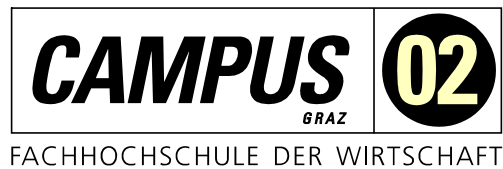


MASTERARBEIT

BESCHLEUNIGUNG DER LEISTUNGSERBRINGUNG IN ITIL ORIENTIERTEN IT-ORGANISATIONEN

ausgeführt am



Studiengang
Informationstechnologien und Wirtschaftsinformatik

Von: Stefan Weigl
Personenkennzeichen: 1610320013

Graz, am 22. März 2018

.....
Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meines Studiums und während des Verfassens dieser Arbeit unterstützt und motiviert haben.

Mein besonderer Dank gilt Herrn FH-Hon.Prof. Ing. Dipl.-Ing. (FH) Dipl.-Ing. Dr.techn. Michael Georg Grasser, MBA MPA, der meine Masterarbeit betreut und begutachtet hat. Für die Geduld, die hilfreichen Anregungen und die konstruktive Kritik möchte ich mich herzlich bedanken.

KURZFASSUNG

Unternehmen stehen vor der Herausforderung, die Geschwindigkeit der Leistungserbringung an die Geschwindigkeit der sich verändernden Rahmenbedingungen und Anforderungen anzupassen. Unternehmen stehen vor der Herausforderung, sich mit ihren etablierten Prozessen gegen die Flexibilität von Startups durchzusetzen. Schnell fortschreitende Entwicklungen erfordern kürzere Innovationszyklen in der IT. Zeitgleich sollen IT Services weiterhin stabil und wirtschaftlich betrieben werden. IT-Organisationen, die Prozesse angelehnt an das weitverbreitete IT Service Management Best Practices Framework ITIL etabliert haben, sind gefordert, die Geschwindigkeit der Leistungserbringung zu erhöhen, ohne die Stabilität des Systems negativ zu beeinträchtigen.

In dieser Arbeit wird zunächst ein Arbeitsablauf zur Leistungserbringung und dessen Durchlaufzeit beobachtet. Die Beobachtung findet in einem Großunternehmen statt, welches Prozesse etabliert hat, die sich an ITIL orientieren. Im Anschluss daran wird das IT Service Management Framework ITIL beschrieben und die darin enthaltenen Prozesse den beobachteten Arbeitsschritten gegenübergestellt. Unter Zuhilfenahme einer Literaturrecherche werden Ansätze identifiziert, die eine Beschleunigung der Leistungserbringung versprechen. Als weiterer Schritt werden Anwendungsfälle aus den Beobachtungen ausgewählt und Lösungsansätze entwickelt, die die Vorteile aus ITIL und DevOps vereinen, um eine schnelle Leistungserbringung bei hoher Systemstabilität zu ermöglichen. Auf Basis der identifizierten Lösungsansätze wird ein adaptierter Arbeitsablauf beschrieben und Rückschlüsse auf die potentielle Zeiteinsparung bei gleichbleibendem Ressourceneinsatz gezogen. Durch den Einsatz einer Deployment Pipeline pro Service wird ein standardisierter Weg zur Überführung von Änderungen in die Produktivumgebung erzeugt. Die hohe Anzahl an automatisierten Tests innerhalb der Deployment Pipeline sorgt für eine hohe Systemstabilität. Aufgrund des dadurch verringerten Risikos können IT Change Management Prozesse als Typ „Standard-Change“ anstatt „Normal-Change“ durchgeführt werden. Hierdurch werden kürzere Deploymentzyklen zugelassen und durch den Einsatz von agilen Softwareentwicklungsframeworks, die häufige Deployments vorsehen, eine frühere Nutzung der Software ermöglicht wird.

ABSTRACT

Businesses are facing challenges to adapt their speed of service delivery to changing framework conditions and requirements. Businesses with long established processes are facing challenges to compete with the flexibility of start-ups. Fast progressing developments require shorter innovation cycles within IT. Simultaneously IT services should be stable and economically operated. IT organisations, which established their processes inspired by widespread IT Service Management Best Practices Framework ITIL, are challenged to increase the speed of the service delivery without affecting the stability of the systems negatively.

In this thesis, a workflow for service performance is described. The investigation takes place in a large enterprise with established processes inspired by ITIL. Following this, the IT service management framework, ITIL, is described and the processes involved compared with the observed working steps. With the help of literature research, approaches are identified which promise to accelerate service delivery. As a further step, cases of application from the observation are selected and solutions developed, which combine advantages from ITIL and DevOps, to allow faster service delivery with high system stability. Based on the identified solution approaches, an adapted workflow is described and conclusions of the time saving at steady use of resources are drawn. With the use of a deployment pipeline for each service a standardised path to transfer changes into a productive system is created. The high amount of test automation within the deployment pipeline ensures high system stability. Due to the reduced risk, IT change management processes can be realised as a “standard change” instead of “normal change”. Hereby it allows shorter deployment cycles and the application of agile software development frameworks, which allow earlier use of the software.

INHALTSVERZEICHNIS

1	EINLEITUNG	9
1.1	Zielsetzung	9
1.2	Vorgehen und Methodik	9
1.3	Aufbau der Arbeit.....	10
2	ABLAUF DER LEISTUNGSERBRINGUNG	13
2.1	Bedarfsidentifizierung und Umsetzungsentscheidung	13
2.2	Lösungsentwicklung	16
2.2.1	Vorbereitung zur Lösungsentwicklung	16
2.2.2	Entwicklung der Lösung	18
2.3	Test und Inbetriebnahme.....	20
2.3.1	Vorbereitung des Deployments	20
2.3.2	Überblick Change Management Prozess	23
2.3.3	Überführung in die Produktivumgebung	26
2.4	Kurzzusammenfassung	29
3	IT SERVICE MANAGEMENT NACH ITIL.....	30
3.1	ITIL Grundlagen.....	30
3.2	Service Strategy	31
3.2.1	Strategy Management for IT Services	32
3.2.2	Service Portfolio Management	33
3.2.3	Financial Management for IT Services	34
3.2.4	Demand Management	34
3.2.5	Business Relationship Management	34
3.2.6	Kurzzusammenfassung	34
3.3	Service Design.....	35
3.3.1	Design Coordination	36
3.3.2	Service Catalogue Management	37
3.3.3	Service Level Management	37
3.3.4	Availability Management.....	38
3.3.5	Capacity Management.....	38

3.3.6	IT Service Continuity Management	38
3.3.7	Information Security Management.....	39
3.3.8	Supplier Management	39
3.3.9	Service Design Package	39
3.3.10	Kurzzusammenfassung	40
3.4	Service Transition	40
3.4.1	Transition Planning and Support	41
3.4.2	Change Management	42
3.4.3	Service Asset and Configuration Management	42
3.4.4	Release and Deployment Management	43
3.4.5	Service Validation and Testing	44
3.4.6	Change Evaluations	45
3.4.7	Knowledge Management.....	45
3.4.8	Kurzzusammenfassung	46
3.5	Service Operation	46
3.5.1	Event Management	48
3.5.2	Incident Management	48
3.5.3	Request Fulfilment.....	49
3.5.4	Problem Management	49
3.5.5	Access Management	49
3.5.6	Funktionen	49
3.5.7	Kurzzusammenfassung	50
3.6	Continual Service Improvement	50
3.6.1	Seven-Step Improvement Process.....	51
3.6.2	Kurzzusammenfassung	53
4	ANSÄTZE	54
4.1	Agile	54
4.2	Scrum	55
4.2.1	Rollen.....	56
4.2.2	Artefakte	57
4.2.3	Ereignisse	58
4.2.4	Zusammenfassung	59
4.3	Lean Management.....	60
4.3.1	Verschwendung (Muda)	61

4.3.2	Unausgeglichenheit (Mura)	62
4.3.3	Überlastung (Muri)	63
4.3.4	Prinzipien	63
4.3.5	minimum viable product.....	64
4.4	DevOps	64
4.4.1	Prinzipien	64
4.4.2	CAMS	66
4.4.2.1.	Culture	66
4.4.2.2.	Automation.....	68
4.4.2.3.	Measurement.....	70
4.4.2.4.	Sharing	71
4.5	Zusammenfassung und Hypothese.....	72
5	BESCHLEUNIGUNG DER LEISTUNGSERBRINGUNG	73
5.1	Überprüfung der Relevanz	73
5.2	Anwendungsfall Anforderungsdefinition	75
5.2.1	Problembeschreibung.....	75
5.2.2	Lösungsansatz.....	75
5.2.3	Erfordernisse und Voraussetzungen	76
5.3	Anwendungsfall Test und Deployment	77
5.3.1	Problembeschreibung.....	77
5.3.2	Lösungsansatz.....	77
5.3.2.1.	Test und Deployment	78
5.3.2.2.	Dokumentation.....	79
5.3.2.3.	Change Management Prozess	79
5.3.3	Erfordernisse und Voraussetzungen	80
5.4	Anpassungen des Arbeitsablaufs	81
6	CONCLUSIO	94
	ANHANG A – UMFRAGE ZUR INTEGRATION VON DEVOPS IN ITIL	96
	ANHANG A1 – ERGEBNISSE DER UMFRAGE	97
	ABKÜRZUNGSVERZEICHNIS.....	106
	ABBILDUNGSVERZEICHNIS	107

TABELLENVERZEICHNIS	108
LITERATURVERZEICHNIS	109

1 EINLEITUNG

Vormals als Kostenstelle bezeichnet, mussten IT-Abteilungen erst wieder beweisen, dass sie durch effiziente Arbeitsweise und am Kerngeschäft orientierte technische Innovation, einen Beitrag zur Wertschöpfung eines Unternehmens leisten. Die neue Herausforderung für die IT ist, die Geschwindigkeit der Wertgenerierung an die Geschwindigkeit der sich verändernden Rahmenbedingungen und Anforderungen im geschäftlichen Umfeld anzupassen. Aus diesen Anforderungen resultiert der Zielkonflikt zwischen Geschwindigkeit der Leistungserbringung und Stabilität des Gesamtsystems. Die Stabilität in der IT wird versucht mit IT Service Management auf einem hohen Niveau zu halten. Die IT Infrastructure Library (ITIL), eine Sammlung von Best-Practice-Vorschlägen, hat sich in den letzten Jahrzehnten als defacto Standard im Bereich von IT Service Management etabliert. Die standardisierten Prozesse, die für neue und/oder geänderte Services durchlaufen werden müssen, nehmen Einfluss auf die Geschwindigkeit der Leistungserbringung. Unternehmen, die Prozesse auf Basis von ITIL etabliert haben, rein nach traditionellen Modellen arbeiten, wasserfallmodellbasierte Projektabwicklung und Funktionen mit starren Hierarchien propagieren, werden aufgrund der daraus resultierenden langen Durchlaufzeiten bei Changes und Projekten zunehmend in Bedrängnis gebracht.

In dieser Arbeit wird untersucht, wie die Geschwindigkeit der Leistungserbringung in an ITIL orientierten IT-Organisationen durch den Einsatz von ausgewählten Methoden und Ansätzen erhöht werden kann. Dabei wird darauf geachtet, dass die durch den Einsatz von IT Service Management nach ITIL gewonnene Stabilität der IT-Systeme nicht negativ beeinträchtigt wird.

1.1 Zielsetzung

Diese Arbeit zeigt Möglichkeiten auf, durch welche die Durchlaufzeiten von Softwareentwicklungsprojekten und Changes in IT Services verringert werden können. Hauptaugenmerk dabei liegt in der Beibehaltung der Systemstabilität, die durch den Einsatz von an ITIL angelehnten Prozessen gewonnen wird.

Die Arbeit trägt dazu bei die folgende Forschungsfrage zu beantworten:

„Wie kann die Geschwindigkeit der Leistungserbringung in an ITIL angelehnten Prozessen erhöht werden, ohne die durch ITIL gewonnene Stabilität zu verringern?“

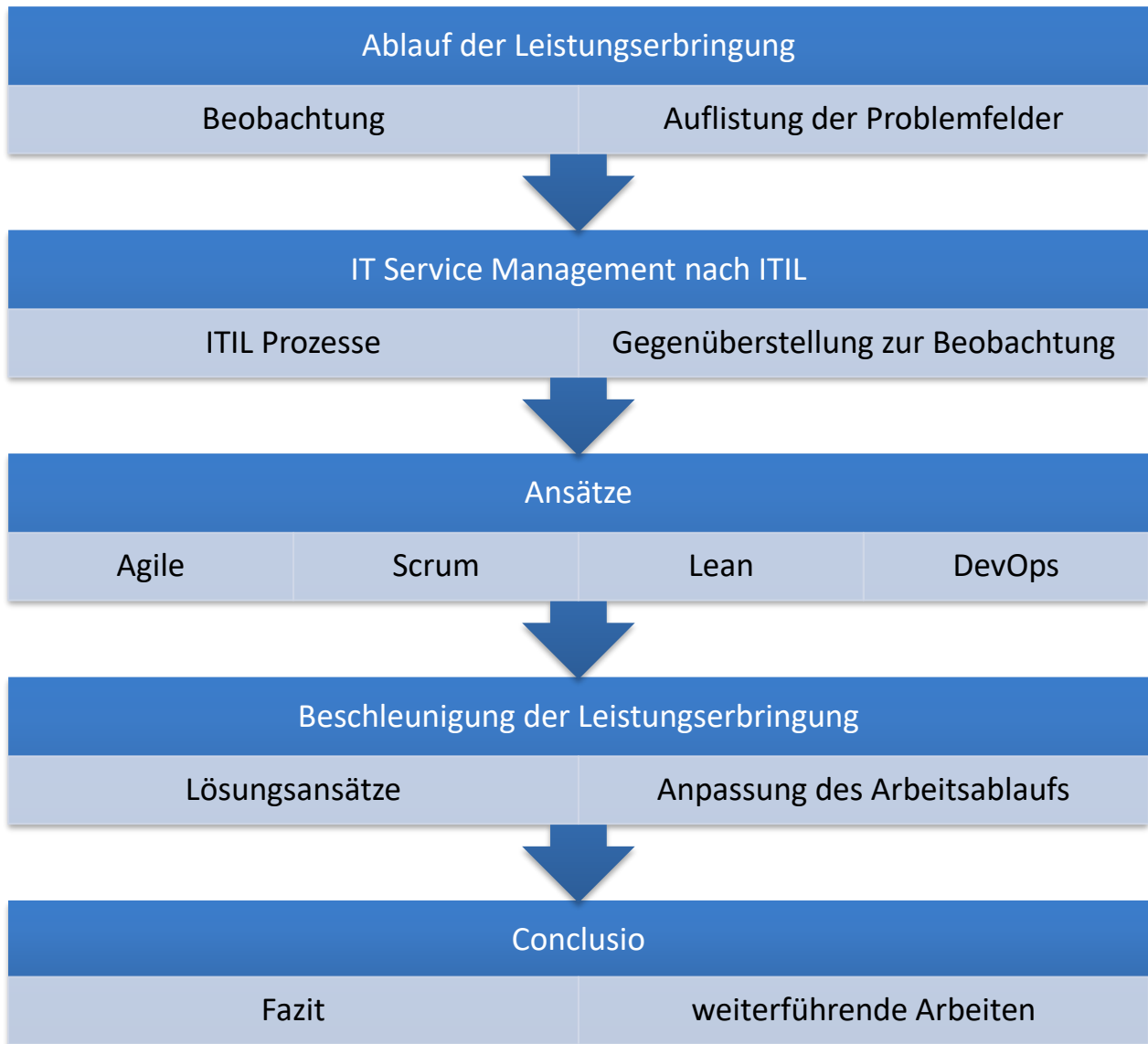
1.2 Vorgehen und Methodik

Zur Bearbeitung des Forschungsthemas wird zu Beginn eine Beobachtung eines Arbeitsablaufs zur Leistungserbringung in einem Betrieb mit an ITIL angelehnten Prozessen durchgeführt. Während der Beobachtung wird das Augenmerk auf die Durchlaufzeit gelegt.

1 Auf Basis der Beobachtungen werden Problemfälle aufgelistet, die einen hohen Einfluss auf die
2 Durchlaufzeit ausüben, und Ursachen für die Verzögerung aufgezeigt. Anschließend wird das
3 Rahmenwerk ITIL näher beschrieben und die in ITIL angeführten Prozesse den beobachteten
4 Tätigkeiten gegenübergestellt. Als theoretische Grundlage dienen die ITIL Rahmenwerke Service
5 Strategy, Service Design, Service Transition, Service Operations und Continual Service
6 Improvement, die jeweils vom britischen Cabinet Office im Jahr 2013 zur Verfügung gestellt
7 wurden. Ausgehend von der Beobachtung werden unter Zuhilfenahme einer Literaturrecherche
8 Ansätze beschrieben, die eine Verkürzung der Durchlaufzeit versprechen. In einer
9 theoriebasierten Exploration werden unterschiedliche Ansätze, die im Kontext der
10 Softwareentwicklung eine Verkürzung der Durchlaufzeit versprechen, betrachtet und daraus die
11 Arbeitshypothese erstellt. Mit Hilfe einer quantitativen Umfrage wird überprüft, ob die Vorteile, die
12 eine Integration von Ansätzen aus DevOps verspricht, für Organisationen mit an ITIL angelehnten
13 Prozessen relevant sind. Für die aufgelisteten Problemfelder werden Lösungsansätze
14 beschrieben, die auf Methoden und Prinzipien der identifizierten Ansätze basieren. Im Anschluss
15 daran wird ein Gedankenexperiment durchgeführt, welches die beschriebenen Lösungsansätze
16 mit dem beobachteten Arbeitsablauf kombiniert. Daraus resultiert ein adaptierter Arbeitsablauf,
17 der durch Integration von Ansätzen aus DevOps, Lean und Agile die Geschwindigkeit der
18 Leistungserbringung bei gleichzeitiger Beibehaltung der Systemstabilität erhöht.

19 **1.3 Aufbau der Arbeit**

20 In diesem Abschnitt erfolgt die Beschreibung über den Aufbau der vorliegenden
21 Forschungsarbeit. Abbildung 1 stellt die Struktur grafisch dar und soll einen Überblick über den
22 Aufbau der Arbeit geben.



- 1
- 2 *Abbildung 1: Aufbau der Arbeit*
- 3 In Abbildung 1 ist ersichtlich, dass mit Kapitel 2, dem Ablauf der Leistungserbringung, begonnen
- 4 wird. Dieses Kapitel enthält die Beobachtung eines typischen Ablaufs zur Leistungserbringung in
- 5 einer IT Organisation mit an ITIL angelehnten Prozessen und eine Auflistung der identifizierten
- 6 Problemfelder.
- 7 Kapitel 3 beschäftigt sich mit IT Service Management nach ITIL. Es werden die fünf Phasen des
- 8 Service Life Cycle und deren Prozesse beschrieben. Den ITIL Prozessen werden im Zuge der
- 9 Prozessbeschreibung die Tätigkeiten aus Kapitel 2 in Relation gestellt.
- 10 In Kapitel 4 werden Ansätze beschrieben, die eine Beschleunigung der Leistungserbringung
- 11 unterstützen können. Es werden Agile und das agile Framework Scrum betrachtet, Lean
- 12 beschrieben und auf die Ideen hinter DevOps eingegangen. In diesem Kapitel wird die
- 13 Hypothese, die dieser Arbeit zugrunde liegt, beschrieben.

- 1 In Kapitel 5 werden Lösungsansätze beschrieben, die die Problemfälle aus Kapitel 2 betreffen. In
- 2 weiterer Folge werden diese Lösungsansätze in einem Gedankenexperiment in den
- 3 Gesamtarbeitsablauf der Leistungserbringung integriert.
- 4 Das letzte Kapitel fasst die Erkenntnisse zusammen und gibt einen Ausblick über mögliche
- 5 weiterführende Forschungen.

2 ABLAUF DER LEISTUNGSERBRINGUNG

Dieses Kapitel beschäftigt sich mit der systematischen Beobachtung eines Tätigkeitsablaufs zur Leistungserbringung in einer IT Organisation, die an ITIL angelehnte Prozesse implementiert hat.

Das Objekt der Beobachtung ist ein Arbeitsablauf zur Leistungserbringung in einem Großunternehmen mit an ITIL angelehnten Prozessen, welches in weiterer Folge als Basis- und Referenzinformation für die Beschleunigung der Leistungserbringung dient. Die Leistungserbringung startet bei der Kommunikation des Bedarfs und endet bei der Nutzung durch den Kunden. Aus diesem Grund werden die Bereiche der Kontaktaufnahme und Erstaufnahme der Anforderungen nur am Rande beschrieben. Das Hauptaugenmerk liegt auf dem prozessualen Tätigkeitsablauf, weshalb der Inhalt des Projekts lediglich in einem Genauigkeitsgrad dokumentiert wird, der für die weitere Bearbeitung und die Leistungssteigerung erforderlich ist.

Eine systematische Beobachtung liegt nach Peter Atteslander (2010) unter anderem dann vor, wenn die Beobachtung in einen theoretischen Bezugsrahmen eingearbeitet wird und systematisch geplant wird. Da es sich bei Projekten um Konstellationen mit einmaligen Rahmenbedingungen handelt, können die weiteren Kriterien der systematischen Beobachtung in Bezug auf die Reproduzierbarkeit nicht erfüllt werden.

Als Beobachtungsmethode dient die teilnehmende, verdeckte Beobachtung. Dies bedeutet, dass der Beobachter Teil der Beobachtungssituation ist, das Beobachtungsumfeld allerdings nicht weiß, dass es beobachtet wird.

Ziel des beobachteten Tätigkeitsablaufs ist die Digitalisierung eines Angebotsfreigabeprozesses für Entwicklungsprojekte im Automotive Bereich.

Der Tätigkeitsablauf gliedert sich in 3 Teilbereiche:

- Bedarfsidentifizierung und Umsetzungsentscheidung
- Lösungsentwicklung
- Test und Inbetriebnahme

Der Beobachtungszeitraum erstreckt sich von Mai 2017 bis August 2018. Um die vergangene Zeitspanne messen zu können, ist bei jedem Einzelschritt das Fertigstellungsdatum hinterlegt. Die Beobachtungen wurden von der Person, die die Rolle des Consultants einnimmt, durchgeführt. Im späteren Verlauf der Beobachtung übernimmt dieselbe Person die Rolle des Projektleiters.

2.1 Bedarfsidentifizierung und Umsetzungsentscheidung

Der erste Teilbereich startet mit der Kommunikation des Bedarfs des Fachbereichs und umfasst alle Tätigkeiten, die zur Erwirkung einer Umsetzungsentscheidung erforderlich sind.

In Tabelle 1 sind die inhaltlichen und zeitlichen Abläufe aufgelistet.

ID	Datum	Inhalt
1	09.05.2017	Eine Vertretergruppe eines Fachbereichs (in weiterer Folge Auftraggeber genannt) kommuniziert mittels Request im Ticketsystem den Wunsch nach technischer Unterstützung bei der Digitalisierung des Angebotsprozesses und beschreibt die groben Anforderungen.
2	10.05.2017	Auf Basis von Ressourcenverfügbarkeit, betrieblichem Umfeld des Auftraggebers und vermuteten technischen Umfeld der potenziellen Lösung wird ein geeigneter Vertreter der IT Abteilung (in weiterer Folge Consultant genannt) nominiert, der sich um die weitere Bearbeitung der Anfrage kümmert.
3	10.05.2017	Um ein gemeinsames Verständnis der Problemstellung und der Anforderungen zu schaffen, kontaktiert der Consultant den Auftraggeber und vereinbart einen persönlichen Termin.
4	23.05.2017	Der Consultant und der Auftraggeber treffen sich, um die Problemstellung näher zu beleuchten. Der Consultant dokumentiert die Problemstellung und die Rahmenbedingungen, das angestrebte Ziel und die groben Vorstellungen der potenziellen Lösung des Kunden. Die Anforderungen des Kunden sehen die Entwicklung einer individuellen Softwarelösung vor, die auf Funktionalitäten von bereits bestehenden Standardsystemen (Microsoft SharePoint und Salesforce) aufbaut und eine integrierte Lösung über die Systemgrenzen hinaus repräsentiert.
5	24.05.2017	Der Consultant stellt die Anforderungen des Auftraggebers bereits existierenden Lösungen anderer Fachbereiche gegenüber, um zu überprüfen, ob eine ähnliche Lösung bereits vorhanden ist. Diese Prüfung fällt negativ aus.
6	29.05.2017	Der Consultant fasst die Anforderungen und die Ziele zusammen, listet die betroffenen Geschäftsprozesse und potenziell involvierten Systeme der Lösung auf und initiiert einen digitalen Freigabeprozess, in dem entschieden wird, ob die IT-Anforderung weiterverfolgt wird. Die Entscheidung hängt unter anderem vom zu erwartenden Nutzen, der Strategiekonformität und der Verfügbarkeit von Ressourcen ab und kann in mehreren Iterationen erfolgen. Die Verantwortung für diesen Freigabeprozess hat die IT Steering Group.
7	08.06.2017	Die IT Steering Group entscheidet sich für eine Weiterverfolgung der IT-Anforderung und definiert, dass die Anforderung in Form eines Projekts umzusetzen ist. Der Consultant wird als zuständiger Projektleiter definiert.
8	08.06.2017	Der Projektleiter vereinbart einen Termin mit dem Auftraggeber, um die bereits grob dokumentierten Anforderungen näher zu spezifizieren. Ziel dieses Termins ist die Abgrenzung des Projektinhalts.

ID	Datum	Inhalt
9	20.06.2017	Der Termin zur näheren Abgrenzung des Projektinhalts wird abgehalten. Ziele und Nicht-Ziele werden definiert und zusammengefasst.
10	22.06.2017	Der Projektleiter stellt auf Basis des dokumentierten Projektinhalts das gewünschte Projektteam zusammen und vereinbart mit den vier gewünschten Projektkerneammitgliedern einen Termin, um eine Aufwandsschätzung für die dokumentierten Anforderungen zu erstellen.
11	28.06.2017	Das Kernteam trifft sich zur Erstellung der Aufwandsschätzung und zur Identifikation von möglichen Risiken. Auf Basis dieser Informationen wird ein Projektantrag und aufgrund der zu erwartenden Aufwände ein Business Case erstellt. Sowohl dem Projektkerneam als auch dem Projektleiter ist klar, dass der zur Verfügung stehende Detaillierungsgrad der Anforderungen für eine Aufwandsschätzung mit hoher Schätzgenauigkeit unzureichend ist. Anm.: Ein Business Case ist nach Brugger (2009) eine betriebswirtschaftliche Methode, um die Wirtschaftlichkeit eines Vorhabens auf Basis des voraussichtlichen Gesamtmiteinsatzes sowie der voraussichtlichen Einsparungen über den Nutzungszeitraum zu bewerten. Der Business Case dient als Entscheidungsgrundlage für die tatsächliche Umsetzung.
12	14.07.2017	Der fertiggestellte Business Case und der Projektantrag, der noch keinen Zeitplan beinhaltet, werden dem Budgetverantwortlichen in der Linienorganisation des Auftraggebers zur Freigabe vorgelegt. Da aufgrund interner Leistungsverrechnung für jeden in die IT ausgelagerten Aufwand ausgaberelevante Kosten entstehen, haben sich Auftraggeber und Auftragnehmer darauf geeinigt, dass Tätigkeiten wie Projektmarketing, Schulungen und das Erstellen von Anleitungen durch die Mitarbeiter des Fachbereichs selbst erbracht werden.
13	27.07.2017	Die Projektumsetzung ist finanziell genehmigt.

1 *Tabelle 1: zeitlicher Ablauf der Bedarfsidentifizierung und Umsetzungsentscheidung*

2 Tabelle 1 zeigt den zeitlichen Ablauf der Tätigkeiten, die zwischen der Kommunikation des
3 Bedarfs und der positiven Entscheidung zur Umsetzung erfolgen. Klar erkennbar sind die
4 teilweise langen Abstände zwischen den einzelnen Schritten. Die Gründe der vier zeitintensivsten
5 Schritte werden nachfolgend kurz erläutert.

6 Übergang Vereinbarung Ersttermin (ID 3) zu Abhaltung Ersttermin (ID 4): Der Grund für die Dauer
7 von 13 Tagen zwischen der Terminvereinbarung und der Abhaltung des Ersttermins waren
8 urlaubsbedingte Abwesenheiten der für den Ersttermin erforderlichen Personen.

