
- [34] R. Pichler, „THE GO PRODUCT ROADMAP,“ 02 2020. [Online]. Available: <https://www.romanpichler.com/tools/the-go-product-roadmap/>. [Zugriff am 09 09 2021].
- [35] G. D. Angermeier, „Burndown-Chart,“ *Projekmagazin*, 6 März 2019. [Online]. Available: <https://www.projektmagazin.de/glossarterm/burndown-chart>. [Zugriff am 4 Oktober 2022].

- [36] „The Evidence-Based Management Guide - Measuring Value to Enable Improvement and Agility,“ 2020. [Online]. Available: <https://www.scrum.org/resources/evidence-based-management-guide>. [Zugriff am 28 10 2021].
- [37] J. Andresen, Erfolgreiche Retrospektiven - Ablauf, Regeln & Methoden, Leanpub, 2013.
- [38] T. Torres, „Product Discovery Basics: Everything You Need to Know,“ Product Talk, 18 08 2021. [Online]. Available: <https://www.producttalk.org/2021/08/product-discovery/>. [Zugriff am 06 10 2022].
- [39] H. Obendorf, I. Petersen, R. Gibbert und T. Memmel, „Agile UX. Wege zur agilen nutzerzentrierten Entwicklung – eine Einführung,“ *Tagungsband UP10*, pp. 175-179, 2010.
- [40] Wikipedia-Autoren, „DevOps,“ Wikipedia – Die freie Enzyklopädie., 29 04 2021. [Online]. Available: <https://de.wikipedia.org/w/index.php?title=DevOps&oldid=211412763>. [Zugriff am 10 09 2021].
- [41] Wikipedia-Autoren, „Software Craftsmanship,“ Wikipedia – Die freie Enzyklopädie., 18 05 2021. [Online]. Available: https://de.wikipedia.org/w/index.php?title=Software_Craftsmanship&oldid=212100063. [Zugriff am 10 09 2021].
- [42] G. Kim, P. Debois, J. Humble und J. Allspaw, *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, Portland, OR: IT Revolution Press, 2016.
- [43] P. M. Duvall, S. Matyas und A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*, Upper Saddle River, NJ: Addison-Wesley, 2007.
- [44] K. Beck, *Test-Driven Development by Example*, Boston: Pearson Education, Inc., 2003.
- [45] V. Vernon, *Domain-Driven Design kompakt*, Heidelberg: dpunkt.verlag GmbH, 2017.
- [46] R. C. Martin, *Clean Code - A Handbook of Agile Software Craftsmanship*, Boston: Pearson Education, Inc, 2009.
- [47] „Architekturrichtlinie für die IT des Bundes,“ Der Beauftragte der Bundesregierung für Informationstechnik, 2020. [Online]. Available: https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/Architekturrichtlinie-IT-Bund/architekturrichtlinie_it_bund_node.html. [Zugriff am 08 09 2021].
- [48] Microsoft, „Microsoft Teams,“ Microsoft, [Online]. Available: <https://www.microsoft.com/de-DE/microsoft-teams/group-chat-software>. [Zugriff am 06 10 2022].
- [49] webex by CISCO, „Eine App für alles. Und alle.,“ CISCO, [Online]. Available: <https://www.webex.com/de/index.html>. [Zugriff am 06 10 2022].
- [50] slido, „Explore Slido,“ slido, [Online]. Available: <https://www.slido.com/product>. [Zugriff am 06 10 2022].
- [51] Mentimeter, „Audience Engagement Platform,“ Mentimeter, [Online]. Available: <https://www.mentimeter.com/features>. [Zugriff am 06 10 2022].
- [52] GitLab, „Use GitLab,“ GitLab, [Online]. Available: <https://docs.gitlab.com/ee/user/>. [Zugriff am 06 10 2022].
- [53] Microsoft, „Microsoft Planner,“ Microsoft, [Online]. Available: <https://www.microsoft.com/de-de/microsoft-365/business/task-management-software>. [Zugriff am 06 10 2022].

- [54] Atlassian, „Jira Software,“ Atlassian, [Online]. Available: <https://www.atlassian.com/de/software/jira>. [Zugriff am 06 10 2022].
- [55] Jenkins, „Jenkins Handbook,“ Jenkins, [Online]. Available: <https://www.jenkins.io/doc/book/>. [Zugriff am 11 10 2022].
- [56] U. Vogenschow, B. Schneider und I. Meyrose, Soft Skills für Softwareentwickler: Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle, Heidelberg: dpunkt.Verlag GmbH, 2014.
- [57] Retrium, „Facilitated retrospectives,“ Retrium, [Online]. Available: <https://www.retrium.com/>. [Zugriff am 11 10 2022].
- [58] Miro, „Die visuelle Kommunikationsplattform,“ Miro, [Online]. Available: <https://miro.com/de/>. [Zugriff am 11 10 2022].
- [59] Collaboard, „Collaboard - Das sichere Online-Whiteboard,“ Collaboard, [Online]. Available: <https://www.collaboard.app/de/>. [Zugriff am 11 10 2022].
- [60] Microsoft, „Microsoft 365,“ Microsoft, [Online]. Available: <https://www.microsoft.com/de-DE/microsoft-365>. [Zugriff am 11 10 2022].
- [61] The Document Foundation, „LibreOffice - Free Office Suit,“ The Document Foundation, [Online]. Available: <https://www.libreoffice.org/>. [Zugriff am 11 10 2022].
- [62] Fraunhofer FIT und OrbiTeam Software GmbH & Co. KG, „BSCW - Groupware für effiziente Teamarbeit,“ OrbiTeam Software GmbH & Co. KG, [Online]. Available: <https://www.bscw.de/classic/>. [Zugriff am 11 10 2022].
- [63] O. Wolf, Agile Softwareentwicklung: Vertragsgestaltung und Praxistipps, <https://www.ra-plutte.de/agile-softwareentwicklung-vertragsgestaltung-praxistipps>, 2021.
- [64] Wikipedia, „Bus Faktor,“ [Online]. Available: https://en.wikipedia.org/wiki/Bus_factor. [Zugriff am 08 09 2021].