9 Übergang Initiierung des Freigabeprozesses (ID 6) zu Freigabe zur inhaltlichen Weiterverfolgung
10 (ID 7): Die Dauer von 10 Tagen zwischen dem Start des Freigabeprozesses und der inhaltlichen

- 1 Freigabe liegt an der Frequenz, in der die IT Steering Group Entscheidungen trifft. Das Gremium
2 trifft sich üblicherweise im Zwei-Wochen-Rhythmus.
- 3 Übergang Terminvereinbarung für die nähere Abgrenzung des Inhalts (ID 8) und
4 Terminabhaltung (ID 9): Die Zeitspanne von 12 Tagen war mit beruflichen Abwesenheiten der
5 erforderlichen Personen begründet.
- 6 Übergang Vorlage von Projektantrag und Business Case (ID 12) und der finanziellen Freigabe
7 (ID 13): Die Ursache für die Dauer von 13 Tagen für die Freigabe war die
8 Ressourcenverfügbarkeit der entscheidenden Stelle.

9 **2.2 Lösungsentwicklung**

10 Der Bereich der Lösungsentwicklung beginnt mit dem Projektstartprozess und endet mit der
11 Fertigstellung der Entwicklungen. Diese Phase gliedert sich in die beiden Abschnitte Vorbereitung
12 zur Lösungsentwicklung und Entwicklung der Lösung. Das Zusammenführen der Entwicklungen
13 (Integration) und der Übergang in die Test- sowie Produktivumgebung werden im Kapitel 2.3
14 „Inbetriebnahme“ behandelt.

15 **2.2.1 Vorbereitung zur Lösungsentwicklung**

16 Der erste von zwei Abschnitten der Lösungsentwicklung beinhaltet die Vorbereitungen zur
17 Entwicklung der Lösung, die notwendig sind, um mit der Entwicklung zu beginnen. Tabelle 2 listet
18 die darin enthaltenen Arbeitsschritte auf.

ID	Datum	Inhalt
14	01.08.2017	Nach der finanziellen Freigabe startet der Projektleiter den Projektstartprozess gemäß der internen Projektabwicklungsrichtlinie. Diese sieht im Startprozess eine detailliertere Planung des Projekts vor. Neben Zielen, Aufgaben und Kosten werden auch Termine und personelle Ressourcen in einem höheren Detailgrad geplant. Der Projektleiter überträgt Punkte aus dem erstellten Projektantrag und dem Business Case in seine Planungsdokumentation. Er stellt das Projektkernteam aus vier Personen mit unterschiedlichen Kernkompetenzen zusammen und vereinbart einen Termin, um Arbeitspakete zu identifizieren und diese in den Projektstrukturplan zu übertragen. Die vier Kernteammitglieder werden jeweils von einer unbekanntem Anzahl an Personen bei der Abarbeitung ihrer Arbeitspakete unterstützt.
15	07.08.2017	Der Termin zur Identifizierung der Arbeitspakete und Erstellung des Projektstrukturplans wird abgehalten. Zwischen dem Projektleiter und dem Projektkernteam baut sich ein Zielkonflikt auf. Der Projektleiter möchte eine Planung erstellen, die als Basis für den weiteren Projektverlauf dient und für Controllingaufgaben als belastbar gilt. Das Projektkernteam möchte sich

ID	Datum	Inhalt
		<p>aufgrund zu undetaillierter Informationen auf keine belastbare Planung einlassen.</p> <p>Nach längeren Diskussionen einigt sich das Team darauf, eine Planung mit vielen Unbekannten zu erstellen. Im Gegenzug dazu wird einer Integration der vorgelagerten Phase „Requirements Engineering“ zugestimmt.</p> <p>Der erste Entwurf eines Projektstrukturplans ist erstellt, die Arbeitspakete mit fehlender Arbeitspaketspezifikation sind aufgelistet und der Ressourcenbedarf ist eingegrenzt.</p>
16	18.08.2017	<p>Der Projektleiter bespricht den Ressourcenbedarf mit den Vorgesetzten der einzelnen Projektteammitglieder und hinterlegt bei den Einträgen im Projektstrukturplan Planwerte im Bereich des Fertigstellungsdatums. Um möglichst wenig Zeit zu verlieren, werden jene Projektteammitglieder, die ressourcentechnisch unmittelbar verfügbar sind, damit beauftragt, mit dem Requirements Engineering zu beginnen und die Arbeitspaketspezifikation mit Inhalt zu befüllen.</p> <p>Anm.: Requirements Engineering ist nach Pohl und Rupp (2015) ein systematischer Ansatz, um Anforderungen zu spezifizieren und zu verwalten und gliedert sich in die vier Haupttätigkeiten: ermitteln, dokumentieren, prüfen und abstimmen sowie verwalten.</p>
17	10.11.2017	<p>Das Requirements Engineering ist abgeschlossen und die Anforderungen an die Lösung sind in den Arbeitspaketspezifikationen hinterlegt. Um die Fertigstellung zu erreichen, war eine Vielzahl an Terminen mit dem Auftraggeber und den zukünftigen Benutzern der Lösung erforderlich.</p>
18	22.11.2017	<p>Das Projektteam hat auf Basis der dokumentierten Anforderungsdetails ein Softwarearchitektur- und Infrastrukturkonzept entworfen, welches nach den Vorgaben des Projektabwicklungsprozesses auch die informationssicherheitsbezogene Relevanz in den Punkten Vertraulichkeit, Integrität und Verfügbarkeit abdeckt.</p> <p>Anm.: Die Punkte Vertraulichkeit, Integrität und Verfügbarkeit sind aufgrund der Zertifizierung des Unternehmens nach ISO 27001 ein erforderliches Kriterium.</p>
19	29.11.2017	<p>Aufgrund der ISO 27001 Relevanz der geplanten Umsetzung, ist die Abteilung „global IT Security“ zu involvieren. Der Projektleiter präsentiert das Softwarearchitektur- und Infrastrukturkonzept der unternehmensinternen Abteilung für IT Sicherheit. Aufgrund des klaren Umsetzungskonzepts, welches ebenfalls IT sicherheitsrelevante Punkte betrachtet, wird grünes Licht für die weitere Umsetzung gegeben.</p>

ID	Datum	Inhalt
20	01.12.2017	Der Projektleiter integriert die Konzepte in die Projektplanungsdokumentation und informiert das Projektteam, sodass die Umsetzung gestartet werden kann. Die Vorbereitung der Lösungsentwicklung ist dadurch abgeschlossen.

1 *Tabelle 2: zeitlicher Ablauf zur Vorbereitung der Lösungsentwicklung*

2 Tabelle 2 zeigt den zeitlichen Ablauf der Tätigkeiten, die zwischen der finanziellen Freigabe des
 3 Projekts und der Integration der Konzepte in die Projektplanungsdokumentation liegen. Einer der
 4 beschriebenen Schritte erfordert eine hohe Zeitspanne zur Fertigstellung, dessen Grund
 5 nachfolgend kurz beschrieben wird.

6 Übergang Start des Requirements Engineerings (ID 16) zu Fertigstellung Requirements
 7 Engineering (ID 17): Der Hauptgrund für die Dauer von 84 Tagen zwischen dem Start des
 8 Requirements Engineerings und dessen Beendigung, war der hohe Abstimmungsaufwand mit
 9 dem Kunden, um die Anforderung detailliert zu erheben. Häufig musste der Ansprechpartner auf
 10 Auftraggeberseite Entscheidungen für Funktionsweisen von der vorgesetzten Stelle bestätigen
 11 lassen. Des Weiteren waren Anforderungen von unterschiedlichen Vertretern der
 12 Auftraggeberseite widersprüchlich.

13 **2.2.2 Entwicklung der Lösung**

14 Der zweite Abschnitt der Lösungsentwicklung beinhaltet die eigentliche Entwicklung der Lösung,
 15 in der aufbauend auf die niedergeschriebenen Konzepte die Arbeitspakete laut Spezifikation
 16 umgesetzt werden.

17 Tabelle 3 beinhaltet eine Auflistung der inhaltlichen und zeitlichen Abläufe.

ID	Datum	Inhalt
21	01.12.2017	Die Softwareentwickler starten mit der Umsetzung der Arbeitspakete. Als reiner Umsetzungszeitraum sind sechs Monate veranschlagt. Anm.: Die Entwicklung selbst ist eine, in zwei Teile aufgeteilte, Individualsoftware. Den ersten Teil bildet die Erweiterung von SharePoint Funktionalitäten. Den zweiten Teil bildet eine Erweiterung innerhalb der bestehenden Plattform Salesforce. Die Entwicklung verwendet somit bestehende Hardwareinfrastruktur.
22	09.01.2018	Ein Projektkernteammitglied meldet, dass alle seine verantworteten Arbeitspakete funktionell fertiggestellt sind.
23	01.03.2018	Ein Projektkernteammitglied meldet, dass alle seine verantworteten Arbeitspakete funktionell fertiggestellt sind.
24	12.03.2018	Der Auftraggeber kontaktiert den Projektleiter und informiert ihn, dass aufgrund struktureller Änderungen und Änderungen an einem

ID	Datum	Inhalt
		nachgelagerten Prozess eine Änderung an der Funktionalität der Software erforderlich ist.
25	13.03.2018	<p>Der Projektleiter informiert das Projektteam über die angeforderten Änderungen. Das Projektteam erbittet sich Zeit, die Auswirkungen der Änderung zu evaluieren und den damit verbundenen Mehraufwand und die damit verbundene Verzögerung des Projektendtermins sowie die Änderung des Projektbudgets abzuschätzen.</p> <p>Anm.: Unternehmensintern erfolgt eine innerbetriebliche Leistungsverrechnung, wodurch eine Änderung am Projektaufwand eine Änderung im Projektbudget bedingt.</p>
26	22.03.2018	<p>Das Projektteam liefert die geschätzten Auswirkungen der Änderung auf den Fertigstellungstermin, den Aufwand und das Budget. Der Projektleiter fasst die Informationen in ein „Project Change Request“ Formular zusammen, welches laut Projektabwicklungsprozess für jegliche Änderung am Projektinhalt und/oder Endtermin und/oder Budget erforderlich ist.</p> <p>Das vorgefertigte „Project Change Request“ Formular wird zur formellen Freigabe an die vorgesetzte Stelle des Auftraggebers mit Budgetverantwortung gesendet.</p> <p>Die Lösungsentwicklung wird bis zur Entscheidung auf Freigabe der Änderung gestoppt, um möglicherweise unnötige Arbeit zu verhindern.</p>
27	30.03.2018	Die formale Freigabe des „Project Change Request“ ist erfolgt. Der Projektleiter passt gemeinsam mit dem Team die Planung an. Die Fertigstellung verzögert sich um vier Wochen. Die Lösungsentwicklung wird unter Betrachtung der Änderung fortgesetzt.
28	06.04.2018	Ein Projektkernteammitglied meldet, dass alle seine verantworteten Arbeitspakete funktionell fertiggestellt sind.
29	12.04.2018	Ein Projektkernteammitglied meldet, dass die angeforderten Änderungen in das bereits vorher fertiggestellte Arbeitspaket eingearbeitet wurden und somit wieder alle seine verantworteten Arbeitspakete funktionell fertiggestellt sind.
30	18.05.2018	Das Projektteam meldet, dass die Änderungen eingearbeitet sind und das letzte Arbeitspaket laut Spezifikation fertiggestellt ist.

1 *Tabelle 3: zeitlicher Ablauf der Entwicklung der Lösung*

2 Tabelle 3 zeigt den zeitlichen Ablauf der Tätigkeiten, die zwischen dem Start der Entwicklung der
 3 Lösung und der Fertigstellung der Arbeitspakete laut Spezifikation liegen.

1 Das Projektkernteam, welches aus vier Personen zuzüglich Projektleiter besteht, hat in der
2 durchlaufenen Zeitspanne nicht als Team, sondern als Zusammenschluss mehrerer
3 abgeschotteter Projektsubteams gearbeitet. Aufgrund dieser Tatsache sind die unterschiedlichen
4 Arbeitspaketverantwortlichen zu unterschiedlichen Zeiten mit ihren Arbeitspaketen fertig
5 geworden. Die Zeitspanne zwischen der Fertigstellung aller Arbeitspakete des ersten
6 Projektkernteammitglieds und der Fertigstellung aller Arbeitspakete des letzten
7 Projektkernteammitglieds beträgt über vier Monate. Somit ergeben sich zumindest vier Monate,
8 in denen das gesamte Projektkernteam nicht geschlossen an der Lösungsentwicklung gearbeitet
9 hat.

10 Ein Übergang zwischen zwei Arbeitsschritten sticht durch einen hohen Zeitbedarf hervor. Die
11 Gründe werden nachfolgend kurz beschrieben.

12 Übergang Kommunikation des Bedarfs an einer Projektänderung (ID 24) zur Freigabe der
13 Projektänderung (ID 27): Die 18 Tage, die zwischen der Kommunikation des Bedarfs an das
14 Team und der Freigabe für die Umsetzung der Änderung liegen, sind ohne Entwicklungsfortschritt
15 vergangen. Dies liegt darin begründet, dass Ressourcen für die Evaluierung der Änderung
16 erforderlich waren und die Weiterführung der Lösungsentwicklung bis zur Freigabe der Änderung
17 gestoppt wurde, um Arbeit zu verhindern, die aufgrund der angeforderten Änderung potenziell
18 obsolet sein könnte.

19 **2.3 Test und Inbetriebnahme**

20 Der Bereich Test und Inbetriebnahme gliedert sich in drei Abschnitte: Die Vorbereitung des
21 Deployments, den Überblick über den Change Management Prozess und die Überführung in das
22 Produktivsystem. Nach dem erfolgreichen Durchlaufen dieses Bereichs sind die Vertreter des
23 Auftraggebers in der Lage, die Lösung zu verwenden und das Unternehmen kann erstmals von
24 den Verbesserungen, die diese Lösung mit sich bringt, profitieren. Nachfolgend sind diese
25 Abschnitte näher beschrieben.

26 **2.3.1 Vorbereitung des Deployments**

27 Die Vorbereitung des Deployments umfasst alle Tätigkeiten, die zwischen der Fertigstellung der
28 einzelnen Arbeitspakete und der Durchführung des Change Prozesses liegen. Das beinhaltet die
29 Integration der einzelnen Arbeitspakete und umfasst Deployments und das Ausführen von
30 Testfällen. Sowohl das Ausführen von Testfällen als auch die dazu nötigen Deployments erfolgen
31 im Test- und Qualitätssystem. Der Bereich beinhaltet zudem die inhaltliche Abnahme des
32 Auftraggebers. Die Vorbereitung des Deployments endet mit der Initiierung des Change
33 Prozesses.

34 Tabelle 4 zeigt eine Auflistung der inhaltlichen und zeitlichen Abläufe.

ID	Datum	Inhalt
31	23.05.2018	<p>Das Projektkernteam trifft sich, um die einzelnen Arbeitspakete zusammen zu führen und die Stub-Objekte gegen Schnittstellen zu tauschen, die auf die Testumgebung verweisen.</p> <p>Anm.: Bei Stub-Objekten handelt es sich nach Westphal (2006) um Platzhalter, die die Funktionalität der Informationsquelle simulieren, um voneinander entkoppelte Entwicklungen zu ermöglichen.</p>
32	25.05.2018	<p>Das Projektkernteam meldet, dass bei der Integration einzelner Softwarepakete Probleme auftreten. Eine genannte Ursache sind nicht fertig implementierte Schnittstellen und unterschiedliche Ansichten über die Verantwortlichkeit über diese Schnittstellen. Zudem meldet das Projektteam, dass es für benötigte Drittsysteme (beispielsweise E-Mail) kein Testsystem gäbe und somit entweder mit dem Produktivsystem getestet werden müsste, oder auf Tests dieser Schnittstelle verzichtet werden müsse. Ein Termin zur gemeinsamen Problemanalyse und Variantenbildung zur Problembhebung wird vereinbart.</p> <p>Anm.: Das beobachtete Unternehmen arbeitet mit einer drei Stufen-Architektur, die ein Testsystem, ein Qualitätssystem und ein Produktivsystem vorsieht. Die Systeme sind strukturell ähnlich aufgebaut, können aber aufgrund der Größe des Produktivsystems nicht das gesamte Spektrum bereitstellen.</p>
33	28.05.2018	<p>Der Termin findet statt und es wird entschieden, dass die Testungen in Bezug auf den Dienst E-Mail mit dem Produktivsystem geschehen, unter vorheriger genauester Testdokumentation unter Zuhilfenahme eines Stub-Objekts. Des Weiteren werden die Verantwortlichkeiten für die nicht fertig entwickelten Schnittstellen geklärt und ein Projektkernteammitglied mit der Fertigstellung der Implementierung beauftragt. Aufgrund des geringen erwarteten Aufwands zur Fertigstellung der Implementierung, wird als Fertigstellungsdatum der 08.06.2018 genannt.</p>
34	11.06.2018	<p>Die fehlenden Schnittstellen wurden zeitgerecht am 08.06.2018 fertiggestellt. Das Projektteam beginnt erneut seine einzelnen Arbeitspakete im Testsystem zu integrieren. Um bei späteren Deployments nichts zu vergessen, wird eine Deploymentcheckliste geführt.</p>
35	15.06.2018	<p>Nach kleineren Nacharbeiten und beheben von gefundenen Fehlern ist die Integration fertiggestellt und es existiert ein System im testbereiten Zustand. Der Projektleiter gibt das Ausführen der Testfälle in Auftrag.</p>
36	20.06.2018	<p>Die Testfälle, die beim Requirements Engineering mit dem Auftraggeber dokumentiert wurden, sind ausgeführt. Mit zwei Ausnahmen sind alle Tests auf Anhieb positiv ausgefallen. Die Behebung dieser Probleme konnte mit</p>

ID	Datum	Inhalt
		wenig Aufwand parallel zu weiteren Tests erfolgen. Am Ende des Tages konnten alle Tests positiv absolviert werden. Der Projektleiter bittet das Team ein Deployment auf das Qualitätssystem durchzuführen.
37	26.06.2018	Das Deployment auf das Qualitätssystem ist fertiggestellt. Der Projektleiter gibt erneut das Ausführen der Testfälle in Auftrag.
38	29.06.2018	Die Testfälle wurden erfolgreich ausgeführt, der Projektleiter demonstriert das vorläufige Ergebnis dem Auftraggeber und bittet diesen, seine Tests durchzuführen.
39	03.07.2018	Der Ansprechpartner der Auftraggeberseite gibt an, dass ein Testfall nicht positiv verläuft. Der Testfall ist nicht explizit dokumentiert und es existiert keine konkrete Anforderung, die auf diesen Testfall abzielt. Der ungeduldige Auftraggeber merkt an, dass diese Anforderung als selbstverständlich anzusehen ist, weil in deren Abteilung seit Jahren so gearbeitet wird. Da die Entwicklungsaufwände zur positiven Abwicklung des Testfalls im einstelligen Stundenbereich liegen, einigen sich Projektleiter und Auftraggeber auf die Abhandlung des Ganzen als Project Change. Sie bereiten den Project Change Request vor und senden ihn zur formellen Freigabe an die vorgesetzte Stelle des Auftraggebers mit Budgetverantwortung (Manager des Auftraggebers).
40	04.07.2018	Spürbar unzufrieden meldet sich der Manager des Auftraggebers beim Projektleiter und bittet um Stellungnahme. Er kann nicht nachvollziehen, weshalb eine Änderung mit einem Entwicklungsaufwand von 8 Stunden, eine Verzögerung von circa drei Wochen sowie einen Mehraufwand von vier Personentagen erfordert. Nach transparenter Aufschlüsselung der zu erwartenden Aufwände (Entwicklungsaufwand, Integrationsaufwand, zweifacher Deploymentaufwand, zweifacher Testaufwand) entschließt sich der Manager des Auftraggebers widerwillig zur Freigabe des Change Requests.
41	20.07.2018	Die Adaptierung der Testfälle, die Implementierung, sowie das Deployment auf Test- und Qualitätssystem inklusive erfolgreicher Durchführung der Testfälle ist abgeschlossen. Der Auftraggeber hat seine eigenen Testfälle ausgeführt und ist mit dem Ergebnis zufrieden. Er gibt die inhaltliche Freigabe zur Überführung in das Produktivsystem.
42	20.07.2018	Der Projektleiter informiert das Team über die positive Nachricht und initiiert einen Change Management Prozess, der für jede Änderung an der IT Infrastruktur erforderlich ist. Die Vorbereitung des Deployments ist dadurch abgeschlossen.

1 Tabelle 4 veranschaulicht den zeitlichen Ablauf der Tätigkeiten, die zwischen der Fertigstellung
2 der spezifizierten Arbeitspakete und des Starts des Change Management Prozesses liegen. Drei
3 Problemfelder, deren Durchführung eine hohe Zeitspanne erforderten, werden inklusive deren
4 Ursachen nachfolgend beschrieben.

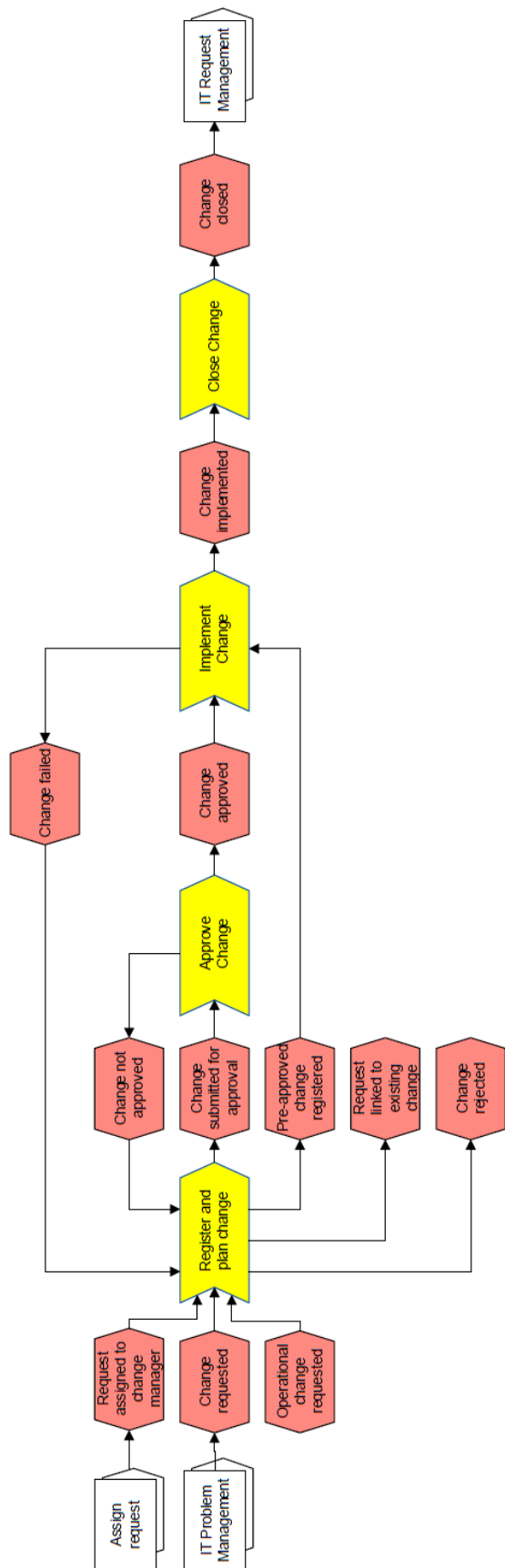
5 Übergang Start der Integration (ID 31) zu Fertigstellung der Integration (ID 35): Der Zeitraum in
6 dem fertiggestellte Arbeitspakete integriert und in einen testfähigen Zustand gebracht werden
7 beträgt 23 Tage. Über 60% dieser Zeitspanne wird für das Nachliefern von Schnittstellen benötigt,
8 die aufgrund von Kommunikationsproblemen nicht innerhalb der Entwicklungsphase
9 bereitgestellt wurden.

10 Übergang Start des Deployments von Test- auf Qualitätsumgebung (ID 36) zu Fertigstellung der
11 Testung im Qualitätssystem (ID 38): Die Ursache für die Dauer von 9 Tagen liegt in der manuellen
12 Ausführung der Tests und des Deployments begründet.

13 Übergang Start der Benutzerakzeptanztests (ID 39) zu inhaltlicher Freigabe des Kunden (ID 41):
14 Zwischen dem Start der Tests durch den Kunden und der inhaltlichen Freigabe der Lösung
15 vergehen 17 Tage. Durch die späte Integration des Kunden und den bürokratisch intensiven
16 Umgang mit Änderungswünschen entsteht ein nicht einkalkulierter Aufwand, der eine
17 Verlängerung der Durchlaufzeit nach sich zieht.

18 **2.3.2 Überblick Change Management Prozess**

19 Die Initiierung des Change Management Prozesses bildet die Grenze zwischen den Abschnitten
20 „Vorbereitung des Deployments“ und „Überführung in die Produktivumgebung“. Dieser Abschnitt
21 dient zur Verdeutlichung und zum näheren Verständnis des Prozesses und beschreibt diesen
22 nachfolgend. Der Change Management Prozess ist ein mehrstufiger Prozess, der
23 unterschiedliche Startpunkte, unterschiedliche Verläufe und unterschiedliche Abzweigungen
24 vorsieht. Die grobe Ausrichtung des Prozesses ist in der Abbildung 2: Überblick Change
25 Management Prozess dargestellt.



1

2 *Abbildung 2: Überblick Change Management Prozess*

1 Die Abbildung 2 zeigt, dass der Change Management Prozess unterschiedliche Startpunkte
2 haben kann und auf Basis der gegebenen Rahmenbedingungen unterschiedliche Abzweigungen
3 verfolgt. Die nachfolgenden Beschreibungen fokussieren den Anwendungsfall eines Software
4 Deployments und betrachten die möglichen anderen Startpunkte, die möglichen Wege bei
5 negativer Freigabe und/oder eines fehlgeschlagenen Deployments nicht gesondert.

6 Der Prozess gliedert sich in die vier Haupttätigkeiten „Register and plan change“, „Approve
7 Change“, „Implement Change“ und „Close Change“, die nachfolgend beschrieben werden:

8 **Register and plan change:** In der Tätigkeit „Register and plan change“ werden Change
9 Anforderungen, auch Request for Change (RfC) genannt, einem von fünf Change Typen
10 zugeordnet und detailliert beschrieben. Dies beinhaltet die Darlegung der vom Change
11 betroffenen Services und Systeme (Configuration Items). Des Weiteren sieht dieser Schritt,
12 sofern es sich um einen „Standard Change“ handelt, eine Risikoanalyse, die Planung und
13 Dokumentation eines Roll-Back Szenarios, sowie die genaue Dokumentation der
14 auszuführenden Schritte vor. Die angesprochenen fünf Change Typen sind nachfolgend
15 überblicksmäßig beschrieben:

- 16 • **Normal Change:** Als Normal Change wird jegliche Änderung an IT Services, IT Systemen
17 und/oder der IT Infrastruktur angesehen, die eine Änderung der Funktionalität eines
18 Service hat, die Verfügbarkeit des Service (temporär) einschränkt, oder ein Risiko für die
19 negative Beeinträchtigung des Service birgt. Der Normal Change bildet den Standardfall
20 ab und sieht vor, dass der gesamte Change Prozess durchlaufen wird.
- 21 • **Pre-approved Change:** Pre-approved Changes sind vorab definierte und genehmigte
22 Änderungen an IT Services, IT Systemen und/oder der IT Infrastruktur, die ein geringes
23 Risiko nach sich ziehen und für die eine Arbeitsanweisung oder standardisierte Methode
24 für die Umsetzung zur Verfügung steht. Pre-approved Changes erfordern ein verringertes
25 Set an Informationen, die im Schritt „register and plan change“ dokumentiert werden
26 müssen. Es wird auf die Risikoanalyse, die Planung und Dokumentation eines Roll-Back
27 Szenarios, sowie die genaue Dokumentation der auszuführenden Schritte verzichtet. Pre-
28 Approved Changes überspringen zudem die explizite Change Freigabe (Schritt „Approve
29 Change“).
- 30 • **Emergency Change:** Ein Emergency Change ist ein Change mit höchster Priorität in der
31 Organisation. Emergency Changes sind als Changes definiert, die in kürzester Zeit
32 evaluiert, geprüft und entweder genehmigt oder abgelehnt werden. Dieser Typ eines
33 Changes wird für Änderungen an IT Services, IT Systemen und/oder der IT Infrastruktur
34 verwendet, die dazu dienen, größeren Schaden abzuwenden.
- 35 • **SAP Change:** Alle Änderungen, die das System SAP betreffen, werden in einem
36 gesondert beschriebenen Prozess abgewickelt.
- 37 • **Firewall Changes:** Änderungen an den Firewall Einstellungen werden in Form von
38 Service Requests abgehandelt und sind nach den Vorgaben einer gesonderten
39 Verfahrensanweisung abzuarbeiten und zu dokumentieren.

1 Ist eine entsprechende Zuordnung erfolgt und die Beschreibung der geforderten Punkte adäquat,
2 wird an den nachgelagerten Hauptschritt „Approve Change“ übergeben.

3 **Approve Change:** Um eine Freigabe für einen Change zu bekommen, ist die Zustimmung aller
4 Verantwortlichen aller vom Change betroffenen Systeme (Configuration Items Owner)
5 erforderlich. Um diese Freigabe zu erwirken, muss die Dokumentation im vorgelagerten Schritt
6 detailliert genug sein, um alle Verantwortlichen mit den für eine Entscheidung notwendigen
7 Informationen zu versorgen. Die letztverantwortliche Freigabe übernimmt der Change Manager,
8 der im Regelfall vom Service Owner des betroffenen Service übernommen wird. Der Change
9 Manager stellt vor der Freigabe sicher, dass Maßnahmen getroffen wurden, um die Gefahr eines
10 Fehlschlagens des Changes zu minimieren.

11 **Implement Change:** Nach erfolgreicher Freigabe des Changes werden die im RfC
12 dokumentierten Schritte zur Umsetzung des Changes in die Wege geleitet und dokumentiert.

13 **Close Change:** Sobald der Change erfolgreich implementiert wurde, werden passende Tests
14 durchgeführt und deren Ergebnisse dokumentiert. Abschließend erfolgt die Anpassung der
15 Configuration Management Database (CMDB). Die CMDB ist eine Auflistung aller für den Service
16 notwendigen Configuration Items und deren Beziehung zueinander. Ein Configuration Item kann
17 unter anderem ein IT System, ein Mitarbeiter, eine Dokumentation, eine Infrastrukturkomponente
18 oder ein anderer Service sein.

19 In Abbildung 2 ist „IT Request Management“ als nachfolgender Schritt zu „Close Change“
20 gekennzeichnet. Dies soll verdeutlichen, dass der Change Management Prozess eine
21 Prozesskaskade von IT Request Management darstellen kann und somit mit IT Request
22 Management in einer internen Kunden-Lieferanten-Beziehung stehen kann. Dies setzt voraus,
23 dass der Change Management Prozess durch den IT Request Management Prozess gestartet
24 wird. Im vorliegenden Szenario wird der Change Management Prozess nicht vom Prozess „IT
25 Request Management“ initiiert, wodurch auch keine Übergabe der Informationen des Schritts
26 „Close Change“ erforderlich sind.

27 An der Beschreibung des Change Management Prozesses ist gut erkennbar, dass das
28 Durchlaufen eines Normal Changes aufgrund des manuellen Freigabeprozesses eine Vielzahl
29 personeller Ressourcen bindet und dadurch erheblichen Einfluss auf die Dauer eines
30 Deployments in die Produktivumgebung nimmt.

31 **2.3.3 Überführung in die Produktivumgebung**

32 Die Überführung in die Produktivumgebung startet mit dem Durchlaufen des in Kapitel 2.3.2
33 skizzierten Change Management Prozesses und endet mit der Nutzung der Software durch die
34 Vertreter des Auftraggebers. Die Tabelle 5 beinhaltet eine Auflistung der inhaltlichen und
35 zeitlichen Abläufe.

ID	Datum	Inhalt
43	20.07.2018	Der Projektleiter startet mit der Hauptaktivität „register and plan change“ des Change Management Prozesses und wählt als Change Typ „Normal

ID	Datum	Inhalt
		Change“ aus. Aufgrund der weitreichenden Änderungen, die diese Softwarelösung nach sich zieht, werden als betroffene Services neben der Hauptplattform „SharePoint Intranet Services“ (nachfolgend SharePoint genannt) auch „Salesforce CRM“ (nachfolgend Salesforce genannt) identifiziert. Unter Betrachtung der je Service vorhandenen Configuration Management Database (CMDB) startet der Projektleiter die detaillierte Beschreibung der Änderungen, die das geplante Deployment nach sich ziehen wird und fasst diese in einem Dokument zusammen.
44	24.07.2018	Gemeinsam mit dem Projektkernteam wird die geforderte Risikoanalyse durchgeführt und ein Ablaufplan des Deployments dokumentiert. Zusätzlich wird ein Roll-Back Plan erstellt, der ein geordnetes Rückgängigmachen des Changes bei Komplikationen ermöglicht und ebenfalls in das Dokument aufgenommen.
45	27.07.2018	Das vorbereitete Dokument wird an die beiden Serviceverantwortlichen (Service Owner) der Services SharePoint und Salesforce gesendet. Hierdurch geht der Prozess zum Prozessschritt „approve change“ über.
46	01.08.2018	Der Service Owner des Service SharePoint erteilt seine Freigabe.
47	02.08.2018	Der Service Owner des Service Salesforce fordert eine detailliertere Beschreibung an. Die Beschreibung soll um die zu erwartenden Auswirkungen für Mitarbeiter erweitert werden, die den Service Salesforce ohne aktive VPN Verbindung im Kontext des zu verändernden Angebotsfreigabeprozesses konsumieren.
48	06.08.2018	Der Projektleiter ergänzt die fehlenden Informationen.
49	07.08.2018	Der Service Owner des Service Salesforce erteilt seine Freigabe. Der Prozessschritt „approve change“ ist somit abgeschlossen. Um nach dem Deployment keine Zeit zu verlieren, beruft der Projektleiter ein Meeting zur Übergabe der Lösung an den IT Applikationssupport (Handover Meeting) für den 13.08.2018 (geplanter Fertigstellungstag) ein.
50	11.08.2018	Das Projektteam startet das Deployment und beginnt die bestehenden Schnittstellen, die aktuell auf Ziele im Qualitätssystem zeigen, durch Ziele im Produktivsystem zu ersetzen. Diese Tätigkeiten bilden den Kern des Prozessschritts „implement change“.
51	12.08.2018	Das Projektteam finalisiert die Integration, wodurch der Prozessschritt „implement change“ beendet wird. Direkt im Anschluss daran startet der Prozessschritt „close change“, welcher das manuelle Ausführen von vorab definierten Funktionstests beinhaltet. Die Fertigstellungsinformationen werden gemeinsam mit den Testberichten an die beiden Service Owner geschickt.

ID	Datum	Inhalt
52	13.08.2018	Die Service Owner bestätigen den erfolgreichen Ablauf des Changes und veranlassen die Aktualisierung ihrer CMDB auf Basis der zur Verfügung gestellten Informationen. Der Projektleiter wird über den positiven Abschluss des Prozesses informiert, wodurch der Change Management Prozess abgeschlossen ist. Der Projektleiter bittet den Ansprechpartner des Kunden, das Deployment zu überprüfen und das Projekt abzunehmen.
53	13.08.2018	Der Projektleiter hält das geplante Handover Meeting ab, um die Lösung an den IT Applikationsbetrieb zu übergeben. Ziel des Termins ist der Wissenstransfer über den Aufbau und die Funktionsweise der Lösung sowie die Dokumentation von zu erwartenden Fehlern und deren Lösungswege. Anm.: Die Abteilung IT Applikationsbetrieb übernimmt den Second Level Support für eigens entwickelte Applikationen innerhalb des Unternehmens.
54	14.08.2018	Der Ansprechpartner des Kunden bestätigt das positive Deployment. Die durch die Auftraggeberseite geschulten Mitarbeiter verwenden den angepassten Angebotsfreigabeprozess produktiv.
55	16.08.2018	Der Ansprechpartner des Kunden kontaktiert den Projektleiter, da von Seiten des IT Applikationsbetriebs widersprüchliche und teilweise falsche Informationen an die Benutzer der Lösung weitergegeben wurden.
56	17.08.2018	Mitarbeiter des IT Applikationsbetriebs melden sich beim Projektleiter, da sie mehrere Anfragen von Benutzern des neuen Angebotsfreigabeprozess bekommen haben, die nicht beantwortet werden können. Der Projektleiter erörtert die genaue Funktionsweise der Lösung, um ein tieferes Verständnis bei den Supportmitarbeitern zu schaffen, wodurch auch die Beantwortung der Benutzeranfragen kein Problem mehr darstellt.
57	17.08.2018	Der Ansprechpartner des Kunden führt gemeinsam mit dem Projektleiter die formelle Abnahme des Projekts durch.

1 *Tabelle 5: zeitlicher Ablauf der Überführung in die Produktivumgebung*

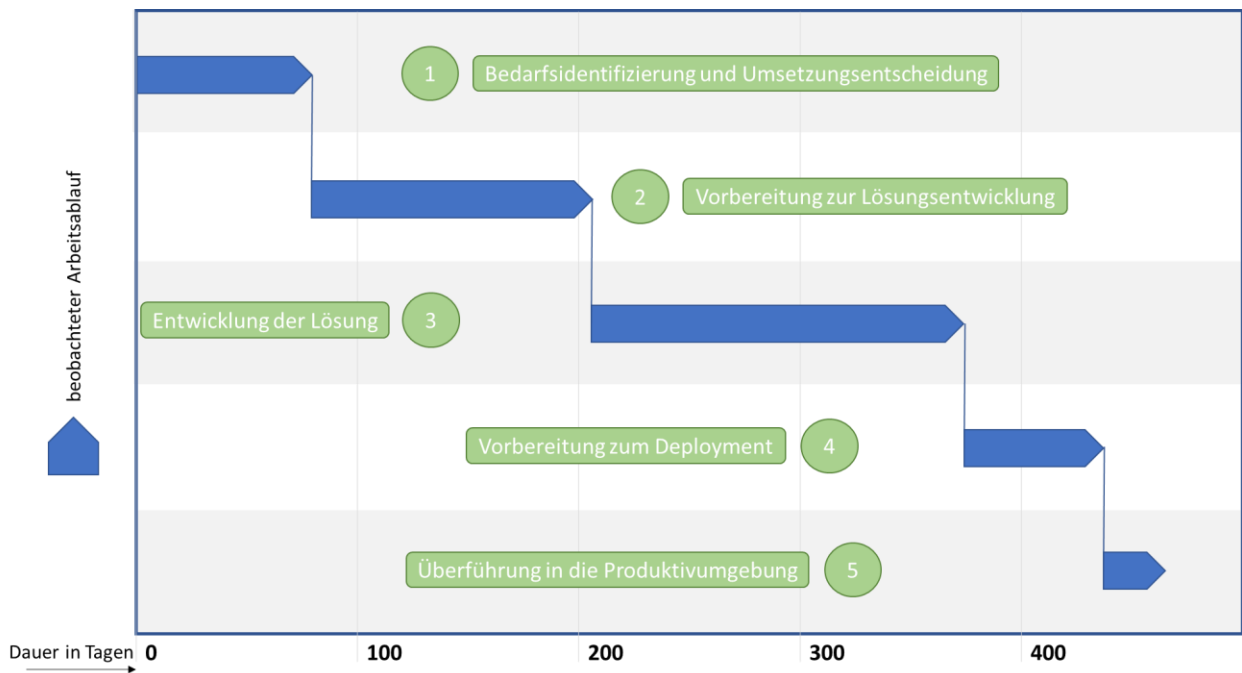
2 Tabelle 5 zeigt den zeitlichen Ablauf der Tätigkeiten, die zwischen dem Start des Change
3 Management Prozesses und der formalen Abnahme des Projekts liegen. Trotz der Ausführung
4 des Deployments ohne Komplikationen sind 25 Tage zwischen dem Start des Change
5 Management Prozesses und der produktiven Nutzung der Lösung vergangen. Nachfolgend wird
6 der zeitintensivste Vorgang beschrieben und auf die Problematik bei der Übergabe an den IT
7 Applikationssupport eingegangen.

8 Übergang Start des Change Management (ID 43) Prozess zur Freigabe des Changes (ID 49):
9 Die Zeitspanne, die zwischen dem Start des Change Management Prozesses und der Freigabe
10 des Changes vergeht, beträgt 18 Tage. Das ist darauf zurück zu führen, dass die Tätigkeiten, die
11 die Freigabeentscheidung eines Changes vorbereiten, manuell durchgeführt werden. Eine

- 1 weitere Ursache besteht darin, dass durch den Projektleiter, das vierköpfige Projektkernteam und
- 2 zwei Service Owner insgesamt sieben Personen in dieser Zeitspanne involviert sind.
- 3 Übergang Durchführung des Handover Meetings (ID 53) zur Befähigung des IT-
- 4 Applikationsbetriebs Benutzeranfragen zu beantworten (ID 57): Zwischen diesen beiden
- 5 Ereignissen liegen 4 Tage. Während dieser Zeitspanne wurden widersprüchliche und teilweise
- 6 falsche Informationen an die Benutzer der Lösung kommuniziert. Das ist darauf zurückzuführen,
- 7 dass das Verständnis der Supportmitarbeiter für die Funktionsweise der Lösung nicht weit genug
- 8 ausgeprägt war. Eine mögliche Ursache dafür ist eine zu späte Integration des IT
- 9 Applikationssupports und ein ineffektiver Ablauf des Handover Meetings.

10 2.4 Kurzzusammenfassung

- 11 Der Ablauf der Leistungserbringung gliedert sich in 5 Abschnitte, die sich den drei Phasen
- 12 Bedarfsidentifizierung und Lösungsentscheidung, Lösungsentwicklung sowie Test und
- 13 Deployment zuordnen lassen. Die nachfolgende Abbildung illustriert den Verlauf zeitlichen Lauf
- 14 der beobachteten Leistungserbringung.



15
16 *Abbildung 3: zeitlicher Verlauf der beobachteten Leistungserbringung*

- 17 Der zeitliche Verlauf in Abbildung 3 zeigt welchen Anteil die fünf Abschnitte an der Gesamtdauer
- 18 haben. Es ist klar ersichtlich, dass die Phase Lösungsentwicklung (Abschnitt 2 und 3) den größten
- 19 Einfluss auf die Gesamtdauer nimmt. In der Abbildung geht hervor, dass das Verhältnis zwischen
- 20 Abschnitt 2, Vorbereitung zur Lösungsentwicklung, und Abschnitt 3, Lösungsentwicklung,
- 21 Potential zur Verbesserung bietet. Die Phase Test und Inbetriebnahme (Abschnitt 4 und 5)

1 **3 IT SERVICE MANAGEMENT NACH ITIL**

2 Dieses Theoriekapitel enthält Grundinformationen zum IT Service Management. Aufgrund der
3 weiten Verbreitung hat sich die IT Infrastructure Library (ITIL) als defacto Standard für IT Service
4 Management etabliert und dient als Grundlage für die nähere Beschreibung von IT Service
5 Management. Dieses Kapitel befasst sich mit den ITIL Grundlagen sowie den fünf
6 Prozessgruppen, die in ITIL Verwendung finden. Prozesse der jeweiligen Prozessgruppen, die
7 für diese Arbeit nur peripher von Relevanz sind, werden zur Schaffung des Überblicks aufgeführt
8 und zusammenfassend beschrieben. Jede Prozessgruppe enthält eine Zusammenfassung,
9 welche die relevanten Punkte für diese Arbeit nochmals hervorhebt. Des Weiteren wird jeder
10 Prozess mit den Beobachtungen aus Kapitel 2 in Relation gesetzt.

11 **3.1 ITIL Grundlagen**

12 Das Great Britain: Cabinet Office (2013, S. 18) definiert ein Service als *„eine Möglichkeit einen*
13 *Mehrwert für Kunden zu erbringen, indem das Erreichen der von den Kunden angestrebten*
14 *Ergebnissen erleichtert oder gefördert wird. Dabei müssen die Kunden selbst keine*
15 *Verantwortung für bestimmte Kosten und Risiken tragen“*. Um dieser Definition gerecht zu
16 werden, bedarf es einer effizienten Verwaltung und Steuerung des Angebots. Bei derartigen
17 Angeboten im IT Umfeld spricht man von IT Service Management. Bei IT Service Management
18 handelt es sich um *„die Implementierung und das Management von qualitätsbasierten IT*
19 *Services, die den Geschäftsanforderungen gerecht werden.“* (Great Britain: Cabinet Office, 2013,
20 S. 21)

21 Die IT Infrastructure Library, kurz ITIL, ist ein britisches Best Practices Rahmenmodell für IT
22 Service Management und wurde in der aktuellsten Version vom Cabinet Office veröffentlicht. ITIL
23 bietet Leitlinien für professionelles IT Service Management und beinhaltet im Wesentlichen ein
24 Prozessmodell aus 37 Prozessen sowie Rollen- und Funktionsmodellen. ITIL in der 2011 Edition
25 bildet die letztgültige Ausgabe und beinhaltet 5 Kernpublikationen, die die einzelnen Phasen des
26 Service Lebenszyklus beschreiben. Ergänzt werden die 5 Kernpublikationen von ITIL durch
27 Complementary Publications sowie Web Support Services. Mit dem Best-Practice-Leitfaden ITIL
28 wird Unternehmen die Möglichkeit gegeben, aus den Erfahrungen anderer Firmen zu profitieren.
29 Die in ITIL beschriebenen Prozesse können mit Kennzahlen bewertet, gesteuert und auf ihre
30 Effizienz gemessen werden.

31 Die Prozesse in ITIL können in 5 Prozessgruppen gegliedert werden:

- 32 • Service Strategy
- 33 • Service Design
- 34 • Service Transition
- 35 • Service Operation
- 36 • Continual Service Improvement

- 1 Gemeinsam bilden diese, wie in Abbildung 4 gezeigt, den ITIL Service Lifecycle.



- 2
3 Abbildung 4: ITIL Service Lifecycle (Great Britain: Cabinet Office, 2013)
4 In der Abbildung 4 ist ersichtlich, dass sich der Service Lifecycle um seinen Kern (Service
5 Strategy) herum aufbaut und einen in sich geschlossenen, immer wiederholenden Kreislauf
6 bildet. Die Abbildung unterstreicht den hohen Stellenwert, den die kontinuierliche Verbesserung
7 (Continual Service Improvement) innerhalb von ITIL einnimmt.
8 Jeder Service sieht eine Person vor, die die Gesamtverantwortung für den Service trägt. Diese
9 Person wird in ITIL als Service Owner bezeichnet.

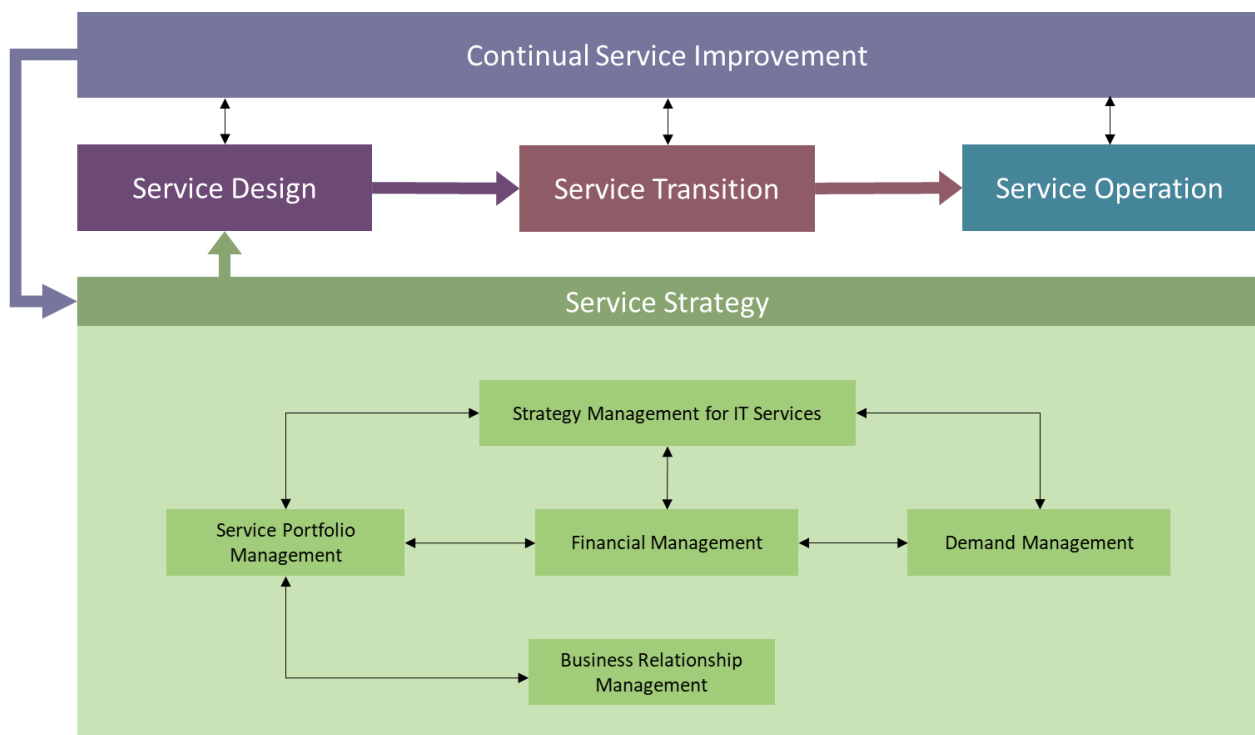
10 3.2 Service Strategy

- 11 Service Strategy bildet, wie in Abbildung 4 angedeutet, den Kern des ITIL Service Lifecycle. Die
12 Phase beschreibt den Wertbeitrag, den der angestrebte IT-Service leistet, welche Assets für die
13 Bereitstellung erforderlich sind und wie diese Assets optimal bereitgestellt werden können.
14 Ziel der Phase Service Strategy ist die Identifikation des Service und dessen Kunden, die
15 Erstellung eines klar definierten Service Bereitstellungsmodells sowie die Definition von
16 Prozessen, um die Umsetzung der erstellten Strategie zu ermöglichen. Die Phase Service
17 Strategy besteht zum einen aus der Definition der Strategie, mittels derer der Service Provider
18 den Service bereitstellt und zum anderen aus der Definition der Strategie, wie dieser Service
19 verwaltet wird. Neben der Erfüllung von markt- und kundengetriebenen Anforderungen

1 beschäftigt sich die Phase Service Strategy mit der Frage, wodurch sich der angebotene Service
2 von Services anderer Dienstleister abhebt.

3 Die Phase Service Strategy umfasst folgende Hauptprozesse: (Great Britain: Cabinet Office,
4 2013)

- 5 • Strategy Management for IT Services
- 6 • Service Portfolio Management
- 7 • Financial Management for IT Services
- 8 • Demand Management
- 9 • Business Relationship Management



10

11 *Abbildung 5: Zusammenhang zwischen den Prozessen in Service Strategy (Great Britain: Cabinet Office, 2013)*

12 Die Abbildung 5 zeigt den engen Zusammenhang den Services untereinander, sowie die
13 Schnittstelle zum Kunden des Service (Service Design). Diese werden in den nachfolgenden
14 Unterkapiteln näher beschrieben.

15 **3.2.1 Strategy Management for IT Services**

16 Strategy Management for IT Services ist für die Definition und das Management der
17 Servicestrategie verantwortlich. Der Prozess stellt sicher, dass die Strategie umgesetzt wird.
18 Hierzu werden drei Hauptaktivitäten durchgeführt. In der Aktivität „strategisches Assessment“ soll
19 Klarheit über die zu erreichenden Ziele geschaffen werden und zudem Kenntnis über den eigenen
20 Standort sowie die eigene Leistungsfähigkeit erlangt werden. Aufbauend auf das strategische
21 Assessment wird in der Aktivität „Erstellung der Servicestrategie“ ein Set an langfristigen

1 übergeordneten Zielen festgelegt. Die dritte Hauptaktivität bildet die „Strategieausführung“.
2 (Great Britain: Cabinet Office, 2013)

3 Zudem wird definiert welche Kundengruppen und Kundensegmente bestehen und mit welchen
4 Services diese bedient werden sollen.

5 Der Prozess Strategy Management for IT Services definiert die Rolle der IT Steering Group,
6 welche aus Mitgliedern des IT- und Business-Managements besteht. Die IT Steering Group gibt
7 die Richtung für die Erarbeitung und Weiterentwicklung des Strategy Managements vor. Die
8 zweite Rolle im Strategy Management for IT Services beschreibt den Service Strategy Manager.
9 Der Inhaber dieser Rolle ist für die Umsetzung und Implementierung der Servicestrategie
10 verantwortlich und kümmert sich ebenso um die weitere Kommunikation der Strategie. Er
11 unterstützt die IT Steering Group bei der Weiterentwicklung einer bestehenden Strategie oder der
12 Erarbeitung neuer Servicestrategien.

13 **3.2.2 Service Portfolio Management**

14 Das Service Portfolio definiert sich in der Gesamtheit aller Services, die von einem Service
15 Provider gemanagt werden. Es wird für das Management des gesamten Lebenszyklus aller
16 Services genutzt. Das Service Portfolio kann in drei Bereiche unterteilt werden:

- 17 • Servicepipeline (geplante Services, die sich in der Entwicklung befinden)
- 18 • Servicekatalog (aktive Services, die aktuell von Kunden bezogen werden können)
- 19 • stillgelegte Services (inaktive Services, die stillgelegt oder aus dem Servicekatalog
20 entfernt wurden).

21 Im Service Portfolio Management wird nach den Ausführungen des Great Britain Cabinet Office
22 (2013) entschieden, welche Services in den Servicekatalog aufgenommen und welche aus dem
23 Portfolio gestrichen werden. Um diese Entscheidungen treffen zu können, werden Analysen zu
24 möglichen Erträgen und akzeptablem Risiko im Zuge eines Business Case durchgeführt.

25 Ein Business Case wird verwendet, um die wirtschaftliche Vorteilhaftigkeit eines Vorhabens zu
26 evaluieren und die strategische Konformität sicherzustellen. (Brugger, 2009)

27 Es wird ebenso entschieden, welche Service Ideen aus der Servicepipeline in die Phase Service
28 Design übergeleitet werden, um in weiterer Folge in den Servicekatalog aufgenommen zu
29 werden. Die Überleitung geschieht in Form eines Service Charters. Ein Service Charter spiegelt
30 das Grobkonzept des Service wider und skizziert den neuen oder geänderten Service. Er enthält
31 Informationen zu den benötigten Ressourcen und dem Umsetzungszeitraum.

32 Die Strategie für die Bereitstellung von Services wird vom Service Portfolio Manager entworfen.
33 In Zusammenarbeit mit der IT Steering Group wird ein Konzept für die Weiterentwicklung der
34 Serviceangebote erstellt und die Kompetenzen des Service Providers erarbeitet.

35 Teile der beschriebenen Agenden des Service Portfolio Managements werden im beobachteten
36 Ablauf zur Leistungserbringung unter Kapitel 2 in den IDs 6 und 7 abgedeckt.

1 **3.2.3 Financial Management for IT Services**

2 Im Financial Management werden IT Services samt deren zugrundeliegenden benötigten
3 Ressourcen finanziell bewertet und ein Wert für den Endkunden ermittelt. Das Financial
4 Management nach ITIL (2013) übernimmt die Verantwortung für die Budgetierung und das
5 Accounting sowie die Leistungsverrechnung. Das Financial Management liefert die
6 wirtschaftlichen Entscheidungsgrundlagen für das Service Portfolio Management.

7 Letzteres spiegelt sich im beobachteten Arbeitsablauf unter Kapitel 2 in ID 11 im Bereich der
8 Erstellung des Business Case wider.

9 **3.2.4 Demand Management**

10 Das Demand Management nach ITIL (2013) soll den IT Service Provider dabei unterstützen, ein
11 Verständnis für den Bedarf des Kunden zu entwickeln. Hauptziel des Demand Management
12 Prozesses ist die Beeinflussung und Steuerung des Bedarfs, um das Risiko für erhöhte Kosten
13 durch übermäßige Kapazitätsvorhaltung zu minimieren.

14 Im Kontext des im Kapitel 2 beobachteten Szenarios ist kein Demand Management Prozess
15 vorgesehen.

16 **3.2.5 Business Relationship Management**

17 Dieser Prozess hat die Etablierung und Pflege der Geschäftsbeziehung zwischen Service
18 Providern und dem (potenziellen) Kunden des Service zum Ziel. Das in ITIL (2013) beschriebene
19 Business Relationship Management beschäftigt sich außerdem mit der Identifikation sowie der
20 Dokumentation von Bedürfnissen bestehender und potenzieller Kunden des Service. Der
21 Prozessverantwortliche, der Business Relationship Manager, hat sich zum Ziel gesetzt, die
22 Kundenzufriedenheit auf einem hohen Level anzusiedeln und stellt eine Feedback- sowie
23 Eskalationsstelle für den Kunden dar.

24 Ein explizites Business Relationship Management ist in den beobachteten Arbeitsabläufen aus
25 Kapitel 2 nicht enthalten.

26 **3.2.6 Kurzzusammenfassung**

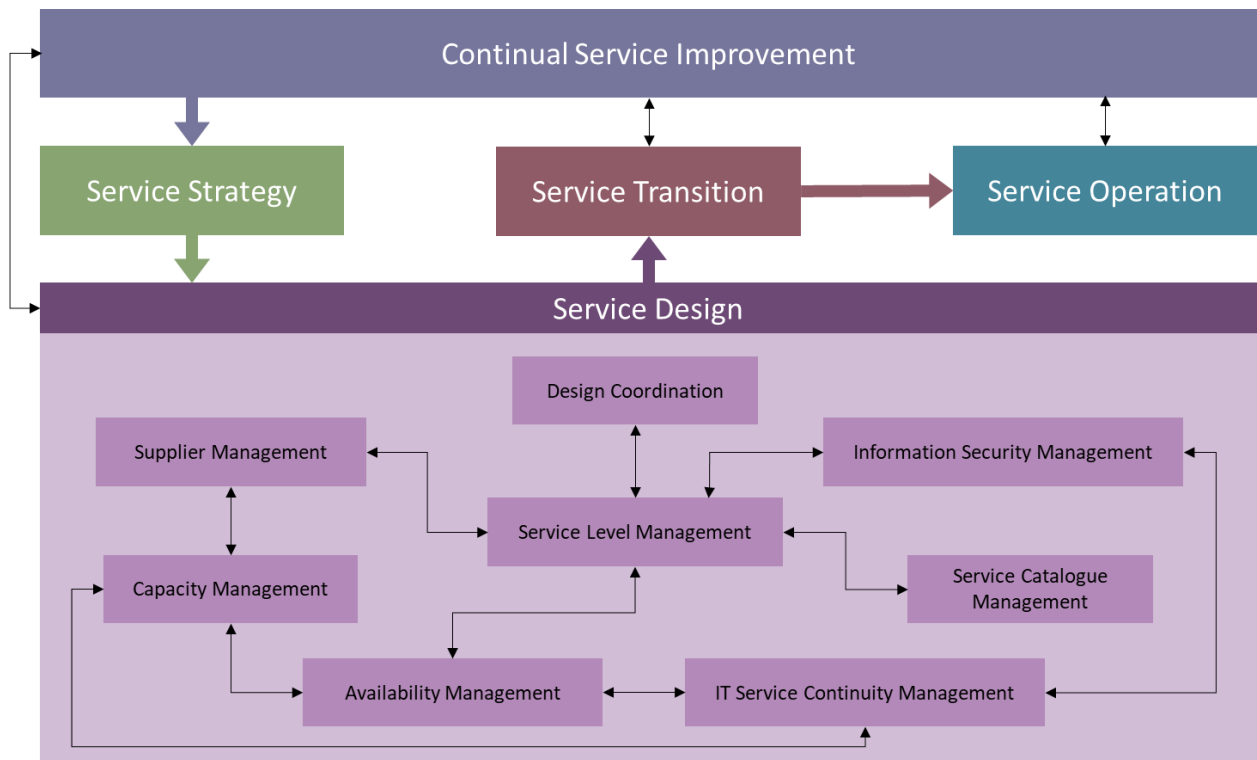
27 Die Phase Service Strategy kümmert sich um die strategisch wichtigen Bereiche und um das
28 Gesamtangebot an Services, welche den Kunden zur Verfügung gestellt werden. Services im
29 Servicekatalog richten sich nach dem Bedarf und den Wünschen der Kunden und werden unter
30 Betrachtung der wirtschaftlichen Rentabilität und im Einklang mit der unternehmerischen
31 Gesamtstrategie entwickelt. Wichtige Aspekte der Phase Service Strategy werden in den IDs 1
32 bis 11, der in Kapitel 2.1 angeführten Tätigkeiten, abgebildet.

1 **3.3 Service Design**

2 In der Phase Service Design werden geeignete IT-Services inklusive dafür benötigter Prozesse,
3 Richtlinien sowie Architektur und Dokumentation entwickelt. Diese Entwicklung basiert auf der
4 vorangegangenen strategischen Ausrichtung des Service sowie dessen Budgetierung. Der neu
5 entwickelte Service erfüllt die gegenwärtigen, bestenfalls auch zukünftigen Anforderungen der
6 Kunden. Neben dem Entwurf von neuen Lösungen fallen ebenso die Änderungen und
7 Verbesserungen bestehender Services und Lösungen in die Phase Service Design.

8 Abbildung 4: ITIL Service Lifecycle zeigt die Eingliederung der Phase Service Design in den ITIL
9 Service Lifecycle und veranschaulicht, dass Service Design einerseits auf Feedback aus der
10 Phase Continuous Service Improvement und andererseits auf den Ergebnissen der Phase
11 Service Strategy aufbaut. Beim Durchlaufen der Phase Service Design werden Service Design
12 Packages erzeugt, die als Eingangsparameter für die nachgelagerte Phase Service Transition
13 dienen. Als Disziplin, der die meisten Prozesse zuzuordnen sind, umfasst sie folgende
14 Hauptprozesse:

- 15 • Design Coordination
- 16 • Service Catalogue Management
- 17 • Service Level Management
- 18 • Availability Management
- 19 • Capacity Management
- 20 • IT Service Continuity Management
- 21 • Information Security Management
- 22 • Supplier Management



1
2 *Abbildung 6: Zusammenhang zwischen den Prozessen in Service Design (Great Britain: Cabinet Office, 2013)*

3 Die Abbildung 6 illustriert die internen Kommunikationswege zwischen den einzelnen Prozessen.
4 Aufgrund der Vielzahl an internen Überschneidungen ist ein eigener Prozess, der die
5 Koordination der anderen Prozesse übernimmt, vorgesehen. Neben dem Prozess Design
6 Coordination nimmt das Service Catalogue Management und das Service Level Management
7 zentrale Positionen im Service Design ein.

8 **3.3.1 Design Coordination**

9 Der Prozess Design Coordination in ITIL (2013) koordiniert alle Aktivitäten, Prozesse und
10 Ressourcen, die im Zusammenhang mit dem Design und Redesign von Services stehen.
11 Aufbauend auf dem Service Charter, der das Grobkonzept aus der Phase Service Strategy
12 darstellt, wird die Erstellung eines Service-Design-Packages (SDP) von diesem Prozess aus
13 koordiniert. Das Service Design Package ist das Detailkonzept des Service, welches beschreibt,
14 wie die Anforderungen aus Kundensicht erfüllt werden sollen. Das SDP deckt sowohl die
15 technische als auch organisatorische Perspektive ab. Design Coordination stellt sicher, dass alle
16 relevanten Aspekte des Service Designs betrachtet werden. Die genaue Planung des Service soll
17 einen Rückgang von Kosten, die durch Nacharbeiten entstehen, nach sich ziehen. Die
18 Verantwortung für Qualität, Sicherheit und Belastbarkeit der neuen oder geänderten Services
19 übernimmt der Service Design Manager.

20 Die Rolle des Service Design Managers sowie dessen Verantwortung werden in den
21 Beobachtungen unter Kapitel 2 vom Projektleiter übernommen.

1 **3.3.2 Service Catalogue Management**

2 Im Service Catalogue Management wird sichergestellt, dass das für den Kunden ersichtliche
3 Serviceangebot gepflegt wird und genaue Informationen zu den angebotenen Services
4 beinhaltet. Der Servicekatalog stellt eine vollständige Liste an vorhandenen und abrufbaren
5 Services dar und bildet die wechselseitigen Abhängigkeiten der Services untereinander ab. Er
6 dient als Informationsbasis für viele weitere ITIL-Prozesse. Er listet als Teil des Service Portfolios
7 nur aktive und vom Service Portfolio Management zum Deployment freigegebene Prozesse auf.
8 Die Verantwortung für die Richtigkeit und die Aktualität liegt beim Service Catalogue
9 Management. (Great Britain: Cabinet Office, 2013)

10 Ein explizites Service Catalogue Management ist in den beobachteten Arbeitsabläufen aus
11 Kapitel 2 nicht enthalten.

12 **3.3.3 Service Level Management**

13 Der Service Level Management (SLM) Prozess fokussiert sich auf die Kernaktivitäten rund um
14 die Vereinbarungen über Ziele und Ergebnisse zwischen Service Provider und Kunden. Er stellt
15 sicher, dass aktuelle sowie geplante IT Services den Vereinbarungen, die in Service Level
16 Agreements (SLA) dokumentiert werden, entsprechend erbracht werden.

17 Der Service Level Manager fungiert in Abstimmung mit dem Business Relationship Management
18 als Kommunikationsschnittstelle für alle Aspekte in Bezug auf Service Level. Er leistet somit einen
19 wichtigen Beitrag zur Entwicklung und Festigung einer Beziehung zwischen dem Service Provider
20 und dem Kunden.

21 Das SLM kümmert sich um die Erstellung von SLA, die die Vereinbarungen zwischen dem
22 Kunden und dem Service Provider schriftlich regeln. Besonderes Augenmerk wird daraufgelegt,
23 dass die Erwartungen des Kunden erkannt und berücksichtigt werden. Die Anforderungen und
24 Erwartungen des Kunden werden in Service Level Requirements (SLR) dokumentiert. Aufbauend
25 auf die SLR werden spezifische Ziele für den Service definiert und adäquate Methoden zur
26 Messung der Zielerreichung gewählt. Die definierten Ziele dienen in weiterer Folge als Maßstab
27 für die Bewertung der erbrachten Services.

28 Um den Anforderungen des SLA gerecht zu werden und eine Zielerreichung zu ermöglichen, ist
29 es teilweise notwendig (Teil-) Leistungen von externen Anbietern zu beziehen. Das Service Level
30 Management kümmert sich darum, dass die Vereinbarungen mit externen Dienstleistern in
31 gesonderten Verträgen Underpinning Contracts (UC), formalisiert werden. Werden für die
32 Erfüllung von SLA, (Teil-) Leistungen von unternehmensinternen Bereichen benötigt, werden
33 diese in Operational Level Agreements (OLA) zusammengefasst. Um die Erbringung der SLA
34 dauerhaft zu ermöglichen, werden die SLA einem Review unterzogen, bei dem das Augenmerk
35 auf eventuell veränderte Anforderungen des Kunden liegt.

36 Der Service Level Manager nach ITIL (2013) ist neben dem Monitoring der Zielerreichung auch
37 für das Management von Kundenrückmeldungen zuständig und für adäquate Reaktionen auf SLA
38 Verletzungen verantwortlich.

1 Teile des Service Level Managements werden während der Requirements Engineering Phase in
2 Kapitel 2.2 abgedeckt. Abhängigkeiten zu anderen Services wurden im beobachteten Szenario
3 nicht explizit niedergeschrieben, werden aber implizit betrachtet.

4 **3.3.4 Availability Management**

5 Die Verfügbarkeit eines IT-Service ist die Wahrscheinlichkeit, dass der Service die Anforderung
6 des Kunden zum Zeitpunkt des Abrufens erfüllt. Das Availability Management ist für die
7 Einhaltung der Verfügbarkeit des Service verantwortlich, die im Service Level Management
8 vereinbart und in SLA festgehalten wird. In den Bereich des Availability Managements fallen
9 neben dem Design von technischen Merkmalen und Prozeduren zur Erfüllung des vereinbarten
10 Verfügbarkeitslevels auch die Überwachung des Prozesses sowie das Reporting der Werte zu
11 anderen Prozessen. (Great Britain: Cabinet Office, 2013)

12 Die Lösung im beobachteten Szenario ist als Weiterentwicklung eines bestehenden Service zu
13 sehen, wodurch die Infrastruktur des bestehenden Service genutzt werden kann. Aus diesem
14 Grund wird dieser Prozess nicht ausgeführt.

15 **3.3.5 Capacity Management**

16 Das Capacity Management verwaltet die Kapazitäten der IT Organisation und stellt sicher, dass
17 entsprechende IT-Kapazität verfügbar ist, um den mit dem Kunden vereinbarten aktuellen und
18 zukünftigen Bedarf zu decken. Der mit dem Kunden vereinbarte Bedarf ist in Service Level
19 Agreements (SLA) dokumentiert. (Great Britain: Cabinet Office, 2013)

20 Im beobachteten Arbeitsablauf aus Kapitel 2 wird ein bestehendes Service erweitert, welches die
21 bereits genutzte Infrastruktur des Service verwendet. Der Capacity Management Prozess muss
22 nicht explizit ausgeführt werden.

23 **3.3.6 IT Service Continuity Management**

24 Der Fokus des IT Service Continuity Managements (ITSCM) liegt auf der Wiederherstellung im
25 Katastrophenfall und sorgt für eine angemessene Kontinuität des Service.

26 Eine der Hauptaufgaben von ITSCM bildet die regelmäßige Durchführung von Business Impact
27 Analysen, um die Aktualität der Kontinuitätspläne sicherzustellen. Eine weitere Hauptaufgabe ist
28 die Risikominderung, wodurch auch die Bewertung von Auswirkungen von geplanten Changes
29 auf die Kontinuitäts- sowie Wiederherstellungspläne beinhaltet ist. (Great Britain: Cabinet Office,
30 2013)

31 Die Lösung im beobachteten Szenario ist als Weiterentwicklung eines bestehenden Service zu
32 sehen, wodurch die Infrastruktur des bestehenden Service genutzt werden kann. Aus diesem
33 Grund wird dieser Prozess nicht ausgeführt.

34

1 **3.3.7 Information Security Management**

2 Das Information Security Management (ISM) unterstützt die Geschäftssicherheit. Es bestimmt die
3 Anforderungen hinsichtlich der Sicherheit eines Service und stellt die zentrale Stelle für alle IT
4 sicherheitsrelevanten Vorfälle dar. ISM ist ein Prozess, der sicherstellt, dass die Vertraulichkeit,
5 Integrität und Verfügbarkeit der Assets, Informationen, Daten und IT Services einer Organisation
6 die vereinbarten Bedürfnisse des Business erfüllen. (Great Britain: Cabinet Office, 2013)

7 In den in Kapitel 2 dokumentierten Beobachtungen werden Tätigkeiten mit ähnlicher Zielsetzung
8 in den IDs 18 und 19 durchgeführt.

9 **3.3.8 Supplier Management**

10 Das Supplier Management dient nach ITIL (2013) der Verwaltung der Partner und Lieferanten
11 sowie deren gelieferten Services, um die Qualität der gelieferten Leistungen dauerhaft zu
12 optimieren. Es ist für die Sicherstellung eines positiven Kosten-Nutzen-Verhältnisses eines
13 Lieferanten zuständig. Der Supplier Manager ist für die Existenz einer geeigneten Strategie
14 verantwortlich, die Leitlinien für die Beschaffung von Waren sowie Services festlegt. Sofern für
15 die Serviceerbringung Leistungen externer Dienstleister erforderlich sind, werden die
16 Rahmenbedingen der Leistungserbringung und Vereinbarungen in Underpinning Contracts
17 festgehalten. Underpinning Contracts sind inhaltlich auf die Service Level Agreements (SLA)
18 abgestimmt, die auf den Kunden ausgerichtet sind.

19 Ein explizites Supplier Management ist in den beobachteten Arbeitsabläufen aus Kapitel 2 nicht
20 enthalten. Die bestehenden Services (SharePoint Intranet Services und Salesforce CRM) haben
21 bestehende Verträge mit den Hauptlieferanten Microsoft und Salesforce.

22 **3.3.9 Service Design Package**

23 Das Service Design Package (SDP) ist kein Prozess von ITIL Service Design, sondern bildet das
24 formale Ergebnis der Phase. Die Ergebnisse der einzelnen Prozesse werden in einem Service
25 Design Package zusammengefasst. Es enthält neben den Anforderungen des Business auch
26 funktionale Anforderungen, Anforderungen in Bezug auf den erforderlichen Service Level und die
27 erforderliche Topologie. Ein SDP kann auch spezifische Richtlinien und Ziele beinhalten, die rein
28 die Phase Service Transition betreffen. Ebenfalls können Richtlinien und Abnahmekriterien für
29 die Phase Service Operations enthalten sein. (Great Britain: Cabinet Office, 2013)

30 Im beobachteten Ablauf zur Leistungserbringung, beschrieben in Kapitel 2, werden die typischen
31 Inhalte des Service Design Package in einer Projektplanungsdokumentation zusammengefasst,
32 die ausgearbeitete Konzepte inklusive funktionaler und nicht-funktionaler Anforderungen
33 umfasst.

1 **3.3.10 Kurzzusammenfassung**

2 Die Phase Service Design beschäftigt sich mit der Entwicklung von Services, um die
3 Unternehmenszielerreichung zu unterstützen. Sie beinhaltet Prozesse, die den vereinbarten
4 Level von aktuellen und zukünftigen IT Services sicherstellen, den Servicekatalog pflegen und ihr
5 Augenmerk darauf richten, dass alle entwickelten Servicemodelle den strategischen
6 Unternehmensanforderungen entsprechen. Darüber hinaus stellen die Prozesse in der Phase
7 Service Design sicher, dass auch den Anforderungen an Verfügbarkeit und Stabilität unter
8 Berücksichtigung des Informationsschutzes Genüge getan wird. Das Ergebnis der Phase Service
9 Design ist das Service Design Package, welches neben den Service Anforderungen auch
10 Richtlinien für die Umsetzung und den Betrieb des Service beinhaltet.

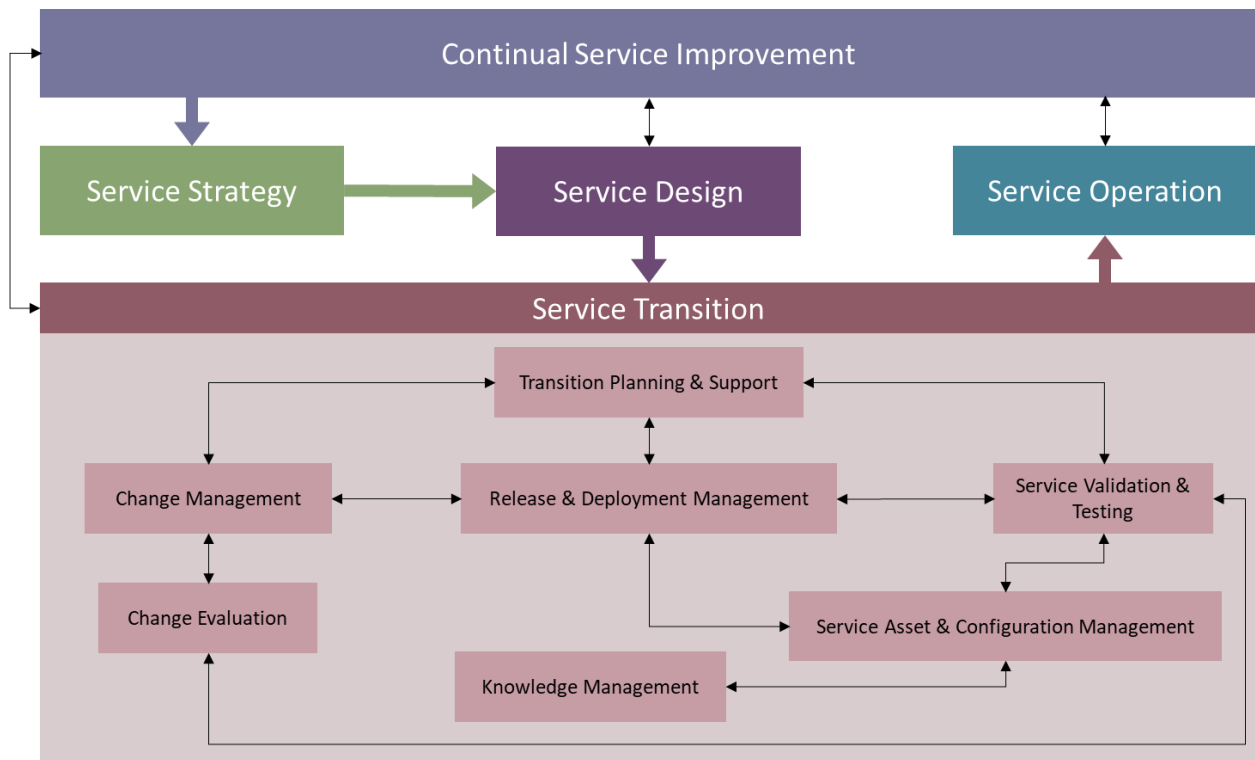
11 Bezugnehmend auf die Beobachtungen zum Ablauf der Leistungserbringung, werden Teile der
12 Anforderungen der Phase Service Design in den Tätigkeits-IDs 14 bis 20 abgedeckt.

13 **3.4 Service Transition**

14 Service Transition beschäftigt sich mit der geordneten Überführung zuvor definierter Services in
15 den Betrieb. Ziel von Service Transition ist sicherzustellen, dass Services den Erwartungen des
16 Unternehmens, welche in den vorangegangenen Phasen Service Strategy und Service Design
17 festgelegt wurden, entsprechen. Das betrifft sowohl neue, zu ändernde als auch stillzulegende
18 Services. Bei neuen Services und Major Changes bildet das Services Design Package (SDP) aus
19 der Phase Service Design die Arbeitsgrundlage.

20 Abbildung 4: ITIL Service Lifecycle zeigt die Eingliederung der Phase Service Transition in den
21 ITIL Service Lifecycle und veranschaulicht, dass Service Transition auf den Ergebnissen der
22 Phase Service Design aufbaut und als Informationslieferant für die nachgelagerte Phase Service
23 Operations dient. Die Phase Service Transition umfasst folgende sieben Prozesse:

- 24 • Transition Planning and Support
- 25 • Change Management
- 26 • Service Asset and Configuration Management
- 27 • Release and Deployment Management
- 28 • Service Validation and Testing
- 29 • Change Evaluation
- 30 • Knowledge Management



1
2 *Abbildung 7: Zusammenhang zwischen den Prozessen in Service Transition (Great Britain: Cabinet Office, 2013)*

3 Aus Abbildung 7 geht der Zusammenhang zwischen den sieben Hauptprozessen in Service
4 Transition hervor. Der Prozess Change Management ist der Kern von Service Transition und
5 dient als Kommunikationsschnittstelle zu den Phasen Service Design und Service Operations.
6 Nachfolgend werden die Hauptprozesse der Phase Service Transition näher beschrieben.

7 **3.4.1 Transition Planning and Support**

8 Das Ziel des Prozesses Transition Planning and Support ist die Durchführung der
9 Gesamtplanung der Service Transition und die Koordination der dafür benötigten Ressourcen.
10 Ziel des Prozesses ist die Sicherstellung, dass die in den vorangegangenen Phasen Service
11 Design und Service Strategy definierten Anforderungen effektiv realisiert und der Service in die
12 nachgelagerte Phase Service Operation überführt wird. Zum Umfang des Prozesses gehören
13 neben dem Lenken und Steuern mehrerer Transitions auch die Ressourcenpriorisierung bei
14 Engpässen und die Budgetplanung der Transitions. (Great Britain: Cabinet Office, 2013)

15 Da sich die Transition innerhalb eines Zeit-, Kosten- und Qualitätsrahmens abspielt, handelt es
16 sich bei Major Changes um Service Transition Projekte. Der Prozess Transition Planning and
17 Support kann somit als das Projektmanagement von ITIL angesehen werden. (Essmat, 2016)

18 Der Herausgeber von ITIL, Axelos Limited, bietet ebenso ein Projektmanagement Framework
19 unter der Marke PRINCE2 an. In den Beobachtungen unter Kapitel 2 folgt das
20 Projektmanagement dem Framework der International Project Management Association (IPMA).
21 Trotzdem kann die Rolle des Projektleiters klar den in Transition Planning and Support
22 beschriebenen Umfängen zugeordnet werden.

1 **3.4.2 Change Management**

2 Der Change Management Prozess in ITIL hat das vorrangige Ziel Änderungen an IT Systemen
3 gesteuert aufzuzeichnen und so zu managen, dass das Risiko für negative Auswirkungen auf
4 Services minimiert wird. Die in ITIL gültige Definition eines Changes ist das „Hinzufügen,
5 Modifizieren, oder Entfernen eines Elements, das Auswirkungen auf die IT Services haben
6 könnte“. (Great Britain: Cabinet Office, 2013, S. 75)

7 Neben der Reduktion des Geschäftsrisikos stellt der Change Management Prozess sicher, dass
8 jeder implementierte Change einen geschäftlichen Mehrwert bietet und dieser gemessen und
9 dokumentiert wird. Nicht autorisierte Änderungen werden verhindert und Verantwortlichkeiten
10 sowie Zuständigkeiten für Changes werden definiert. Der Change Management Prozess sieht
11 vor, dass Änderungen in Form eines Change Requests im Vorfeld dokumentiert werden. Change
12 Requests umfassen die Beschreibung des Changes, das erwartete Risiko und den erwarteten
13 Mehrwert. Auf Basis des Change Requests werden die Planung der benötigten Ressourcen für
14 die Paketierung, das Testen und das Deployment des Releases in der Produktivumgebung
15 geplant. In ITIL werden grundsätzlich drei Typen von Changes unterschieden: (Great Britain:
16 Cabinet Office, 2013)

- 17 • **Standard-Change:** Ein Standard-Change ist üblicherweise mit sehr geringem Risiko
18 behaftet und dessen Implementierung folgt einem definierten und dokumentierten
19 Prozess oder einer Arbeitsanweisung. Standard-Changes bedürfen keiner gesonderten
20 Autorisierung, sind gut nachvollziehbar und folgen einem definierten Anstoß. Standard-
21 Changes werden, gleich wie Emergency-Changes und Normal-Changes, protokolliert und
22 deren Durchführung dokumentiert.
- 23 • **Normal-Change:** Ein Normal-Change ist eine Änderung, die den definierten Change
24 Management Prozess durchlaufen muss.
- 25 • **Emergency-Change:** Ein Emergency-Change ist ein Normal-Change mit höchster
26 Priorität für die Organisation. Emergency Changes sind als Changes definiert, die in
27 kürzester Zeit evaluiert, geprüft und entweder genehmigt oder abgelehnt werden.

28 Das Durchlaufen dieses Prozesses schafft ein Bewusstsein für die zu erwartenden Auswirkungen
29 einer Änderung und leistet somit einen Beitrag zur Verbesserung der Systemstabilität.

30 In den Beobachtungen der Leistungserbringung, die in Kapitel 2 beschrieben sind, bildet der
31 Change Management Prozess eine zentrale Rolle in der Überführung der Lösung in die
32 Produktivumgebung. Kapitel 2.3.2 gibt zudem einen Überblick über eine Implementierung der in
33 ITIL vorgeschlagenen Inhalte eines Change Management Prozesses.

34 **3.4.3 Service Asset and Configuration Management**

35 Service Asset and Configuration Management (SACM) ist für die Steuerung von Ressourcen und
36 Fähigkeiten verantwortlich, die für die Erbringung eines Service erforderlich sind. Ein weiterer Teil
37 des Service Asset and Configuration Managements ist die Bereitstellung von genauen

1 Informationen über die Konfiguration der Assets und deren Beziehung zueinander. Assets, die
2 gesondert gemanagt werden müssen, werden Configuration Items genannt. Configuration Items
3 (CI) eines Service können Infrastrukturkomponenten, Richtlinien, Dokumentationen, Lieferanten
4 aber auch Mitarbeiter und andere Services sein, die zur Erbringung von IT Services erforderlich
5 sind. CIs werden inklusive ihrer Beziehungen zueinander in einer Configuration Management
6 Database (CMDB) dokumentiert. Mehrere CMDBs werden zum übergeordneten Configuration
7 Management System (CMS) zusammengefasst. (Great Britain: Cabinet Office, 2013)

8 Der Prozess SACM ist für die Pflege sowie die Aktualisierung des CMS verantwortlich und führt
9 regelmäßige Checks durch, die sicherstellen, dass das CMS die aktuell produktiv verwendeten
10 CIs repräsentiert. Innerhalb des Service Asset und Configuration Managements wird geregelt,
11 wer befugt ist, die Struktur und den Inhalt des CMS zu ändern. Der Configuration Manager ist für
12 das Service Asset und Configuration Management verantwortlich.

13 Die Beobachtungen zum Ablauf der Leistungserbringungen, beschrieben in Kapitel 2, zeigen,
14 dass das Configuration Management in die Verantwortung der Service Owner fällt und auch im
15 Change Management Prozess Berücksichtigung findet.

16 **3.4.4 Release and Deployment Management**

17 Das Release and Deployment Management verfolgt das Ziel die Builds, Tests und die
18 Deployments von Releases zu steuern. Das beinhaltet neben der Bereitstellung der neu
19 entwickelten Features auch die Vereinbarung und Festlegung von Release und Deployment
20 Plänen und die Wissensübertragung zu den Service Operations Funktionen. Das Release und
21 Deployment Management kümmert sich darum, dass Tests durchgeführt werden. Das Testen
22 selbst erfolgt im Prozess Service Validation and Testing. (Great Britain: Cabinet Office, 2013)

23 Das International Software Testing Qualifications Board (ISTQB) (2018) definiert das Testen als:
24 „Der Prozess, der aus allen Aktivitäten des Lebenszyklus besteht (sowohl statisch als auch
25 dynamisch), die sich mit der Planung, Vorbereitung und Bewertung eines Softwareprodukts und
26 dazugehöriger Arbeitsergebnisse befassen. Ziel des Prozesses ist sicherzustellen, dass diese
27 allen festgelegten Anforderungen genügen, dass sie ihren Zweck erfüllen und etwaige
28 Fehlerzustände zu finden.“

29 Im Umfeld von ITIL wird der Prozess des Testens von Software nach ISTQB in zwei Prozesse
30 aufgeteilt. Das Release and Deployment Management übernimmt die Vorbereitung und die
31 Planung. Die operativen Tätigkeiten des Testens erfolgen im Prozess Service Validation and
32 Testing.

33 Das Release and Deployment Management sieht die Erstellung von Richtlinien vor, um den
34 Umgang mit Releases und Deployments zu vereinheitlichen. Als Beispiel für eine derartige
35 Richtlinie nennt ITIL eine Richtlinie auf Service Provider Ebene: „Alle Changes und Releases
36 müssen vor dem Deployment unter realistischen Arbeitsbedingungen vollständig getestet
37 werden“. (Great Britain: Cabinet Office, 2013, S. 132)

1 Derartig allgemein gehaltene Richtlinien stellen Service Provider, die in stark verknüpften
2 Serviceverflechtungen agieren, vor das Problem, dass die Erstellung realistischer
3 Arbeitsbedingungen einen großen zeitlichen Aufwand mit sich bringt. Ebenfalls ist bei Services,
4 die auf Individualentwicklungen basieren, oder auch bei größeren Changes an individuell
5 entwickelten Services ein vollständiges Testen nur mit sehr hohem Testaufwand möglich. Um
6 einer derartigen Richtlinie Folge zu leisten, ist demnach mit sehr hohen Prüfkosten zu rechnen.

7 Innerhalb des Release and Deployment Management Prozesses werden Release Packages
8 erstellt, welche eine Kombination aus den Tests und Deployments sowie den notwendigen
9 Configuration Items sind. (Great Britain: Cabinet Office, 2013)

10 Im beobachteten Szenario wird das Release Package nicht explizit genannt. Durch das
11 mehrstufige Verfahren im Deployment (Deployment und Test im Qualitätssystem sowie
12 Deployment und Test im Produktivsystem) und die Anpassungen der Schnittstellen, die unter
13 anderem ein Beispiel für ein Configuration Item darstellen, werden die in ITIL genannten
14 Erfordernisse (Tests, Deployments, Configuration Items) erfüllt.

15 **3.4.5 Service Validation and Testing**

16 Der ITIL Prozess Service Validation and Testing setzt die geplanten Tests entsprechend des
17 vorgelagerten Prozess Release and Deployment Management um. Der Prozess dient der
18 Qualitätssicherung und prüft, ob der Service die vereinbarte Einsatzfähigkeit (Warranty) liefert
19 und für das Problem oder die Anforderungen des Kunden zweckmäßig ist (Utility). Diese Prüfung
20 erfolgt auf Basis des Service Design Packages, welches die vereinbarte Service Utility und
21 Warranty im Service Charter dokumentiert hat. Diese Prüfung beinhaltet ebenso die
22 Übereinstimmung der Leistungsfähigkeit des Service mit dem im SLA vereinbarten Servicelevel.
23 Neben diesen beiden Prüfungen besteht das Ziel des Prozesses in der Identifikation von Fehlern
24 und Risiken und in deren Behandlung.

25 ITIL (Great Britain: Cabinet Office, 2013, S. 173) definiert als eines der Ziele, dass dieser Prozess
26 bestätigen soll, dass „[...] alle Fehler oder Abweichungen zu Beginn des Lebenszyklus beseitigt
27 wurden [...]“.

28 Um bestätigen zu können, dass alle Fehler vorab beseitigt wurden, ist eine genaue
29 Dokumentation der getätigten Testschritte erforderlich. Um die möglichen Fehler entdecken zu
30 können, ist eine Testumgebung erforderlich, die der Produktivumgebung so nah wie möglich
31 nachempfunden ist. Gerade bei Testumgebungen muss auf die Datensicherheit geachtet werden
32 und eine klare und offensichtliche Trennung zwischen Test- und Produktivdaten bestehen.
33 Ebenso müssen die Testmodelle so aufgebaut sein, dass sie wiederholbar und nachvollziehbar
34 sind. Bei komplexen, selbst entwickelten Softwarepaketen entsteht durch diese Anforderung eine
35 Unmenge an möglichen Testfällen. Daraus resultiert eine große Zeitspanne, die zwischen
36 Fertigstellung der Entwicklung und dem Nutzen für den Kunden entsteht.

37 Die Beobachtungen aus Kapitel 2 zeigen, dass die in der Projektdokumentation enthaltenen
38 Konzepte (analog zum Service Design Package) die Anforderungen des Auftraggebers abbilden.

1 Das Testen im Test- und Qualitätssystem dient zur Abdeckung der Erfordernisse aus dem
2 Prozess Service Validation and Testing. Bei der Überprüfung der Zweckmäßigkeit (Utility)
3 während des Kundentests konnten Abweichungen (nicht umgesetzte implizite Anforderung)
4 entdeckt und noch vor dem Produktivbetrieb beseitigt werden.

5 **3.4.6 Change Evaluations**

6 Der Prozess Change Evaluations stellt eine standardisierte Methode zur Bewertung von positiven
7 Auswirkungen eines Changes bereit. Change Evaluations fokussiert die Wertschöpfung des
8 Changes in Bezug auf den zu ändernden Service und identifiziert Risiken, die in Verbindung mit
9 der gemeinsam genutzten Infrastruktur oder anderen Prozessen stehen. Neben den zu
10 erwarteten Auswirkungen legt der Change Evaluations Prozess sein Augenmerk auch auf die
11 unerwarteten Folgen eines Changes und sorgt für die Schaffung von Nachvollziehbarkeit der
12 gesamten Auswirkungen des Changes. Eine Evaluierung wird für jeden Change durchgeführt,
13 jedoch durchlaufen nur Changes, die weitreichende Folgen nach sich ziehen, den formalen
14 Change Evaluation Prozess. Die Ergebnisse des Change Evaluation Prozesses dienen oft als
15 Entscheidungsgrundlage für das Change Management in Bezug auf eine Change Autorisierung.
16 Eine Change Evaluierung ist nicht bei jedem Change vorgesehen und bedarf eines Antrags aus
17 dem Change Management. (Great Britain: Cabinet Office, 2013)

18 Im beobachteten Tätigkeitsablauf in Kapitel 2 umfasst der Change Management Prozess in
19 seinen vorbereitenden Schritten bereits die Change Evaluierung.

20 **3.4.7 Knowledge Management**

21 Der Prozess Knowledge Management stellt sicher, dass die richtigen Informationen zur richtigen
22 Zeit am passenden Ort für die Personen, die sie benötigen, zur Verfügung stehen, um fundierte
23 Entscheidungen zu ermöglichen. Durch diesen Prozess wird die Effizienz der
24 Entscheidungsfindung erhöht, da bereits vorhandenes Wissen abrufbereit ist und nicht erneut
25 entwickelt werden muss. Ein Hauptziel ist die Sammlung, Speicherung und Verwendung von
26 Daten, Informationen und Wissen. (Great Britain: Cabinet Office, 2013)

27 ITIL bedient sich dem DIKW-Modell von Russel L. Ackoff (1989), welches den Übergang
28 zwischen Daten zu Wissen beschreibt. DIKW steht als Akronym für Data, Information,
29 Knowledge, Wisdom. In diesem Kontext bezeichnen Daten (Data) die rohen, unstrukturierten
30 Fakten, die im Normalfall für den Menschen in einer nicht verständlichen Form aufliegen. Daten
31 werden zu Informationen (Information), sobald diese strukturiert sind und in Kontext gesetzt
32 worden sind. Wissen (Knowledge) bezeichnet in Beziehung gesetzte Informationen und
33 beschreibt das Verständnis von Mustern. Weisheit (Wisdom) beruht auf Wissen und erfordert das
34 Verständnis der zugrundeliegenden Prinzipien.

35 Service Assets und CIs sowie deren Verknüpfung untereinander werden in Configuration
36 Management Databases gespeichert und im Configuration Management System gesammelt
37 abgespeichert. Das Configuration Management System bildet daher einen besonders wichtigen

1 Bestandteil des übergeordneten Service Knowledge Management System (SKMS). Im SKMS
2 werden weitere servicebezogene Informationen, wie beispielsweise das Service Portfolio,
3 Verträge, wie OLA und SLA gespeichert, oder auch eine Known Error Datenbank gepflegt. Das
4 SKMS bildet somit einen zentralen Speicherort für Daten, Informationen und Wissen, welche aus
5 allen Lebenszyklusphasen in ITIL sowohl befüllt als auch genutzt werden. Einen besonders
6 berücksichtigungswürdigen Aspekt bildet hier der Daten- und Zugriffsschutz. (Great Britain:
7 Cabinet Office, 2013)

8 Der Prozess Knowledge Management ist in den Beobachtungen aus Kapitel 2 nicht explizit
9 erkennbar umgesetzt. Erste Ansätze sind im Bereich des Wissenstransfers in Richtung IT
10 Applikationssupport zu erkennen, in dem erwartete Fehler und deren standardisierte
11 Lösungswege dokumentiert übergeben werden. Des Weiteren pflegen die Service Owner die
12 CMDB ihrer Services, um Wissen zu verteilen.

13 **3.4.8 Kurzzusammenfassung**

14 In der ITIL Phase Service Transition erfolgt die Inbetriebnahme neuer oder geänderter Services,
15 die in der vorgelagerten Phase Service Design konzipiert wurden. Während sich der Prozess
16 Transition Planning and Support um die Umsetzung der Anforderungen mit definierten
17 Ressourcen kümmert, ist das Change Management vorwiegend für das Management von
18 Änderungen an bestehenden IT Systemen verantwortlich. Das Release and Deployment
19 Management stellt gemeinsam mit dem Prozess Service Validation and Testing sicher, dass
20 Services vor der produktiven Nutzung ausreichend getestet werden. Das Knowledge
21 Management kümmert sich unter Zuhilfenahme des Service Asset and Configuration
22 Managements um die zentrale Speicherung, die Verteilung und den Schutz von
23 servicebezogenen Informationen.

24 Die Beobachtungen in Kapitel 2 betreffen hauptsächlich Tätigkeiten, die dieser Phase
25 zuzuordnen sind. Es ist gut erkennbar, dass sich die Kernaktivitäten nahezu jedes Prozesses im
26 beobachteten Szenario wiederfinden.

27 **3.5 Service Operation**

28 ITIL Service Operation ist die Phase, in der die Konzepte und Implementierungen der
29 vorangegangenen Phasen Service Strategy, Design und Transition dem Servicekunden im
30 Tagesgeschäft zur Verfügung gestellt werden. Service Operations stellt durch den richtigen
31 Einsatz von Personen, Prozessen, Infrastruktur und Beziehungen eine hohe Qualität der
32 angebotenen Services sicher und sorgt für die Bereitstellung der vereinbarten Servicelevel bei
33 optimaler Kosteneffektivität. Durch die Bereitstellung und den Support der Services, bildet Service
34 Operations die Schnittstelle zum Endanwender und zum Kunden des Service. Er ermöglicht
35 dadurch das Sammeln wertvollen Feedbacks und dessen Rückführung in die Phasen Service
36 Strategy, Design und Transition. (Great Britain: Cabinet Office, 2013)

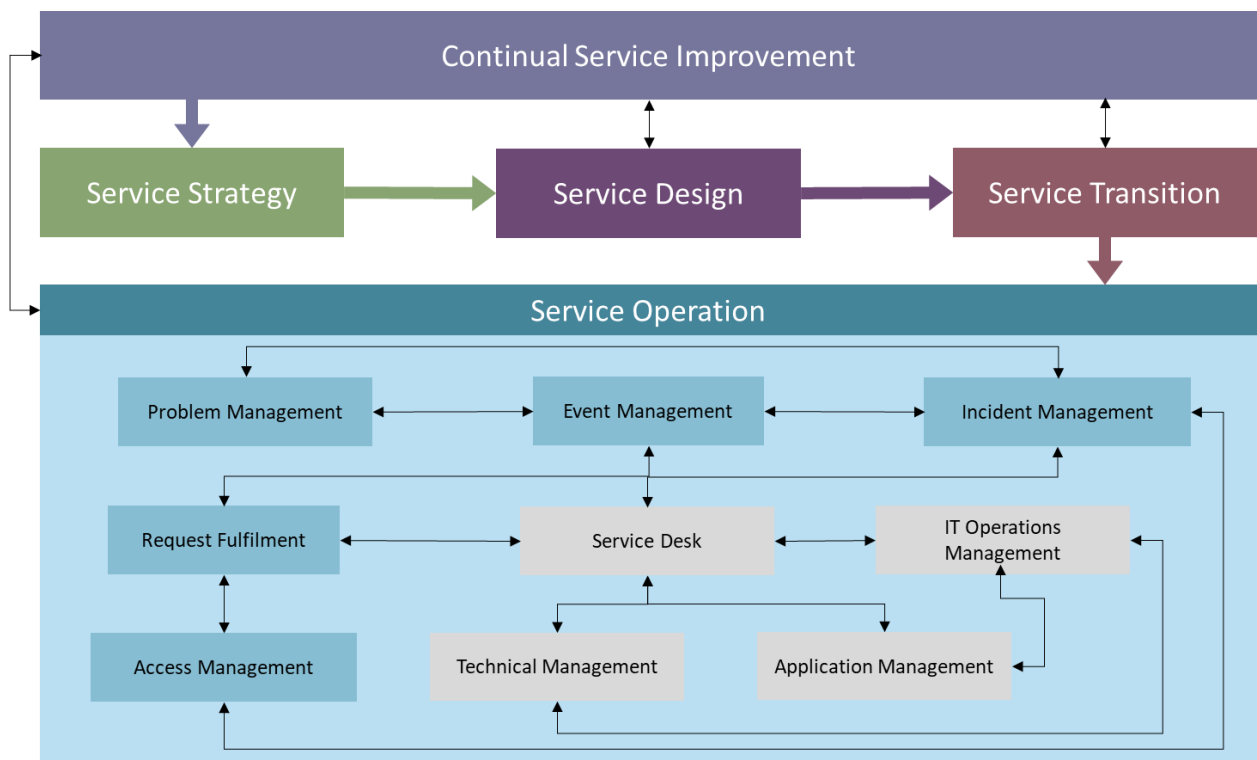
1 Abbildung 4: ITIL Service Lifecycle veranschaulicht die Eingliederung der Phase Service
2 Operations in den ITIL Service Lifecycle und zeigt, dass die Ergebnisse der vorangegangenen
3 Phase Service Transition als Grundlage für die Phase dienen. Die Phase Service Operations
4 umfasst die folgenden fünf Prozesse sowie vier Funktionen:

5 **Prozesse**

- 6 • Event Management
- 7 • Incident Management
- 8 • Request Fulfillment
- 9 • Access Management
- 10 • Problem Management

11 **Funktionen**

- 12 • Service Desk
- 13 • Technical Management
- 14 • IT Operations Management
- 15 • Application Management



16
17 *Abbildung 8: Zusammenhang zwischen den Prozessen in Service Operations (Great Britain: Cabinet Office, 2013)*

18 Abbildung 8 zeigt den Zusammenhang zwischen den fünf Hauptprozessen sowie den vier
19 Funktionen in Service Operations. Der Kontakt zum Endanwender ist in zwei Prozessen sowie
20 einer Funktion vorgesehen. Dadurch ist Service Operations jene Phase mit dem meisten

1 Endanwenderkontakt. Nachfolgend wird auf die Hauptprozesse und die Funktionen der Phase
2 Service Operations näher eingegangen.

3 Das beobachtete Szenario in Kapitel 2 endet mit der Übergabe an den IT Applikationssupport.
4 Sowohl im Service SharePoint Intranet Services als auch im Service Salesforce ändern sich die
5 Prozesse betreffend Service Operations nicht. Die Informationsbasis für den Service Desk
6 (Configuration Management Database und Known Error Database) wird beim Handover Meeting
7 erweitert.

8 **3.5.1 Event Management**

9 Ziel des Event Management Prozesses ist das Identifizieren von Statusmeldungen (Events), die
10 für die Verfügbarkeit und die Nutzbarkeit von Configuration Items relevant sind. Darauf aufbauend
11 zielt der Prozess auf die Definition von Maßnahmen ab, um die aufgetretenen Events an die
12 richtigen Stellen zur Bearbeitung weiterzuleiten. Aufgezeichnete Events können die Basis für
13 Automatisierungen bilden und können die Grundlage für den Leistungsvergleich und die
14 kontinuierliche Verbesserung des Service darstellen. (Great Britain: Cabinet Office, 2013)

15 **3.5.2 Incident Management**

16 In ITIL wird ein Incident als „eine nicht geplante Unterbrechung eines IT Service, eine
17 Qualitätsminderung [...] oder ein Ausfall eines CI ohne bisherige Auswirkungen auf einen IT
18 Service [...]“ (Great Britain: Cabinet Office, 2013, S. 83) beschrieben. Die Bearbeitung dieser
19 Incidents wird im Incident Management vorgenommen. Der Incident Management Prozess kann
20 aus unterschiedlichen Quellen gestartet werden. Beispielsweise durch eine Benachrichtigung aus
21 dem Event Management Prozess oder durch einen Anruf beim Service Desk. Oberste Priorität
22 im Incident Management Prozess hat die schnellstmögliche Wiederherstellung eines Zustands,
23 der eine Nutzung des Service ermöglicht. Im Incident Management steht die schnelle
24 Bereitstellung einer temporären Lösung über der Ursachenbekämpfung. Im Incident Management
25 werden alle Ereignisse betrachtet, die zu einer Minderung der Nutzbarkeit eines Service führen
26 oder führen können. Der Incident Management Prozess stellt standardisierte Methoden zur
27 Bearbeitung von Incidents bereit und schafft dadurch einen professionellen Ansatz für die Lösung
28 von Incidents, der zu einer höheren Anwenderzufriedenheit führt. Innerhalb des Incident
29 Managements wird auf Basis der erwarteten Auswirkungen zwischen Standard-Incidents und
30 Major-Incidents unterschieden. Letztere erfordern ein gesondertes Verfahren und genießen
31 aufgrund der hohen zu erwartenden Auswirkungen eine höhere Priorität. Je nach Art des Major-
32 Incidents kann für diesen Zwischenfall ein eigener Incident Manager definiert werden, der den
33 Fortschritt überwacht und die schnellstmögliche Lösung forciert. Klar abzugrenzen ist ein (Major)
34 Incident zu einem Problem. Das Problem bildet die zugrundeliegende Ursache für einen oder
35 mehrere Incidents, die in einem eigenen Prozess bearbeitet wird und als losgelöste Einheit zu
36 betrachten ist. (Great Britain: Cabinet Office, 2013)

37 Beispiele für Incidents können sein:

- 1 • Ausfall einer Festplatte in einem Server
- 2 • Anmeldeproblem in der Webmaske
- 3 • Fehlermeldung in einer Applikation
- 4 • Ausfall der Datenbankverbindung auf einem Server
- 5 • Nichterreichbarkeit eines Web-Servers

6 **3.5.3 Request Fulfilment**

7 Service Requests, die im Prozess Request Fulfilment behandelt werden, sind in vielen Fällen
8 Anfragen von Kunden, die Informationen oder Änderungen wünschen. In der Regel weisen sie
9 äußerst geringes Risiko und häufig geringe Kosten auf. Im Gegensatz zu Incidents können und
10 sollen Service Requests geplant werden und standardisiert abgewickelt werden. (Great Britain:
11 Cabinet Office, 2013)

12 **3.5.4 Problem Management**

13 Der Problem Management Prozess behandelt die Erkennung, Behebung und Dokumentation von
14 Problemen, die mit der IT Infrastruktur zusammenhängen. Ursachenanalysen werden meist als
15 Folge von Incidents initiiert und dienen der Dokumentation von bekannten Fehlern (Known
16 Errors). Hiermit wird versucht Incidents proaktiv zu vermeiden und die Auswirkungen von
17 unvermeidbaren Incidents zu verringern. (Great Britain: Cabinet Office, 2013)

18 Aus dem Problem Management heraus kann eine Änderungsanforderung entstehen, wodurch
19 eine Rückkoppelung in die Phase Service Design geschieht.

20 **3.5.5 Access Management**

21 Ziel des Access Management Prozesses ist die Berechtigungssteuerung bei Services. Diese
22 umfasst das Erteilen von Zugriffsberechtigungen für autorisierte Personen und das Verhindern
23 von Zugriffen von nicht autorisierten Personen. (Great Britain: Cabinet Office, 2013)

24 **3.5.6 Funktionen**

25 ITIL Service Operations (2013) kennt vier Funktionen, die nachfolgend überblicksmäßig
26 beschrieben werden.

27 **Service Desk**

28 Die Funktion Service Desk dient als zentrale Anlaufstelle für alle servicebezogenen Anfragen.
29 Der Service Desk dient in Richtung der Nutzer als Single Point of Contact (SPOC), wodurch der
30 Nutzer stets weiß, wohin er sich mit seiner Anfrage wenden kann. Die Funktion Service Desk hat
31 als zentrale Aufgabe das Entgegennehmen von Anfragen. Bei diesen Anfragen kann es sich um
32 Änderungswünsche oder Störungsmeldungen handeln oder es werden nähere Informationen

1 zum gebotenen Service gewünscht. Als Eingangskanäle können neben E-Mail auch andere
2 schriftliche Medien (beispielsweise Fax, Onlineformulare etc.) oder auditive Medien
3 (beispielsweise Telefon, Skype, etc.) dienen. Der Service Desk koordiniert die Anfragen der
4 Nutzer und initiiert fallbezogen andere Prozesse wie beispielsweise einen Incident Management
5 Prozess, einen Access Management Prozess oder einen Problem Management Prozess.

6 Aus dem beobachteten Szenario (Kapitel 2) geht hervor, dass diese Funktion im Unternehmen
7 vorhanden ist und beim Support von IT Applikationen die zentrale Schnittstelle zum Nutzer bildet.

8 **IT Operations Management**

9 Diese Funktion gliedert sich in das Facility Management und die Betriebssteuerung. Das Facility
10 Management befasst sich mit dem Management von Rechenzentren und Computerräumen. Die
11 Betriebssteuerung stellt operative Routineaufgaben sicher und ist für das zentralisierte Monitoring
12 zuständig.

13 **Technical Management**

14 Das Technical Management stellt Ressourcen und technische Fähigkeiten zur Unterstützung des
15 Aufbaus von IT Infrastruktur bereit. Die Funktion kümmert sich zudem um den operativen Betrieb
16 der IT Infrastruktur. Mitarbeiter des Technical Management sind häufig auch Teil der IT
17 Operations Management Funktion.

18 **Application Management**

19 Das Application Management ist für das Management des Lebenszyklus von Anwendungen
20 zuständig. Mitarbeiter des Technical Management sind häufig auch Teil der IT Operations
21 Management Funktion.

22 **3.5.7 Kurzzusammenfassung**

23 ITIL Service Operations ist die Phase, die den Betrieb eines Service begleitet und für Probleme,
24 Fragen und Anforderungen der Endbenutzer bereitsteht. Durch proaktive Tätigkeiten kann die
25 Anzahl von potenziellen Fehlerfällen vermindert werden und die Kundenzufriedenheit auf einem
26 hohen Level gehalten werden. Der Service Desk, der als Single Point of Contact für
27 Benutzeranfragen dient, sammelt wertvolle Informationen, die zur Verbesserung des Service
28 herangezogen werden können. Diese Informationen werden beim Design ähnlicher Services
29 genutzt, um den neu zu erstellenden Service besser auf seine zukünftigen Nutzer abzustimmen.

30 **3.6 Continual Service Improvement**

31 Die letzte Phase im ITIL-Servicelebenszyklus befasst sich mit der kontinuierlichen Suche nach
32 Wegen, die Effektivität und Effizienz von Prozessen und Services zu verbessern. Continual
33 Service Improvement (CSI) baut auf dem Grundsatz auf, dass man erst in der Lage ist etwas zu
34 steuern, wenn man in der Lage ist es auch zu messen. Das Ziel von CSI ist neben der
35 Identifizierung von Aktivitäten zur Qualitätssteigerung in der Leistungserbringung auch das

1 Finden von Möglichkeiten, um die Wirtschaftlichkeit einer Prozessausführung zu erhöhen. Um
 2 diese Ziele zu erreichen sind regelmäßige interne Audits, Kundenzufriedenheitsumfragen und
 3 Bewertungen von geschäftlichen Prognosen erforderlich. Um Verbesserungen in der Effizienz
 4 oder Effektivität eines Service zu erreichen, ist ein Change im Service selbst notwendig. Diese
 5 Changes durchlaufen den, in der Phase Service Transition definierten, Change Management
 6 Prozess. (Great Britain: Cabinet Office, 2013)

7 Abbildung 4: ITIL Service Lifecycle zeigt, dass das Continual Service Improvement den gesamten
 8 Servicelebenszyklus umfasst und Anknüpfungspunkte zu den Phasen Service Design, Service
 9 Transition und Service Operations hat. Die Verbesserungsaktivitäten unterstützen ebenso die
 10 Phase Service Strategy, indem das Service Portfolio permanent mit den aktuellen und
 11 zukünftigen Bedürfnissen des Kunden abgestimmt wird. CSI ist besonders für Personen relevant,
 12 die kostenverantwortliche Rollen innerhalb der Serviceorganisation einnehmen. Dies sind
 13 beispielsweise Process Owner, Service Owner oder IT Manager. Auch für IT Architekten sowie
 14 Business Relationship Manager ist CSI von hoher Relevanz.

15 **3.6.1 Seven-Step Improvement Process**

16 Der Seven-Step Improvement Process besteht aus sieben aufeinanderfolgenden Schritten,
 17 welche den PDCA-Zyklus, auch Deming-Cycle genannt, integrieren. In ITIL (2013) ist die
 18 Zuordnung zwischen dem PDCA-Zyklus und dem Seven-Step Improvement Process wie folgt
 19 dokumentiert:

Plan	1.) Identifizieren der Verbesserungsstrategie
	2.) Definieren, was gemessen wird
Do	3.) Erfassen von Daten
	4.) Verarbeiten von Daten
Check	5.) Analysieren der Informationen und Daten
	6.) Präsentieren und Nutzen der Informationen
Act	7.) Implementieren der Verbesserungsmaßnahmen

20 Bereits aus den Titeln der sieben Schritte ist erkennbar, dass hier das DIKW-Modell (Data –
 21 Information – Knowledge – Wisdom) zum Einsatz kommt. Das DIKW Modell von Russel L. Ackoff
 22 (1989) beschreibt die Unterscheidung zwischen Daten, Informationen, Wissen und Weisheit,
 23 wobei der Kern wie folgt zusammengefasst werden kann:

- 24 • Daten sind einzelne, unstrukturierte und somit nicht nutzbare Fakten und Beobachtungen.
- 25 • Informationen sind zusammengeführte, strukturierte und in Beziehung gesetzte Daten.
- 26 • Wissen beschreibt in Kontext gebrachte und interpretierte Informationen.

- 1 • Weisheit bezeichnet die korrekte Anwendung von Wissen.

2 Die Schritte 2 und 3 dienen zur Bereitstellung von Daten (Data), Schritt 4 wandelt die Daten in
3 Informationen (Information) um, Schritt 5 und 6 basieren auf Wissen (Knowledge) und Schritt 7
4 spiegelt die Weisheit (Wisdom) wider. Schritt 1 baut ebenfalls auf die korrekte Anwendung bereits
5 vorhandenen Wissens auf und ist somit der Weisheit (Wisdom) zuzuordnen.

6 **Identifizieren der Verbesserungsstrategie**

7 Auf Basis der Vision, der Mission, der Strategie sowie der taktischen und operativen Ziele werden
8 die Felder identifiziert, in denen Verbesserungspotentiale gesucht werden. Das Ziel ist
9 herauszufinden, wie Verbesserungen innerhalb von Prozessen und Services dazu beitragen
10 können, die gesetzten geschäftlichen Ziele zu erreichen oder der Vision näher zu kommen.

11 **Definieren, was gemessen wird**

12 In den beiden frühen Phasen im Servicelebenszyklus sollten bereits Überlegungen zu den
13 potenziell messbaren Leistungsindikatoren eines Service erfolgt sein, wodurch diese die Basis
14 für diesen Schritt bilden. ITIL (2013, S. 57) beschreibt ein Leitprinzip, welches besagt: „[...]dass
15 das zu messen ist, was am wichtigsten ist“ und verweist darauf, dass die Messungen an den
16 Zielen des Kunden ausgerichtet sein sollen.

17 **Erfassen von Daten**

18 Sobald definiert wurde, welche Informationen gemessen werden sollen, erfolgt die
19 Servicemessung. Erfasst werden alle Daten, die als erforderlich definiert wurden, wobei eine
20 Standardisierung der Datenerfassung bei manueller Eingabe zu beachten ist. Sobald eine
21 Eingabe mehrere Varianten für die gleiche Aussage zulässt, werden Auswertungen erschwert
22 und somit eine effiziente Serviceverbesserung verhindert. ITIL (2013) unterscheidet drei Typen
23 von Messgrößen: Technologie-, Prozess- und Servicemessgrößen, wobei jede Messgröße den
24 Fokus auf eine andere Ebene legt.

25 **Verarbeiten von Daten**

26 In diesem Schritt werden die erfassten Daten aufbereitet und in Berichten zusammengefasst. Der
27 Aufbau der Berichte ist vom Empfänger des Berichts abhängig. ITIL (2013) empfiehlt die erfassten
28 Daten einzelner Komponenten zusammen zu fassen und die Daten aus der Kundensicht zu
29 verarbeiten.

30 **Analysieren der Informationen und Daten**

31 Die Analyse der verarbeiteten Daten versucht eindeutige Trends zu identifizieren und diese zu
32 bewerten, um daraus erforderliche Changes abzuleiten. Die Analyse wird für Prognosen über die
33 Zielerreichung verwendet und gibt Aufschluss darüber, ob und wo Verbesserungen erforderlich
34 sind. Den Kern in der Datenanalyse sieht ITIL (2013) in der richtigen Interpretation der Ergebnisse
35 von Trendanalysen – vor allem in Bezug auf die Ursache und die resultierenden Konsequenzen.

36 **Präsentieren und Nutzen der Informationen**

37 Bei der Präsentation der gewonnenen Informationen, auch Service Reporting genannt, werden die
38 analysierten Daten so aufbereitet, dass dem Empfänger die erforderlichen Maßnahmen

1 präsentiert werden. Bei der Präsentation der Informationen ist es relevant, dass der
2 Empfängerkreis bekannt ist und die Daten so aufbereitet werden, dass der Empfängerkreis den
3 Nutzen dahinter sieht. ITIL (2013) unterscheidet vier Typen von Empfängern: Lieferanten,
4 Servicekunden, interne IT, oberes IT Management. Jeder dieser Typen legt sein
5 Hauptaugenmerk beim Service Reporting anders und erwartet andere Ergebnisse.

6 **Implementieren der Verbesserungsmaßnahmen**

7 Nachdem Verbesserungsmaßnahmen im vorangegangenen Schritt aufgezeigt wurden, müssen
8 diese bewertet und priorisiert werden. Sobald eine positive Entscheidung über die
9 Implementierung einer Verbesserungsmaßnahme gefällt wurde, wird der Service Lebenszyklus
10 fortgesetzt, indem die Service Strategie angepasst wird, ein Change in der Phase Service Design
11 erstellt wird, in der Phase Service Transition in den Betrieb überführt wird und der Phase Service
12 Operations für den Betrieb übergeben wird. (Great Britain: Cabinet Office, 2013)

13 Nach der Identifikation des Verbesserungspotentials und der Definition der
14 Verbesserungsaktivität wird anhand eines Business Cases überprüft, ob die Kosten für die
15 identifizierte Verbesserung für den daraus resultierenden geschäftlichen Nutzen gerechtfertigt
16 sind.

17 **3.6.2 Kurzzusammenfassung**

18 Um kontinuierliche Verbesserung innerhalb von Prozessen und Services zu erreichen, ist eine
19 genaue und permanente Messung erforderlich. Durch genaues Messen wird einerseits die
20 Möglichkeit geboten, Verbesserungspotentiale zu identifizieren und andererseits gezeigt, welche
21 Auswirkungen Aktivitäten zur Verbesserung erwirkt haben. Der 7-Schritte-Verbesserungsprozess
22 bildet einen wichtigen Bestandteil im CSI und ermöglicht die Identifikation von
23 Verbesserungsmöglichkeiten und das Verständnis darüber, welche Leistungsindikatoren in den
24 jeweiligen Prozessen und Services relevant sind, um nachhaltige Verbesserungen zu
25 ermöglichen.

1 **4 ANSÄTZE**

2 Die Beschreibung der IT Infrastructure Library zeigt klar, dass der Fokus auf die genaue Planung
3 und somit möglichst realitätsnahe Vorhersage von zukünftigen Ereignissen ausgerichtet ist.
4 Dadurch wird versucht, ein hohes Maß an Systemstabilität zu erzeugen. Bevor ein geplanter
5 Service oder eine geplante Änderung in die Produktivumgebung überführt wird, müssen viele
6 Stationen durchlaufen werden. Dadurch wird die Lead Time verlängert. Die Lead Time nach Mary
7 und Tom Poppendieck (2006) bezeichnet die Zeitspanne zwischen der Identifizierung der
8 Serviceidee oder dem Entschluss eine Verbesserung umzusetzen und der Nutzbarkeit des neuen
9 Service oder der Verbesserung. In dieser Wartezeit kann der Service oder die Verbesserung nicht
10 genutzt werden, wodurch potenzielle Vorteile für das Unternehmen ungenutzt bleiben. Um die
11 Geschwindigkeit der Leistungserbringung innerhalb ITIL orientierten Organisationen zu erhöhen,
12 werden Bestandteile unterschiedlicher Ansätze untersucht.

13 Um passende Ansätze zu identifizieren, werden im Zuge einer Literaturrecherche die Begriffe
14 Agile und Lean IT näher betrachtet.

15 **4.1 Agile**

16 Der Begriff Agile ist in der Literatur als Sammlung von Prinzipien beschrieben und hat seinen
17 Ursprung in der Softwareentwicklung. Das Ziel agiler Softwareentwicklung ist die Erhöhung der
18 Flexibilität im Vergleich zu starren klassischen Vorgehensmodellen. Als Beispiel für starre
19 klassische Vorgehensmodelle dienen das Wasserfall- und das darauf aufbauende V-Modell.
20 Beide Modelle gliedern die Softwareentwicklung in Phasen, deren Ergebnisse jeweils die
21 Vorgaben für die nachfolgenden Phasen bilden. Der Umfang und die zu erreichenden Ergebnisse
22 sind bei beiden Modellen zu Beginn festgelegt.

23 Das Manifest für Agile Softwareentwicklung (Beck, et al., 2001) nennt vier zentrale Werte, stellt
24 sie direkt vier anderen Werten gegenüber und priorisiert diese.

25 **„Individuen und Interaktionen** mehr als Prozesse und Werkzeuge

26 **Funktionierende Software** mehr als umfassende Dokumentation

27 **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung

28 **Reagieren auf Veränderung** mehr als das Befolgen eines Plans“

29 Es wird ausdrücklich darauf hingewiesen, dass die Werte auf der rechten Seite als wichtig
30 einzuschätzen sind, jedoch den Werten auf der linken Seite eine höhere Priorität beigemessen
31 wird.

32 Ergänzend zu diesen Werten werden 12 Prinzipien als Grundlage für das Agile Manifest genannt.

33 Die Prinzipien können wie folgt zusammengefasst werden: (Beck, et al., 2001)

- 34
- frühe und kontinuierliche Auslieferung wertvoller Software

- 1 • Änderungen der Anforderungen sind während der gesamten Entwicklungszeit
2 willkommen
- 3 • funktionierende Software soll in regelmäßigen Abständen von wenigen Wochen oder
4 Monaten geliefert werden
- 5 • Experten aus dem Fachbereich und Entwickler arbeiten täglich zusammen
- 6 • Bereitstellung des Umfelds und der Unterstützung, welche von motivierten Personen für
7 die Aufgabenerfüllung benötigt wird
- 8 • der effektivste Informationsaustausch findet im direkten Gespräch statt
- 9 • funktionierende Software ist das wichtigste Fortschrittsmaß
- 10 • die Arbeitsgeschwindigkeit soll so gewählt werden, dass sie sowohl vom
11 Entwicklungsteam als auch vom Auftraggeber konstant gehalten werden kann
- 12 • ständiges Augenmerk auf technische Exzellenz und gutes Design
- 13 • Einfachheit in der Entwicklung ist essenziell
- 14 • selbstorganisierte Teams bei der Erstellung von Architekturen, Entwürfen und
15 Anforderungen
- 16 • regelmäßige Retrospektiven, um die Arbeit effektiver zu gestalten

17 Die Einhaltung dieser Prinzipien und der oben genannten Werte soll den Nutzen für den Kunden
18 erhöhen.

19 Agile selbst ist kein Vorgehensmodell, sondern dient als Mindset und somit als Basis für eine
20 Vielzahl von Vorgehensmodellen, die diese Prinzipien umsetzen und nach den beschriebenen
21 Werten ihre Prioritäten setzen. Vorgehensmodelle, die auf die agilen Werte und Prinzipien
22 aufsetzen, bedienen sich oftmals mehrerer Methoden, die auf Agilität ausgerichtet sind. Beispiele
23 hierfür sind die Verwendung von Story-Cards, die die Anforderungen der Anwender aus
24 Anwendersicht dokumentieren, oder das von Kent Beck (2003) geprägte Test-Driven-
25 Development (TDD), welches Testfälle bereits vor der Entwicklung der jeweiligen
26 Softwarekomponente erstellt.

27 Beispiele für Vorgehensmodelle, die agilen Werten und Prinzipien folgen, sind Feature Driven
28 Development, eXtreme Programming und Scrum.

29 **4.2 Scrum**

30 Dieses Kapitel befasst sich mit den Grundlagen von Scrum und gibt einen Überblick über die
31 Merkmale und gängigen Begrifflichkeiten.

32 Scrum erfreut sich großer Beliebtheit und zählt zu den bekanntesten Vorgehensmodellen, welche
33 agilen Prinzipien folgen. Das Vorgehensmodell selbst kennzeichnet ein Gebilde, welches einen
34 starken prozessualen Ablauf verfolgt. Aus diesem Grund ist neben den synonym verwendeten

1 Bezeichnungen Scrum-Flow, Scrum-Methode, Scrum-Framework auch der Terminus Scrum-
2 Prozess gebräuchlich.

3 Der Begriff Scrum wurde nach Schwaber (1997) von den Japanern Nonaka und Takeuchi
4 geprägt, die den Begriff als Analogie für sehr erfolgreiche Teams in der Produktentwicklung
5 verwendet haben. Später haben Ken Schwaber und Keff Sutherland, die zu den Autoren des
6 Agilen Manifestes zählen, diesen in die Softwareentwicklung übernommen, wodurch der Scrum-
7 Guide entstanden ist. Der Scrum-Guide ist in über 40 verschiedenen Sprachen verfügbar,
8 beschreibt in 18 Seiten die Merkmale des Scrum-Prozesses und wird kontinuierlich
9 weiterentwickelt.

10 Die Ausführungen des Scrum Guides (Schwaber & Sutherland, 2017, S. 3) dienen als Definitions-
11 und Beschreibungsgrundlage, wonach Scrum *„ein Rahmenwerk, innerhalb dessen Menschen*
12 *komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt*
13 *werden, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern“* ist, welches
14 einfach zu verstehen, aber schwierig zu meistern ist.

15 Der Scrum Guide beschreibt Scrum mit Hilfe von drei Rollen, unterscheidet fünf Ereignisse und
16 kennt drei Artefakte.

17 **4.2.1 Rollen**

18 Der Scrum Guide (2017) definiert die drei Rollen Product Owner, Entwicklungsteam und Scrum
19 Master. Zudem wird von Scrum der Begriff Scrum-Team verwendet, welcher die drei Scrum-
20 Rollen vereinigt und deutlich machen soll, dass dieses Team als Einheit selbstständig und
21 unabhängig von anderen Teams arbeiten kann.

22 **Product Owner:** Der Product Owner in Scrum vertritt als Einzelperson die Sicht des
23 Auftraggebers sowie der Stakeholder und ist für die Aktualität und Richtigkeit des Product
24 Backlogs sowie die Priorisierung innerhalb des Product Backlogs verantwortlich. Der Product
25 Owner definiert somit *was* in Umsetzung kommt.

26 **Scrum Master:** Der Scrum Master ist für die Einhaltung des Scrum Prozesses zuständig und
27 kümmert sich um die Moderation von später beschriebenen Scrum Ereignissen. Er identifiziert
28 und beseitigt Störungen, die das Team bei der Erledigung ihrer Aufgaben behindern. Damit
29 übernimmt er die Rolle eines Servant Leaders, der seine Dienste nach Greenleaf und Spears
30 (1998) dienend an den Interessen des Teams ausrichtet. Der Scrum Master definiert *wie* die
31 Umsetzung prozessual abläuft.

32 **Entwicklungsteam:** Das Entwicklungsteam besteht in der Regel aus fünf bis zehn Personen und
33 setzt sich interdisziplinär zusammen. Neben Architekten, Entwicklern und Testern können auch
34 weitere, je nach Aufgabenstellung erforderliche, Themenfelder vertreten sein, wobei es wichtig
35 ist herauszustreichen, dass Scrum keine weitere Unterteilung innerhalb des Entwicklungsteams
36 kennt. Ein Entwicklungsteam ist selbstorganisiert und entscheidet eigenständig, wie die Aufgaben
37 innerhalb der später beschriebenen zeitlichen Rahmen (Sprints) erledigt werden.

1 Während die Rolle des Entwicklungsteams von mehreren Personen wahrgenommen wird, sind
2 die Rollen Product Owner und Scrum Master nur von einzelnen Personen wahrzunehmen. Um
3 Interessenskonflikte zu vermeiden, ist es gängige Praxis, dass Product Owner und Scrum Master
4 nicht im Entwicklungsteam vertreten sind und keine Personalunion bilden.

5 Organisationen, denen Teamgrößen von bis zu zehn Personen sowie Einzelpersonen als Product
6 Owner und Scrum Master nicht genügen, können Erweiterungen, wie das von Larman und Vodde
7 (2010) beschriebene Large Scale Scrum (LeSS), nutzen. LeSS beschreibt Ansätze, die durch
8 den Einsatz eines übergeordneten Produkt-Owneers und mehreren Bereichs-Product-Ownern die
9 verteilte Arbeit mehrerer Scrum-Teams am selben Produkt ermöglichen.

10 **4.2.2 Artefakte**

11 Artefakte sind greifbare Nebenprodukte, die während der Produktentwicklung erstellt werden. Als
12 Beispiele werden die im Scrum Guide (2017) angeführten Artefakte Product Backlog, Sprint
13 Backlog und Inkrement genannt und nachfolgend überblicksmäßig beschrieben.

14 **Product Backlog:** Der Product Backlog ist eine sich ständig ändernde Auflistung aller
15 Anforderungen, die der Product Owner in Vertretung der Stakeholder an das Produkt richtet. Der
16 Product Backlog beinhaltet neben Features und Funktionalitäten auch Verbesserungen und
17 Fehlerbehebungen. Einträge im Product Backlog enthalten neben dem Titel und der
18 Beschreibung auch eine Reihenfolge und eine grobe Aufwandsschätzung, die relativ zu anderen
19 Einträgen berechnet ist. Je detaillierter ein Eintrag spezifiziert wird, desto höher steigt der Eintrag
20 im Product Backlog nach oben und desto früher wird dieser abgearbeitet. Der Product Owner,
21 der die Verantwortung über die Verfeinerung und die Aktualität des Product Backlogs hat, kann
22 somit steuern, welche der Funktionalitäten höher gereiht werden. Die obersten Einträge im
23 Product Backlog sollten so detailliert beschrieben sein, dass das Entwicklungsteam diese
24 während eines Sprints umsetzen kann. Detailliert beschriebene Einträge enthalten neben der
25 Beschreibung oft eine Definition of Done (DoD) und eine Beschreibung von Tests, die die
26 Fertigstellung überprüfen. Neben der Definition von Anforderungen dient der Product Backlog als
27 Informationsgrundlage, um Gesamtfortschrittsprognosen zu erstellen.

28 **Sprint Backlog:** Ein Sprint Backlog wird vorab für jeden Sprint erstellt und enthält ausgewählte
29 Einträge des Product Backlogs, die im Laufe des Sprints fertiggestellt werden sollen. Der Sprint
30 Backlog beschreibt die Funktionalitäten, die im kommenden Inkrement enthalten sein sollen und
31 welcher Aufwand hierfür erforderlich ist. Während des Sprints dient der Sprint Backlog als stets
32 aktuelles Abbild der momentan ausstehenden Arbeit. Es obliegt dem Entwicklungsteam die
33 Schätzung des erforderlichen Aufwands je nach Fortschritt anzupassen und erforderliche neue
34 Arbeitspakete hinzuzufügen oder obsoletere Arbeitspakete zu entfernen. Der Sprint Backlog dient
35 als Basis für die Überwachung des Sprintfortschritts, welche im Zuge des später beschriebenen
36 Daily Scrum Meetings erfolgt. Ein essenzieller Punkt, um den Fortschritt messen zu können, ist
37 die Unterscheidung zwischen nicht begonnenen, in Arbeit befindlichen und fertiggestellten
38 Arbeitspaketen. Um diese Unterscheidung eindeutig treffen zu können, beinhalten die
39 Beschreibungen der Arbeitspakete bereits eine Definition, ab wann das Arbeitspaket als

1 „fertiggestellt“ angesehen kann. Hierfür wird der englische Begriff Definition of Done (DoD)
2 verwendet.

3 **Inkrement:** Ein Inkrement beschreibt die Summe der während des Sprints fertiggestellten
4 Einträge des Sprint Backlogs sowie aller zuvor erstellten Inkremente. Ein Inkrement muss sich
5 am Ende eines Sprints in einem an den Kunden auslieferbaren Zustand befinden und somit
6 verwendbar sein.

7 **4.2.3 Ereignisse**

8 Innerhalb des Scrum Guides (2017) werden fünf Ereignisse unterschieden, auf die in den
9 kommenden Absätzen überblicksmäßig eingegangen wird.

10 **Sprint:** Projekte, die dem Scrum Prozess folgen, werden in inkrementellen Fortschritten
11 entwickelt. Die Zeitspanne, in der ein derartiges Inkrement entwickelt wird, wird Sprint genannt.
12 Das Ergebnis jedes Sprints ist die Entwicklung eines potenziell auslieferungsfähigen
13 Produktinkrements. Die Dauer, die für die Erstellung des Produktinkrements zur Verfügung steht,
14 ist auf eine feste Zeitdauer fixiert (timeboxed) und beträgt üblicherweise zwischen zwei bis vier
15 Wochen. Rubin (2014) identifiziert den gleichmäßigen, regelmäßigen Rhythmus in der Scrum-
16 Entwicklung – die Kadenz – als wesentlichen Faktor, um den Aufwand für die Planung und
17 Koordination der Sprints gering zu halten. Jeder Sprint startet umgehend nachdem der
18 vorangegangene Sprint abgeschlossen ist. Der Sprint selbst beinhaltet neben der
19 Entwicklungsarbeit weitere Scrum Ereignisse, wie Sprint Planning, Daily Scrum, Sprint Review
20 und Sprint Retrospektive.

21 **Sprint Planning:** Das Sprint Planning ist zeitlich zu Beginn jedes Sprints und dient zur Definition
22 der Ziele, die innerhalb des Sprints erreicht werden sollen. Für die Durchführung des Sprint
23 Planning Meetings sind maximal, unabhängig von der Dauer des Sprints, 8 Stunden vorgesehen.
24 Während des Sprint Plannings werden Einträge des Product Backlogs ausgewählt, die innerhalb
25 des Sprints abgeschlossen werden können und mit einem konkreten Umsetzungsplan in den
26 Sprint Backlog übertragen werden.

27 **Daily Scrum:** Nach der erfolgreichen Planung des Sprints startet die Entwicklungsarbeit für die
28 restliche Dauer des Sprints. Analog zu den anderen Ereignissen ist auch das Daily Scrum
29 timeboxed und findet regelmäßig statt. Für das Daily Scrum sind täglich 15 Minuten vorgesehen,
30 bei dem die Arbeitsergebnisse des Vortags geprüft und die Entwicklungen der nächsten 24
31 Stunden geplant werden. Das Daily Scrum ist ein Ereignis für das Entwicklungsteam, bei dem in
32 Bedarfsfällen weitere Personen, wie beispielsweise der Product Owner, teilnehmen können.
33 Innerhalb dieses täglichen Zeitfensters werden potenzielle Hindernisse besprochen, die das
34 Weiterkommen behindern oder das Sprintziel gefährden.

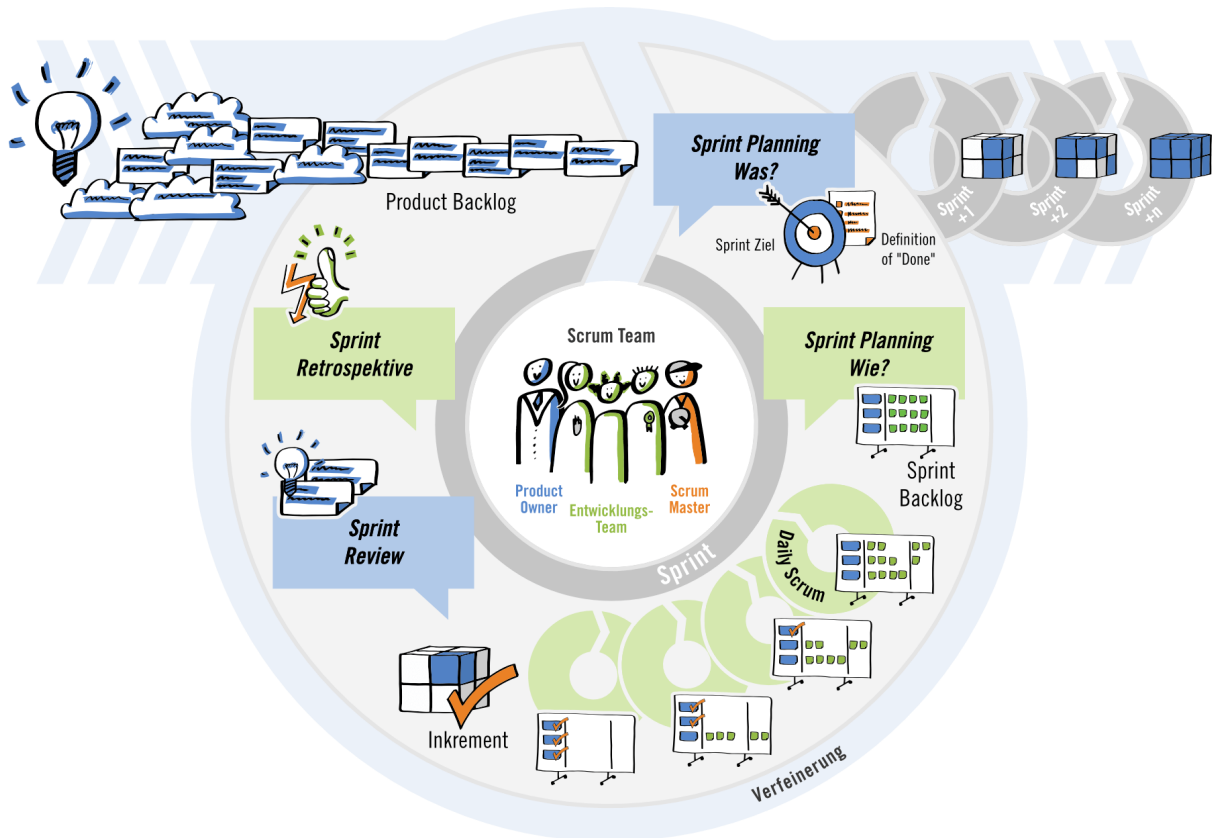
35 **Sprint Review:** Am Ende eines Sprints werden die im Sprint Planning gesetzten Ziele überprüft
36 und der Product Backlog entsprechend dem Fortschritt angepasst. Ergänzend dazu wird dieses
37 Meeting dazu verwendet, um Feedback einzuholen und den Wert des Produkts für den
38 Auftraggeber zu erhöhen. Diese Tätigkeiten werden gemeinsam mit dem Scrum Team und den

1 Stakeholdern informell durchgeführt und als Sprint Review bezeichnet. Die vorgegebene timebox
 2 für das Sprint Review Meeting sieht eine Maximaldauer von vier Stunden vor.

3 **Sprint Retrospektive:** Ebenfalls am Ende eines Sprints wird eine Sprint Retrospektive
 4 durchgeführt. Ähnlich zum Sprint Review handelt es sich um eine vergangenheitsbezogene
 5 Betrachtung des vorangegangenen Sprints. Im Unterschied zum Sprint Review, dessen Fokus
 6 auf dem Produkt selbst liegt, wird in der Sprint Retrospektive die Konzentration auf die
 7 Arbeitsweise gelegt, mit deren Hilfe das Produkt entstanden ist. Neben der Betrachtung von
 8 Entwicklungspraktiken und Prozessen wird allgemein nach Verbesserungen im prozessualen
 9 Ablauf gesucht, um die Arbeit effektiver oder effizienter zu gestalten. Die identifizierten
 10 Verbesserungen werden als Basis für die kommenden Sprints verwendet. Die timebox für die
 11 Sprint Retrospektive hat eine Maximaldauer von drei Stunden.

12 **4.2.4 Zusammenfassung**

13 Scrum ist ein prozessähnliches Vorgehensmodell, welches auf Basis von agilen Werten arbeitet
 14 und den Fokus auf wertvolle Ergebnisse für den Kunden legt.



15
 16 *Abbildung 9: Der Scrum Zyklus (DasScrumTeam AG, 2018)*

17 In Abbildung 9 wird der Scrum Zyklus visuell dargestellt und zeigt, dass der Produktentwicklung
 18 eine Auflistung von Anforderungen – dem Backlog – zu Grunde liegt. Diese Anforderungen
 19 werden in mehreren Sprints iterativ abgearbeitet, wodurch nach jedem Sprint ein weiteres
 20 Inkrement des Endprodukts entsteht. Die Abbildung veranschaulicht weiters, dass den Kern

1 dieses Modells die Personen des Scrum Teams bilden. Innerhalb des Sprints folgt auf das Sprint
2 Planning die Entwicklungszeit, in welcher unter täglicher Abstimmung Aufgaben abgeschlossen
3 werden, die zusammengefasst das Produktinkrement beschreiben. Der Sprint endet mit den
4 beiden Ereignissen Sprint Review und Sprint Retrospektive, die zur Produkt- und
5 Ablaufverbesserung dienen.

6 Die Vorgehensweise von Scrum zeigt, dass der Fokus auf die Leistungserbringung in kurzen
7 zyklischen Abständen gesetzt wird, die im Zuge dieser Arbeit als unterstützend angesehen wird.
8 Scrum behandelt den prozessualen Ablauf der Entwicklung wertvoller Software und integriert
9 erfolgreich agile Prinzipien. Zeitgleich geht Scrum von Rahmenbedingungen aus, die eine hohe
10 Anzahl an Deployments ermöglichen.

11 **4.3 Lean Management**

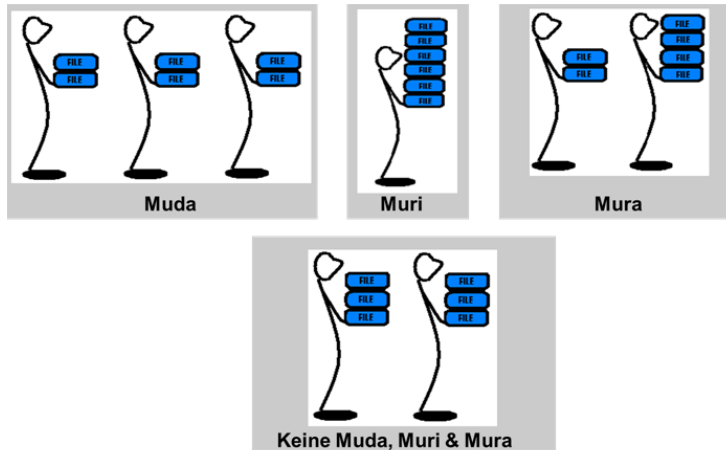
12 Der Begriff lean im Kontext von Lean Management ist in der Literatur durch eine Sammlung von
13 Prinzipien und Methoden beschrieben. Eine allgemein gültige Definition existiert laut Kammel &
14 Groth (1994) nicht.

15 Ursprünglich leitet sich der Begriff Lean Management von Lean Manufacturing ab, welches als
16 Produktionsstrategie der damals vorherrschenden Massenproduktion entgegengesetzt wurde. Im
17 Gegensatz zur Massenproduktion verfolgte Lean Manufacturing das Ziel, eine beliebige
18 Produktvielfalt ohne Lagerbestände und ohne Produktionsfehler zu erreichen. (Womak, Jones, &
19 Roos, 1992)

20 Nach der Definition von Wallace Hopp und Mark Spearman (2011) kann eine Fertigungsstraße
21 im Sinne von Lean Manufacturing dann als *lean* angesehen werden, wenn die definierten Ziele
22 mit minimalen Pufferkosten erreicht werden können. Das impliziert, dass eine derartige
23 Fertigungsstraße stets unter voller Nutzung der vorhandenen Ressourcen die exakt benötigte
24 Menge, ohne Nacharbeit durch Qualitätsprobleme, ohne Wartezeit für den Kunden, ohne
25 Verspätung, ohne Rohmaterial- und Materiallager und mit der geringstmöglichen Anzahl
26 laufender Arbeiten produziert.

27 Diese Fertigungsstraße ist die Vision, die Lean Manufacturing verfolgt.

28 Lean hat somit als zentrales Ziel die Vermeidung von Tätigkeiten, die dem Endkunden keinen
29 Mehrwert bieten.



- 1
- 2 *Abbildung 10: Muda, Muri & Mura (Hanschke, 2015)*
- 3 Taiichi Ōno (1988) beschreibt die drei Übel eines Produktionssystems mit den japanischen
- 4 Begriffen Muda, Muri, Mura. Inge Hanschke gibt mit ihrer in Abbildung 10 dargestellten Illustration
- 5 einen Überblick über die drei Übel des Produktionssystems. Muda steht für die Verschwendung
- 6 und bezeichnet den Umstand, dass der gleiche Nutzen für den Kunden mit geringerem
- 7 Ressourcenaufwand erbracht werden kann. Muri bezeichnet die Überlastung von Ressourcen
- 8 und Mura bezeichnet unausgeglichene Auslastung von Ressourcen. Da die drei Begriffe Muda,
- 9 Muri und Mura teilweise eine Ursachen-Wirkungs-Beziehung aufweisen, ist es ein Ziel, im Lean
- 10 Management, die Beseitigung von Verschwendung zu erreichen, ohne dabei Unausgeglichenheit
- 11 oder Überlastung zu generieren.
- 12 Die nachfolgenden Unterkapitel beschreiben die Verschwendung im engeren Sinn (Muda) und
- 13 geben einen groben Überblick über Mura und Muri.

14 **4.3.1 Verschwendung (Muda)**

15 Die Verschwendung (Muda) bezeichnet grundsätzlich Tätigkeiten, die Ressourcen erfordern,

16 ohne einen Mehrwert für den Kunden zu bieten und kann in zwei Bereiche unterteilt werden. Auf

17 der einen Seite steht die notwendige Verschwendung, deren Auftreten nahezu nicht verhindert,

18 sondern nur deren Notwendigkeit bestmöglich reduziert werden kann. Auf der anderen Seite steht

19 die offensichtliche Verschwendung, welche grundsätzlich vermieden werden kann. Des Weiteren

20 können 7 Arten der Verschwendung unterschieden werden, aus denen sich das Akronym

21 TIMWOOD ergibt. Die Buchstaben stehen jeweils für den Anfangsbuchstaben der von Taiichi

22 Ōno (1988) geprägten Verschwendungsarten der Produktion (in Englischer Sprache). Mary und

23 Tom Poppendieck haben (2003) analog dazu die sieben Verschwendungsarten der

24 Softwareentwicklung beschrieben. Nachfolgend sind die sieben Verschwendungsarten nach Ōno

25 beschrieben und mit Analogien zur Softwareentwicklung versehen.

26 **Transport** (Transport): Der Transport bezeichnet Verschwendung, die beim Transport von Waren

27 oder Betriebsmitteln auftritt. Ebenfalls fallen unnötige Wege von Informationen und Unterlagen in

28 diese Kategorie. In der Softwareentwicklung wird die ähnliche Problematik mittels Task Switching

1 beschrieben. Task Switching erzeugt Unterbrechungen und stört den Arbeitsfluss des
2 Mitarbeiters.

3 **Inventory** (Lagerbestände): Die auftretende Verschwendung durch Lagerung von Waren und
4 Materialien. Hier treten einerseits Kosten für die Lagerung selbst auf, als auch Kosten durch die
5 Bindung von Kapital. Die Analogie aus der Softwareentwicklung ist ein teilweise fertiggestelltes
6 Arbeitspaket, welches anders als in der Produktion kein Kapital bindet, jedoch (Entwicklungs-)
7 Ressourcen bindet, ohne dass ein Mehrwert erzeugt wird.

8 **Motion** (Bewegung): Verschwendung tritt bei unnötiger Bewegung von Menschen auf, die aus
9 suboptimaler Anordnung von Betriebsmitteln entstehen kann und so zu ineffizienten
10 Bewegungsabläufen führt. Analog dazu geht es in der Softwareentwicklung um die effiziente
11 Bereitstellung von nötigen Informationen.

12 **Waiting** (Wartezeiten): Wartezeiten bezeichnen Verschwendungen, die durch nicht zeitgerechte
13 Verfügbarkeit von Materialien, Maschinen oder Informationen entstehen. In der
14 Softwareentwicklung bezieht sich das Warten häufig auf Freigaben, die Fertigstellung von Builds,
15 Starten von Systemen oder die Fertigstellung exzessiver Anforderungsdokumentationen.

16 **Overprocessing** (Übererfüllung): Die Übererfüllung beschreibt überflüssige oder nicht
17 angemessene Abläufe und Tätigkeiten. In der Softwareentwicklung wird hingegen die
18 Verschwendung mit Zusatzprozessen beschrieben. Beispiele hierfür sind die Notwendigkeiten
19 von papierbasierten Abnahmeprotokollen, papiergetriebenen Freigaben von Changes oder
20 anderen Dokumenten, die zum Selbstzweck erstellt werden und nur einen geringen Mehrwert
21 erbringen.

22 **Overproduction** (Überproduktion): Überproduktion hat zur Folge, dass Investitionen für die
23 Erstellung eines Produkts aufgewendet werden, obwohl kein Bedarf des Kunden identifiziert
24 wurde. Umgelegt auf die Softwareentwicklung bedeutet das, dass Funktionalitäten in die Software
25 integriert werden, obwohl der Bedarf noch nicht besteht. Dies hat zur Folge, dass neben höherem
26 Implementierungsaufwand auch Verschwendung durch höheren Testaufwand geschieht.

27 **Defects** (Fehler): Die unter dem Begriff Defects zusammengefasste Verschwendung beschreibt
28 Fehlerbehebungen und Nacharbeiten auf Basis von Fehlern, die vor oder während der
29 Herstellung entstanden sind. In dieser Art der Verschwendung unterscheidet sich die
30 Softwareentwicklung kaum von der Produktion. Ein plakatives Beispiel aus der
31 Softwareentwicklung sind falsch verstandene Anforderungen und die damit verbundenen
32 Änderungsaufwände.

33 **4.3.2 Unausgeglichenheit (Mura)**

34 Mura bezeichnet Verluste, die durch Unausgeglichenheit entstehen. Dies kann durch fehlende
35 Abstimmung zwischen den Produktionssteuerungen oder durch zu hohe
36 Auslastungsschwankungen auftreten. In der Softwareentwicklung verlagert sich dies auf die
37 fehlende Abstimmung innerhalb des Entwicklungsteams und durch Schwankungen, die in der
38 Aufgabenverteilung auftreten.

1 **4.3.3 Überlastung (Muri)**

2 Muri bezeichnet Verluste, die durch die über die natürlichen Grenzen hinausgehende Belastung
3 entstehen. Die Überlastung von Maschinen kann höheren Verschleiß oder höhere
4 Wartungskosten mit sich bringen, oder in Wartezeiten (eine der Verschwendungen in Muda)
5 resultieren. Folgen von Überbelastung bei personellen Ressourcen äußern sich in erhöhter
6 Fehleranfälligkeit, Stress, Verschlechterung des Betriebsklimas und höherer Unfallgefahr. Im
7 Kontext der Softwareentwicklung können hierdurch Fehler entstehen oder fehlerhafte
8 Schnittstellen die Systemintegration behindern.

9 **4.3.4 Prinzipien**

10 Lean IT versucht die Prinzipien von Lean Manufacturing in das Umfeld der
11 Informationstechnologie zu transferieren.

12 Umgelegt auf die IT bedeutet das, dass es essenziell ist, die Konzentration auf die für den Kunden
13 wesentlichen Leistungen zu legen und dabei auf eine geringstmögliche Anzahl und Dauer von
14 sinnlosen Tätigkeiten zu achten.

15 Die fünf Grundprinzipien von Lean Thinking nach Womak und Jones (2013) lauten:

16 **Spezifikation des Werts:** Dieses Prinzip beschreibt die genaue Definition des zu erbringenden
17 Werts für den Kunden. Das kann die Spezifikation eines Produkts, eines Service, eines
18 Projektergebnisses oder eines Geschäftsprozesses sein.

19 **Identifikation des Wertstroms:** Der Weg, der bei der Erbringung des spezifizierten Werts
20 durchlaufen wird, muss identifiziert werden. Der Aufwand für die Identifikation des Wertstroms
21 erhöht sich bei Auslagerung von Dienstleistung oder Zukauf von Materialien. Die Schritte entlang
22 der Wertschöpfung müssen klar und so transparent sein, dass potenzielle Verschwendung
23 erkannt und beseitigt werden kann.

24 **Flow:** Das Prinzip des Flows ermöglicht einen kontinuierlichen Arbeitsfluss bei der
25 Wertschöpfung, der ohne Unterbrechungen, Ausschuss oder Rückflüsse in Richtung Kunden
26 voranschreitet. Hierzu kann es nötig sein, Schnittstellen zu reduzieren oder die Abfolge von
27 Prozessschritten zu verändern. Die kontinuierliche Bearbeitung steht hier im Gegensatz zur
28 Stapelverarbeitung, die in nach Funktionen oder Abteilungen strukturierten Unternehmen typisch
29 sind.

30 **Pull:** Das Pull-Prinzip zielt darauf ab, nur dann zu arbeiten, wenn Bedarf besteht. Es wird auf
31 Überproduktion und Lagerung verzichtet. Übertragen auf die IT, gibt dieses Prinzip den
32 Fachabteilungen die Möglichkeit, neue Anforderungen an IT-Services zu kommunizieren, anstatt
33 ausschließlich zu entscheiden, ob sie von der IT-Abteilung vorab bereitgestellte Services (Push-
34 Prinzip) einsetzen möchten oder nicht.

35 **Perfektion:** Dieses Prinzip beschreibt das ständige Streben nach Perfektion. Um der Perfektion
36 näher zu kommen, ist eine kontinuierliche Verbesserung der Umsetzung der oben genannten
37 Prinzipien erforderlich.

1 **4.3.5 minimum viable product**

2 Das Lean Gedankengut ist bereits in mehrere Bereiche übertragen worden. So auch in den
3 Bereich der Startups. Eric Ries beschreibt (2011) im Zusammenhang mit Startups das Prinzip
4 des „minimum viable product“ (MVP). MVP beschreibt ein schnell erstelltes Produkt, welches mit
5 der nötigsten Funktionalität ausgestattet ist, um möglichst früh Feedback vom Markt zu
6 bekommen. Hierdurch wird der ebenfalls beschriebene Build-Measure-Learn Zyklus initiiert,
7 durch den es ermöglicht wird, mit geringstmöglichem Aufwand den größtmöglichen Lernerfolg zu
8 erzielen.

9 Im Kontext der Softwareentwicklung kann dieser Ansatz durch die Entwicklung schneller
10 Prototypen dazu genutzt werden, um durch häufiges und frühzeitiges Feedback des Kunden eine
11 Lösung mit höchstmöglichem Wert für den Kunden zu erreichen. Hierdurch wird das Prinzip der
12 Perfektion verfolgt.

13 **4.4 DevOps**

14 DevOps selbst ist kein kodifizierter Standard, sondern ist eher als Bewegung zu sehen. DevOps
15 kann im Kern als die Verbindung zwischen dem Development (Softwareentwicklung) und
16 Operations (Systemadministration und Betrieb) gesehen werden. Den Ursprung hat DevOps in
17 der Umstellung auf agile Softwareentwicklung. In der agilen Softwareentwicklung wird das
18 Augenmerk mitunter auf die Lieferung potenziell auslieferbarer Software in kurzen regelmäßigen
19 Intervallen gelegt (Gloger, 2016). Die zyklische Erstellung von funktionierender, potenziell
20 auslieferbarer Software erfordert eine Organisation, die ein Ausliefern von Software in kurzen
21 zyklischen Abständen unterstützt. DevOps behandelt hierbei den typischen Zielkonflikt zwischen
22 Development (maximale Flexibilität) und Operations (maximale Stabilität und Verfügbarkeit) mit
23 Hilfe einer Sicht auf den gesamten Wertstrom, der im Business beginnt und sich sowohl über
24 Development als auch Operations erstreckt.

25 DevOps unterstützt bei der Übertragung von agilen Methoden auf den IT Betrieb und ermöglicht
26 eine engere Zusammenarbeit zwischen Development und Operations. Dadurch werden in
27 weiterer Folge über den gesamten Softwarebereitstellungsprozess idente Verfahren verwendet.

28 **4.4.1 Prinzipien**

29 Die DevOps Agile Skills Association (2017) fasst sechs Prinzipien, die DevOps zu Grunde liegen,
30 wie folgt zusammen:

31 **Customer-Centric Action:** DevOps Teams und Organisationen erfordern höchste Flexibilität im
32 Umgang mit Kundenanforderung. Das hat zur Folge, dass die Anzahl von Feedbackzyklen zu
33 Endanwendern und Auftraggebern erhöht und die Zeit zwischen den einzelnen Feedbackzyklen
34 verringert werden muss. Jegliche Innovation innerhalb des DevOps Teams oder der Software
35 wird an den Bedürfnissen des Kunden ausgerichtet.

1 Dieses Prinzip deckt sich mit dem Prinzip der Kundenorientierung aus Agile und erfüllt das Prinzip
2 der Definition des Werts aus Lean.

3 **Create with End in Mind:** DevOps versucht, die prozessorientierte Vorgehensweise zu einer
4 produktorientierten Vorgehensweise weiter zu entwickeln. Das mündet in dem Ergebnis, dass die
5 Mitarbeiter des DevOps Teams gemeinsam die Verantwortung am funktionierenden Produkt
6 tragen und die klassische Rollen- oder Funktionssicht weitgehend ablegen. Daraus resultiert,
7 dass die Mitarbeiter sich auf die Entwicklung von funktionierenden Produkten konzentrieren, die
8 an den Endkunden geliefert werden. Das bedeutet ebenfalls, dass ein Team die Verantwortung
9 für das gesamte Aktivitäts-, Funktions- und Prozessspektrum innerhalb des Produkts trägt.
10 Hierdurch wird der Blick für das Produkt als Ganzes geschärft.

11 Dieses Prinzip implementiert die Idee der Identifizierung des Wertstroms aus Lean.

12 **End-to-End Responsibility:** Das Prinzip der Ende-zu-Ende Verantwortung stellt bewusst das
13 Ende des klassischen hand-over Prozesses zwischen Software-Entwicklung und Software-
14 Betrieb dar und bedeutet, dass dem DevOps Team eine Verantwortung von der Konzeption über
15 die Entwicklung und den Betrieb bis hin zur Ablöse der Lösung übertragen wird.

16 Amazon vertritt den Standpunkt „You build it, you run it“ (Vogels, 2006), der ebenfalls die Ende-
17 zu-Ende Verantwortung beschreibt.

18 Die Ende-zu-Ende Verantwortung überschneidet sich mit dem Grundsatz der autonomen Teams
19 aus Agile, die alle notwendigen Kompetenzen zur Aufgabenerfüllung haben und erfüllt das Prinzip
20 des Flows aus Lean.

21 **Cross-Functional Autonomous Teams:** Im Gegensatz zu klassischen Organisationen, in
22 denen Lösungen von einer Abteilung konzipiert, von einer weiteren Abteilung umgesetzt und zu
23 einer weiteren Abteilung in den Betrieb übergeben werden, verfolgt DevOps den Ansatz, dass
24 Teams nicht prozessorientiert organisiert, sondern funktionsübergreifend aufgebaut werden.
25 Durch diesen vertikalen Aufbau entstehen selbst-organisierte und autonom agierende Teams.

26 Der selbst-organisierte, vertikale Aufbau des Teams spiegelt sich in den Prinzipien des Agile
27 Manifests wider und unterstützt den in Lean geforderten Flow.

28 **Continuous Improvement:** Funktionsübergreifende, autonome Teams mit Ende-zu-Ende
29 Verantwortung erfordern eine permanente Evaluierung von technischen Rahmenbedingungen,
30 Kundenanforderungen sowie Gesetzen und Normen, um geänderte Rahmenbedingungen zu
31 erkennen. Kontinuierliche Verbesserung beschreibt die Fähigkeit einerseits adäquat auf die
32 erkannten, geänderten Rahmenbedingungen reagieren zu können und andererseits
33 Verschwendung zu vermeiden, Kosten zu senken und das gelieferte Produkt ständig zu
34 verbessern. Um dies zu erreichen ist eine Änderung in der Fehlerkultur erforderlich, die darauf
35 abzielt, aus Fehlern zu lernen, anstatt diese als etwas Negatives zu betrachten.

36 Kleinaltenkamp, Plinke & Geiger haben diesen Umstand wie folgt zusammengefasst:

37 *„Man lernt nicht nur aus Erfolgen, sondern vor allem aus den vielen Unebenheiten, Konflikten und*
38 *Barrieren, gescheiterten Maßnahmen, etc.“ (2013, S. 341)*

1 Die geänderte Fehlerkultur geht nach Mike Rother (2013) so weit, dass Vorgesetzte (von Rother
2 Mentoren genannt) ihre Mitarbeiter (von Rother Mentees genannt) absichtlich kleinere Fehler
3 begehen lassen, solange diese den Kunden nicht erreichen, um bestmöglich daraus zu lernen.

4 **Automate everything you can:** Dieses Prinzip setzt auf die Automatisierung weitgehend aller
5 Tätigkeiten. Die Automatisierung umfasst einerseits den Entwicklungsprozess im Sinne des
6 Build-, Test- und Deployment-Prozesses und andererseits die automatisierte Erstellung von der
7 zur Entwicklung nötigen Infrastruktur. Je nach Reifegrad wird auf unterschiedliche Sammlungen
8 von Techniken zurückgegriffen. Diese beginnen mit Continuous Integration, erstrecken sich über
9 Continuous Delivery bis hin zu Continuous Deployment. Continuous Integration ermöglicht, dass
10 die Arbeit mehrerer unterschiedlicher Entwickler mehrmals am Tag in den gemeinsamen
11 Entwicklungsfortschritt (Trunk) eingecheckt und getestet wird. Continuous Delivery erweitert
12 Continuous Integration um die Abbildung von automatisierten Akzeptanztests. Continuous
13 Deployment ergänzt Continuous Delivery um die automatisierte Verteilung in das
14 Produktivsystem. Um diesen hohen Automatisierungsgrad zu ermöglichen, ist eine hohe Anzahl
15 an automatisierten Tests in unterschiedlicher Ausprägungsstufe erforderlich. Um diese Tests
16 durchführen zu können, ist die Bereitstellung der dafür erforderlichen IT Infrastruktur notwendig.
17 Um diese IT Infrastruktur in einem hohen Automationsgrad bereitstellen zu können, bedient sich
18 DevOps Techniken wie Infrastructure as Code (IaC). Mit IaC soll der manuelle Aufwand zur
19 Erstellung einer Entwicklungs- und Testumgebung minimiert und gleichzeitig die
20 Übereinstimmung zwischen Produktiv-, Entwicklungs- und Testumgebung maximiert werden.

21 Dieser Grundgedanke implementiert die Idee der Vermeidung von Verschwendung aus Lean und
22 ermöglicht einen verbesserten Flow.

23 **4.4.2 CAMS**

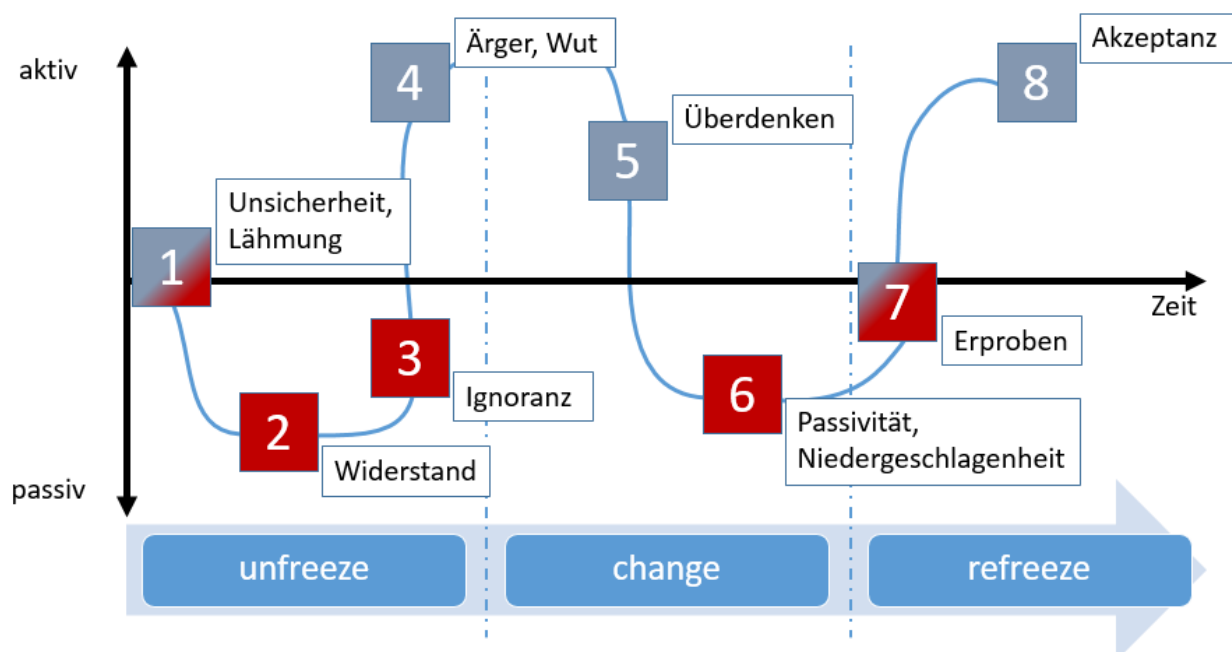
24 Bereits im Jahr 2010 wurde auf den DevOpsDays in Mountain View das Akronym CAMS von
25 Damon Edwards und John Willis geprägt, welches die DevOps zugrunde liegenden Werte
26 beschreibt. CAMS steht für die Begriffe Kultur (culture), Automation (automation), Messbarkeit
27 (measurement) und dem Teilen (sharing). Die einzelnen Bestandteile des Akronyms CAMS sind
28 nachfolgend beschrieben.

29 **4.4.2.1. Culture**

30 Um die Vorteile aus DevOps nutzen zu können, ist ein Kulturwandel erforderlich, der eine
31 gemeinsame geteilte Verantwortung zwischen Developer und Operations in der
32 Softwarebereitstellung vorsieht. Im Speziellen übertragen auf das ITIL Framework bedeutet das,
33 dass ein Kulturwandel geschehen muss, sodass Personen aus dem Operations Umfeld in die
34 Projektteams integriert werden und Vertreter aus den Development Teams in regelmäßiger
35 Abstimmung mit Operations stehen. Bei der Bearbeitung von Incidents sollen Developer direkt
36 bei der Analyse und Behebung der Störfälle, sofern nötig, unterstützen. (Humble & Molesky,
37 2011)

1 Um eine Kultur zu schaffen, ist eine weitreichende Änderung des menschlichen Verhaltens
 2 notwendig, was Change Management im betriebswirtschaftlichen Sinne erforderlich macht. Vahs
 3 und Weiland (2010) sehen Change Management als *"die Vorbereitung, Analyse, Planung,
 4 Realisierung, Evaluierung und laufende Weiterentwicklung von ganzheitlichen
 5 Veränderungsmaßnahmen mit dem Ziel, ein Unternehmen von einem bestimmten Ist-Zustand zu
 6 einem erwünschten Soll-Zustand weiterzuentwickeln und so die Effizienz und Effektivität aller
 7 Unternehmensaktivitäten nachhaltig zu steigern. Das Management des Wandels bewertet damit
 8 die aktuellen Potenziale und Fähigkeiten einer Organisation und plant systematisch die
 9 notwendigen Veränderungsschritte"*. (Vahs & Weiland, Workbook Change Management:
 10 Methoden und Techniken, 2010, S. 7)

11 Innerhalb eines derartigen Change Prozesses ist damit zu rechnen, dass während der drei
 12 Phasen des Changes nach Lewin, die als „unfreezing“, „changing“ und „refreezing“ von Mayer,
 13 Walt, Gleich & Wagner (2008) beschrieben werden, auch die acht Phasen der emotionalen
 14 Reaktionen nach Becker und Labucay (2012) durchlaufen werden.



15
 16 *Abbildung 11: 8 typische Reaktionen in den 3 Phasen der Veränderung (in Anlehnung an Becker & Labucay (2012))*

17 Abbildung 11 zeigt die acht Phasen der emotionalen Reaktionen in Zuordnung zur Phase der
 18 Veränderung. Zudem wird mit Hilfe der farblichen Markierung der Grad der aktiven Reaktion
 19 angezeigt, wobei die grau gekennzeichneten Bereiche aktive Reaktionen beschreiben und die
 20 rote Kennzeichnung für eine passive Reaktion steht. Die acht Reaktionen gliedern sich in eine
 21 Abfolge von Unsicherheit, Widerstand, Ignoranz und Wut während der „unfreezing“ Phase, eine
 22 Erkenntnis des Nutzens und Passivität in der „change“ Phase sowie das aktive Umsetzen der
 23 Änderung und der Akzeptanz und die dauerhafte Praktizierung in der „refreezing“ Phase. Wie
 24 allgemein in Change Management Projekten, ist es auch bei Kulturänderungen im DevOps
 25 Umfeld ratsam, die sieben Grundtypen der Veränderung zu betrachten. Nach Vahs (2007)
 26 gliedern sich diese in die folgenden Typen auf: Visionäre & Missionare, Aktive Gläubige,

1 Opportunisten, Abwartende & Gleichgültige, Untergrundkämpfer, offene Gegner, Emigranten.
2 Diese sieben Typen sind meistens in einer der Normalverteilung ähnelnden Form in
3 Organisationen anzutreffen, wobei der Anteil an Visionären & Missionaren sowie Emigranten als
4 sehr gering einzustufen ist und der größte Anteil der Belegschaft die Position der Abwartenden &
5 Gleichgültigen einnimmt.

6 **4.4.2.2. Automation**

7 Die Automatisierung des Build-, Test- und Deployment-Prozesses dient als Schlüssel für die
8 Vermeidung von Fehlern, Verkürzung der Durchlaufzeit und der zeitnahen Sammlung wichtigen
9 Kundenfeedbacks. Dies kann erreicht werden, indem in einer Deployment Pipeline gearbeitet
10 wird, die für alle Änderungen am System einen einzelnen Weg in die Produktivumgebung
11 bereitstellt. Jegliche Änderung am System, unabhängig des Typs, beispielsweise Konfiguration,
12 Datenbank, Code oder Infrastruktur, durchläuft dieselben Stadien von automatisierten
13 Komponententests auf den Entwicklungssystemen, bevor diese in ebenso automatisierten
14 Integrationstests in der Testumgebung in die Qualitäts- und Produktionsumgebung überführt wird.
15 Die Überführungen (Deployments) werden ebenfalls automatisiert und können auf Knopfdruck
16 gestartet werden. Wenn von Deployments gesprochen wird, sind nicht ausschließlich
17 Änderungen am Code gemeint, sondern ebenso Änderungen an der Infrastruktur oder sonstigen
18 Rahmenbedingungen. Selbst die Erstellung und Einrichtung der Entwicklungsumgebung sollte
19 automatisiert werden, wodurch auf Knopfdruck die richtige Infrastruktur virtuell bereitgestellt wird.
20 (Humble & Molesky, 2011)

21 Für die Automatisierung des Build-, Test- und Deployment-Prozesses wird im Umfeld von
22 DevOps auf Continuous Integration gesetzt. Continuous Integration (CI) ist als eine der Praktiken,
23 die Kent Beck (2000) in der Softwareentwicklungsmethode „eXtreme Programming“ beschrieben
24 hat, bekannt geworden. CI beschreibt, dass jeder abgeschlossene Task eines einzelnen
25 Softwareentwicklers mehrmals täglich in den Gesamtsoftwarestrang (Trunk) integriert wird und
26 sich somit in der Codebasis aller anderen Entwickler widerspiegelt. Das erfordert, dass die Dauer
27 einer Integration kurzgehalten werden muss.

28 Hiermit wird laut Wiest (2011) das Ziel verfolgt, dass die Integration aller Codeänderungen am
29 Ende eines Softwareprojekts den gleichen überschaubaren Gefahren- und Komplexitätsgrad
30 annimmt, der auch bei der mehrmals täglichen Integration auftritt. Somit stellt die Integration am
31 Ende des Projekts lediglich einen weiteren, gut geprobten Vorgang innerhalb des
32 Entwicklungszyklus dar.

33 Martin Fowler (2006) ist der Ansicht, dass für die Einführung von erfolgreichem CI zehn Praktiken
34 vorausgesetzt werden.

35 1.) Gemeinsame Codebasis:

36 Die gemeinsame Codebasis ist die Grundlage und setzt ein Versionsverwaltungssystem
37 voraus.

38 2.) Automatisierter Build:

1 Es muss ohne zusätzlichen manuellen Aufwand möglich sein, einen neuen Build zu
2 erstellen. Die Übersetzung des Codes, sowie das Verlinken der Komponenten sind
3 Bestandteile des Builds. Dessen Automatisierung bringt einerseits
4 Geschwindigkeitsvorteile und andererseits eine Reduktion der möglichen Fehlerquellen.

5 3.) Erschaffen eines selbsttestenden Builds:

6 Ein erfolgreicher Build bestätigt nur, dass das Programm funktionstüchtig ist, nicht aber
7 dass das Programm richtig arbeitet. Hierfür ist es notwendig, dass innerhalb des
8 Buildprozesses statische Tests enthalten sind.

9 4.) Tägliche Integration:

10 Mindestens einmal pro Tag integrieren die Entwickler ihre Änderungen am Code in die
11 Versionskontrolle. Dieser Vorgang wird auch als „einchecken“ oder „commit“ bezeichnet.

12 5.) Umgehende Behebung fehlgeschlagener Builds:

13 Nach jedem Einchecken in die Versionskontrolle geschieht ein automatisierter Build und
14 eine automatisierte Testung des Builds. Hierdurch wird erreicht, dass Fehler in
15 Änderungen schnell aufgedeckt werden und entsprechend schnell behoben werden
16 können. Schlägt nach dem Einchecken einer Änderung ein Build fehl, liegt die Priorität
17 an der Behebung dieses Umstandes.

18 6.) Kurze Testdurchlaufzeiten:

19 Kernpunkt von CI ist die Bereitstellung schneller Rückmeldungen. Um diese schnellen
20 Rückmeldungen zu ermöglichen, muss die Durchlaufzeit bei automatisierten Tests so
21 gering wie möglich gehalten werden. Es muss daher sorgsam abgewogen werden,
22 welche Tests im Zuge einer Integration durchgeführt werden müssen und welche im
23 Anschluss an die Integration laufen können.

24 7.) Tests in gespiegelter Produktivumgebung:

25 Um die Testergebnisse so repräsentativ wie möglich zu halten, ist es ratsam, eine
26 Testumgebung aufzubauen, die der Produktivumgebung so weit wie möglich gleicht.
27 Das beinhaltet neben der identen Version von Datenbank- und Betriebssystemen auch
28 eine nahezu idente Infrastruktur. Um das zu erreichen, bieten sich virtuelle Umgebungen
29 an.

30 8.) Einfacher Zugriff auf Build- und Testergebnisse:

31 Die aktuellste Version der Software sollte nicht nur allen beteiligten Softwareentwicklern
32 zur Verfügung stehen, sondern ebenso allen anderen involvierten Personen im
33 Softwareprojekt.

34 9.) Transparenz:

35 Jeder im Projektteam sollte über den Zustand der Software und des Build Bescheid
36 wissen. Nicht nur die einzelnen Ergebnisse eines Buildvorgangs, sondern auch
37 historische Daten, wie beispielsweise die Dauer, die ein Build in einem nicht

1 funktionsfähigen Zustand war, sollten dargestellt werden. Das kann beispielsweise durch
2 automatisierte Visualisierung erreicht werden.

3 10.) Automatisierte Verteilung:

4 In CI werden unterschiedliche Systeme für unterschiedliche Tests verwendet. Ein
5 System bearbeitet Tests, die beim Einchecken ausgeführt werden, andere Systeme
6 bilden nachgelagerte Tests ab. Um den Vorgang effizient zu gestalten, ist es erforderlich,
7 dass auch die Verteilung der Software über die verwendeten Systeme automatisiert wird.

8
9 DevOps berücksichtigt diese Praktiken und erhöht nochmals den Automatisierungsgrad, indem
10 auf Continuous Delivery (CD) gesetzt wird.

11 Im Unterschied zu Continuous Integration, ist in Continuous Delivery die Software nach jedem
12 erfolgreichen Check-In in einem an den Endkunden auslieferungsfähigen Zustand und kann auf
13 Knopfdruck in die Produktivumgebung übertragen werden. Dies erfordert weitreichende
14 Automatisierung, wodurch neben Integrations-Tests auch Akzeptanztests automatisiert
15 abgebildet werden. (Humble & Farley, 2010)

16 Als Erweiterung von Continuous Delivery ist Continuous Deployment (CDe) anzusehen, wobei
17 hier selbst das Deployment in die Produktivumgebung automatisiert ist und Änderungen somit
18 noch schneller dem Endkunden zur Verfügung gestellt werden. (Fitz, 2009)

19 Ein weiterer, in DevOps häufig Anwendung findender Schritt zur Erhöhung des
20 Automatisierungsgrades ist die Verwendung von Infrastructure as Code (IaC). Mit IaC soll erreicht
21 werden, dass der manuelle Aufwand zur Erstellung einer Entwicklungs- und Testumgebung
22 minimiert wird und gleichzeitig die Übereinstimmung zwischen Produktiv- Entwicklungs- und
23 Testumgebung maximiert wird. Dadurch wird erreicht, dass Tests aussagekräftiger werden und
24 einen tatsächlichen Rückschluss auf das Verhalten von Software in der Produktivumgebung
25 erlauben (Guckenheimer, 2018). Der Code für die Erstellung der (virtuellen) IT Infrastruktur wird
26 im Versionsverwaltungssystem eingecheckt und steht somit, wie andere Codeartefakte der
27 Software, jedem Entwickler zur Verfügung.

28 **4.4.2.3. Measurement**

29 Durch die Aufzeichnung und richtige Definition von aussagekräftigen Kennzahlen werden die
30 aktuelle Leistung und potenzielle Engpässe transparent dargestellt. Diese können die Basis für
31 weitere Verbesserungen bilden. DevOps Messgrößen können in drei Bereiche unterteilt werden:
32 (Riley, 2015)

- 33 • personenbezogene Messgrößen
- 34 • prozessbezogene Messgrößen
- 35 • technologiebezogene Messgrößen

36 Viele Messgrößen umfassen mehr als nur eine der genannten Bereiche.

1 Ein Beispiel einer interessanten Messgröße könnte die Dauer zwischen vorliegenden
2 Anforderungen bis hin zur funktionierenden Software sein.

3 Weitere Beispiele von potenziell interessanten Messgrößen, die auf mögliche Probleme innerhalb
4 des DevOps Prozesses hinweisen, sind die Gesamtdurchlaufzeit eines Changes (Lead Time)
5 oder die durchschnittliche Dauer zwischen dem Eintritt eines Fehlers und der Wiederherstellung
6 des Service (Mean Time to Recover MTTR). Aufgrund der gestiegenen Häufigkeit von
7 Deployments gewinnt auch die Fehlerrate bei Deployments an Bedeutung.

8 Messgrößen für den Erfolg von DevOps können die Häufigkeit von Deployments, die Dauer eines
9 Deployments und die Mitarbeiterzufriedenheit innerhalb des DevOps Teams sein.

10 **4.4.2.4. Sharing**

11 Hinter dem Begriff Sharing im Umfeld von DevOps verbirgt sich die Möglichkeit, von
12 vorhandenem Wissen und vorhandener Erfahrung im Unternehmen zu profitieren und die eigene
13 Erfahrung breiter zu streuen. Sharing umfasst die Art und Weise wie Informationen zwischen den
14 Personen über die Team- und Funktionsgrenzen hinweg innerhalb der Organisation geteilt
15 werden. Beispielhaft können paarweise Quellcodeüberprüfungen (Peer-Code Reviews),
16 Großbildschirme mit Statusinformationen (Information Radiators) und andere Methoden genannt
17 werden, die eine Feedbackschleife etablieren, die mehrere Personen oder Teams umfassen. Zu
18 den zu teilenden Informationen und Erfahrungen zählen ebenso Fehler, wobei der Fokus nicht
19 auf dem Verteilen von Informationen über den Verursacher des Fehlers liegt, sondern darauf,
20 dass das erneute Auftreten des Fehlers vermieden werden kann. Um dieses Wissen
21 gewinnbringend zu streuen, ist eine Kultur erforderlich, die offen mit Fehlern umgeht und
22 Lösungen anstatt Schuldige sucht.

23 Das Hernstein Institut hat (2017) eine Umfrage mit über 1500 Führungskräften und Unternehmern
24 durchgeführt. Die Teilnehmer der Umfrage, die aus unterschiedlichen Unternehmen in Österreich
25 und Deutschland stammen, wurden zum Thema Kultur im Unternehmen und Umgang mit Fehlern
26 befragt. Laut des daraus resultierenden Berichts sehen über 30% der befragten Personen des
27 unteren Managements das Problem, dass im Unternehmen meist Dauerstress bestehe, wodurch
28 keine Zeit zum Lernen aus den eigenen Fehlern bleibe. Aus demselben Bericht geht hervor, dass
29 über 30% der befragten Führungskräfte die Kultur im Unternehmen nicht als „lösungsorientiert“,
30 „fehlerverzeihend“, „aus Fehlern lernend“, „von Vertrauen geprägt“ oder „Kooperation fördernd“
31 bezeichnen. Daraus ergibt sich, dass in Unternehmen in Österreich und Deutschland noch
32 Potential zur Verbesserung vorhanden ist.

33 Mit Sharing ist nicht nur das Teilen von Wissen und der Wissenstransfer zwischen Development
34 und Operations gemeint, sondern auch das Teilen gemeinsamer Erfolge. Auch das Teilen im
35 Sinne der gemeinsamen Verwendung von Tools und Techniken ist ein wichtiger Aspekt.

1 **4.5 Zusammenfassung und Hypothese**

2 Die ausgewählten Betrachtungsweisen von Agile und Lean bieten Potential, die
3 Leistungserbringung in an ITIL orientierten IT-Organisationen zu erhöhen, betrachten aber die
4 Stabilität der IT Systeme zu wenig. Scrum, als eines der bekanntesten agilen Vorgehensmodelle,
5 ermöglicht eine erhöhte Flexibilität betreffend sich ändernder Anforderungen, setzt für den
6 effektiven Einsatz aber die unkomplizierte Möglichkeit von zyklischen Deployments in kurzen
7 Abständen voraus. Die DevOps Bewegung baut auf die Prinzipien und Werte des Agilen
8 Manifests auf und implementiert Ideen aus Lean. DevOps setzt sich zum Ziel, die
9 Zusammenarbeit und den Wissensaustausch zwischen den Bereichen Entwicklung und Betrieb
10 zu verbessern. Durch den verbesserten Wissensaustausch, den hohen Grad der Automatisierung
11 und die frühe Einbindung von IT Operations wird auch der Aspekt der Systemstabilität betrachtet.

12 Durch das breitere Spektrum von DevOps, welches neben der Softwareentwicklung auch den
13 Betrieb von Systemen abdeckt, bietet sich DevOps als Ansatz an, um die Beschleunigung der
14 Leistungserbringung bei gleichzeitiger Beibehaltung der gewonnen Systemstabilität zu erreichen.
15 DevOps setzt sehr stark auf die Automatisierung wiederkehrender Tasks und ermöglicht mittels
16 durchgängiger Verantwortung ein verbessertes Verständnis für den Nutzen, den die Anwendung
17 für das Kerngeschäft des Unternehmens erbringt.

18 Aus den betrachteten Ansätzen leitet sich die folgende Hypothese ab:

19 H1: „Die gezielte Integration von Ansätzen aus DevOps, Lean und Agile in Arbeitsabläufe, die an
20 ITIL angelehnt sind, kann die Geschwindigkeit der Leistungserbringung erhöhen, ohne die
21 Stabilität des Systems negativ zu beeinflussen.“

1 **5 BESCHLEUNIGUNG DER LEISTUNGSERBRINGUNG**

2 In Kapitel 3 sind die Best Practices eines Service Lebenszyklus nach dem IT Service
3 Management Rahmenmodell ITIL beschrieben, die in vielen Unternehmen die Grundlage für die
4 Definition der gelebten Prozesse bilden. ITIL verfolgt das Ziel einer hohen Systemstabilität.
5 Dieses Ziel wird teilweise, wie in den Beobachtungen in Kapitel 2 beschrieben, auf Kosten der
6 Flexibilität oder der Geschwindigkeit der Leistungserbringung erreicht. In Kapitel 4 sind Ansätze
7 beschrieben, die ihr Augenmerk auf die Erhöhung der Flexibilität bei der Umsetzung von
8 Anforderungen legen.

9 Die nachfolgenden Punkte beschäftigen sich mit der Kombination an ITIL angelehnten
10 Prozessen, die auf Systemstabilität ausgelegt sind und mit Ansätzen, die den Fokus auf
11 Flexibilität und Geschwindigkeit legen.

12 Da in DevOps einerseits Werte und Prinzipien von Lean und Agile integriert werden und
13 andererseits eine Mitbetrachtung des IT Betriebs vorgesehen ist, wird DevOps für die
14 Kombination mit ITIL gewählt.

15 Zu Beginn wird in Form einer Online-Umfrage evaluiert, ob die Vorteile, die eine Integration von
16 Ansätzen aus von DevOps verspricht, für Organisationen mit an ITIL angelehnten Prozessen
17 relevant sind. Im Anschluss daran werden zwei Anwendungsfälle betrachtet, die in der
18 Beobachtung in Kapitel 2 einen großen Einfluss auf die Geschwindigkeit der Leistungserbringung
19 hatten. In beiden Anwendungsfällen werden Lösungsansätze beschrieben, welche durch
20 Integration von Methoden, Techniken und Herangehensweisen aus DevOps die Geschwindigkeit
21 der Leistungserbringung erhöhen können, ohne die durch ITIL gewonnene Stabilität maßgeblich
22 negativ zu beeinflussen.

23 Die beschriebenen Anwendungsfälle verzichten bewusst auf die Integration des Konzepts der
24 Bereitstellung von Entwicklungs-, Test- und Qualitätsumgebungen mittels IaC. Der beobachtete
25 Arbeitsablauf betrifft Erweiterungen an den beiden monolithischen Plattformen Salesforce und
26 SharePoint, die bereits alternative Ansätze zur Bereitstellung von Entwicklungs-, Test- und
27 Qualitätsumgebungen bieten.

28 **5.1 Überprüfung der Relevanz**

29 Um zu prüfen, ob die Vorteile, die eine Integration von Ansätzen aus DevOps verspricht, für
30 Organisationen mit an ITIL angelehnten Prozessen relevant sind, wird eine quantitative Methode,
31 ein Online-Fragebogen, gewählt.

32 Der Online-Fragebogen war vom 14.05.2017 bis zum 30.10.2018 zur Beantwortung
33 freigeschaltet. Die Zielgruppe der Befragung bildeten Mitarbeiter und Mitarbeiterinnen von
34 Unternehmen, die in ITIL orientierten IT Organisationen im Wertstrom der Leistungserbringung
35 tätig sind. Beispiele dieser Zielgruppe sind: Angestellte im Bereich der Softwareentwicklung,
36 Systemadministratoren und Systemadministratorinnen, Mitarbeitende im Supportbereich,

1 Vertreterinnen und Vertreter aus dem Bereich der IT-Governance sowie Mitglieder der
2 strategischen und operativen IT-Leitung. Um möglichst passende Testpersonen zu finden, wurde
3 der Link zum Fragebogen an IT- und IT-Projektmanagement Abteilungen sowie Bereiche im
4 Umfeld der Softwareentwicklung österreichischer Firmen versandt. Zusätzlich wurde der Link
5 zum Fragebogen in sozialen Medien geteilt und in Gruppen im Softwareentwicklungsumfeld
6 innerhalb dieser sozialen Medien beworben. Um die Anzahl an vollständigen Beantwortungen
7 hoch zu halten, wurde darauf geachtet, dass einfache Formulierungen verwendet werden und die
8 Länge des Fragebogens einen entsprechenden Wert nicht übersteigt. Umfragen mit mehr als 15
9 bis 25 geschlossenen Fragen sind laut Gräf (1999, S. 157ff) bereits zu lange. Zusätzlich wurde
10 das Augenmerk daraufgelegt, dass die Beantwortung der Fragen einfach gestaltet ist und
11 überlange Fragestellungen vermieden werden.

12 Insgesamt beinhaltet der Fragebogen zehn Fragen, wobei zwei zur Sicherstellung der Zielgruppe
13 dienen. Der Fragebogen besteht aus drei Ja/Nein Fragen und sieben geschlossenen Fragen, die
14 sich in zwei Auswahlfragen mit einfacher Auswahloption und fünf Fragen, die durch Zuhilfenahme
15 der Likert-Skala die Zustimmung der Probanden erfragen. Die Ja/Nein Fragen wurden so
16 gewählt, dass auf die ITIL Prozesse der Phase Service Transition abgezielt wird, die
17 geschlossenen Fragen mit fünfstufiger Likert-Skala (1,„trifft gar nicht zu“ bis 5,„trifft vollkommen
18 zu“) dienen der Untersuchung auf Relevanz der dargebotenen Problematik. Auf die Abfrage von
19 demografischen Informationen wurde bewusst gänzlich verzichtet, um die Einfachheit aufrecht zu
20 erhalten.

21 Der Online-Fragebogen wurde 24 Mal ausgefüllt und davon zu über 90% vollständig beantwortet.
22 Die Teilnehmenden der Umfrage sind gleichmäßig zwischen richtungsweisender IT (IT
23 Management und IT Governance) und operativer IT (IT Architekt, Anwendungsentwicklung, IT
24 und Anwendungssupport) verteilt und arbeiten mehrheitlich in Unternehmen mit mehr als 250
25 Angestellten. Zwei Drittel der Befragten geben an, dass ein Change Prozess im Unternehmen
26 verankert ist. Etwa jede zweite Person hat die Frage nach der Existenz eines Prozesses zur
27 Autorisierung und Inbetriebnahme von Softwareprodukten mit Ja beantwortet.

28 Die geschlossenen Fragen wurden wie folgt beantwortet:

29 *Manuelle Anwendungstests im Entwicklungs- Qualitäts- und Produktivsystem erzeugen*
30 *einen hohen personellen Aufwand wodurch tägliche Softwarereleases verhindert werden.*

31 Ø 4,22 von 5

32 *Aus der strikten Trennung zwischen Entwicklungsabteilung und dem IT-Betrieb entstehen*
33 *Kommunikationsprobleme, die langfristig erhöhte Kosten im Betrieb der Anwendung nach*
34 *sich ziehen.*

35 Ø 3,83 von 5

36 *Mehrstufige Deployment-Freigabeprozesse verhindern die schnelle Auslieferung von*
37 *neuen Funktionen.*

38 Ø 3,83 von 5

1 *Fehlendes Verständnis für den Gesamtnutzen der Anwendung im Kerngeschäft des*
2 *Unternehmens führt zu Fehlentscheidungen in der Konzeption, Entwicklung und im*
3 *Betrieb der Anwendung.*

4 Ø 4,52 von 5

5 *Sich schnell ändernde Marktanforderungen erfordern das Verständnis über die zu*
6 *unterstützenden Kerntätigkeiten des Kunden, um schnelle Änderungen an Lösungen*
7 *bereitstellen zu können.*

8 Ø 4,14 von 5

9 Aus den Ergebnissen der Umfrage ist ersichtlich, dass agilere Ansätze und Ideen aus DevOps
10 für Unternehmen mit an ITIL angelehnten Prozessen von Relevanz sind.

11 Auf Basis dieser Ergebnisse werden in den nachfolgenden Unterkapiteln Ideen und Methoden
12 aus DevOps integriert und in einer Anpassung des beobachteten Arbeitsablaufs beschrieben.
13 Weiteres werden Lösungsansätze aufgezeigt, um den identifizierten Problemfeldern aus den
14 beobachteten Abläufen entgegen zu wirken.

15 **5.2 Anwendungsfall Anforderungsdefinition**

16 Die Anforderungsdefinition dient zur vollständigen Dokumentation der Anforderungen. Die
17 Ergebnisse dienen als Grundlage für Aufwands- und Kostenschätzungen und für die
18 Entscheidung, ob die Umsetzung der Lösung oder der Änderung erfolgreich war. Die
19 Anforderungsdefinition dient zudem als Grundlage für Detailkonzepte im Bereich der IT
20 Infrastruktur der Softwarearchitektur, die als Entscheidungsbasis für die Freigabe durch die IT
21 Sicherheitsabteilung dient.

22 **5.2.1 Problembeschreibung**

23 Eine vollständige Anforderungsdefinition, die vor Beginn der Umsetzung erfolgt, geht davon aus,
24 dass alle Anforderungen bereits bekannt sind oder gemeinsam mit dem Ansprechpartner des
25 Auftraggebers erarbeitet werden können. In einigen Fällen, wie auch im beobachteten Fall, sind
26 nicht alle Anforderungen vorab bekannt. Daraus resultiert eine lange Durchlaufzeit, da die
27 Anforderungen erst gemeinsam mit dem Auftraggeber erarbeitet werden müssen. Ebenfalls steigt
28 durch den sequenziellen Ablauf das Risiko, fehlende oder falsch verstandene Anforderungen erst
29 spät im Projektverlauf zu erkennen und entsprechend darauf reagieren zu können.

30 **5.2.2 Lösungsansatz**

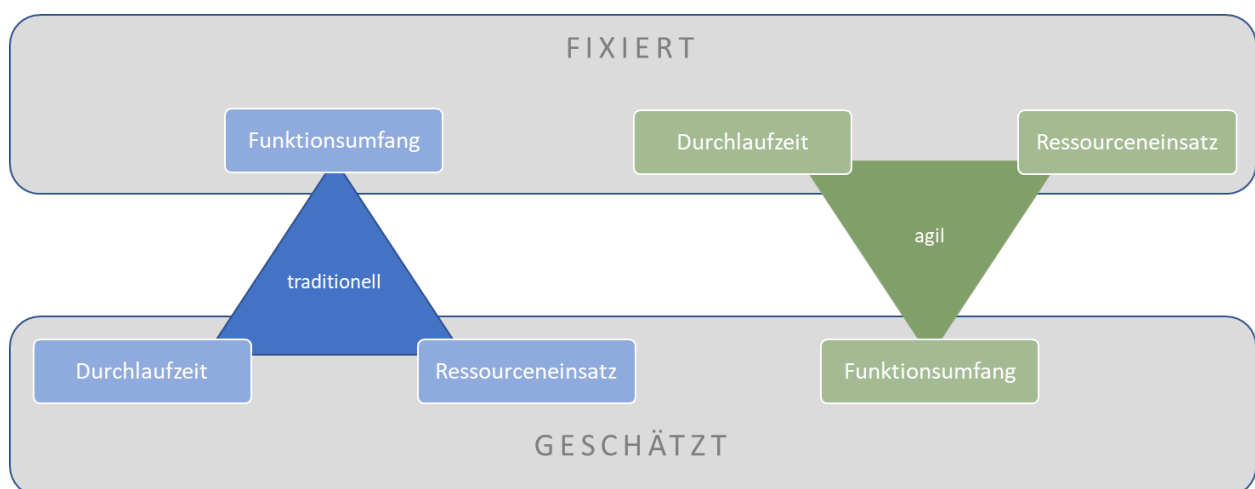
31 Das Umsetzungsteam, welches aus den Erfordernissen, die im Service Charter dokumentiert
32 sind, zusammengestellt wird, erfüllt das DevOps Prinzip der cross-functional-teams. Bereits zu
33 Beginn wird darauf geachtet, dass das Projektteam den Wert für den Auftraggeber versteht und

1 dem Prinzip „create with end in mind“ nachgegangen wird. Anders als in typischen Scrum Teams
 2 vorgesehen, sind auch Experten aus dem Bereich des Service Operations Teammitglieder.
 3 Die Wertgenerierung erfolgt in zeitlich begrenzten Zyklen, die, wie aus dem aus Scrum bekannten
 4 Konzept der Sprints, zu Beginn vom Team inhaltlich und strukturell geplant und direkt im
 5 Anschluss umgesetzt werden. In die Umsetzungskonzepte fließen die Erfahrungen der
 6 Teammitglieder des Bereichs Service Operations ein, die bereits im Projektteam integriert sind.
 7 Die initiale Anforderungsdefinition wird so schlank wie möglich gehalten, um die Zeitspanne
 8 zwischen dem Start der Anforderungserhebung und dem Start der Entwicklung der Lösung gering
 9 zu halten. Das Hauptaugenmerk wird daraufgelegt, schnellstmöglich einen Prototypen („minimum
 10 viable product“) für den Auftraggeber erstellen zu können. Der Prototyp dient als Gesprächs- und
 11 Feedbackgrundlage. Die Rückmeldungen des Auftraggebers fließen in die Priorisierung der
 12 Anforderungen ein. Hierdurch entsteht eine Überlappung der Phasen Service Design und Service
 13 Transition. Die Anforderungen, die aus den Rückmeldungen des Auftraggebers auf Basis des
 14 Prototyps entstehen, werden, wie aus Scrum bekannt, in einem Product Backlog dokumentiert
 15 und priorisiert. Der Product Backlog dient als Grundlage für die iterativen Erweiterungen des
 16 Prototyps. Die Anforderungsdefinition wird somit von einem vormals vorgelagerten Arbeitsschritt
 17 zu einem wiederkehrenden Schritt innerhalb des Ablaufs der Wertschöpfung transformiert.

18 5.2.3 Erfordernisse und Voraussetzungen

19 Die Umsetzung dieses Lösungsansatzes erfordert die Erfüllung des Prinzips der „End-To-End
 20 Responsibility“ und wird durch eine Personalunion des Verantwortlichen der Phasen Service
 21 Design und Service Transition erleichtert. Um eine Anforderungsdefinition mit Feedbackzyklen in
 22 kurzen Abständen zu ermöglichen, sind Rahmenbedingungen erforderlich, die eine hohe Anzahl
 23 an Deployments in kurzen zeitlichen Abständen ermöglichen.

24 Die inkrementelle Entwicklung erfordert ein Umdenken in der Betrachtung von variablen und
 25 fixierten Bestandteilen von Software Entwicklungsprojekten. Eine derartige Betrachtung ist in
 26 agilen Softwareentwicklungsframeworks, wie beispielsweise Scrum, bereits integriert.



27
 28 *Abbildung 12: Agiles Projektdreieck (in Anlehnung an Highsmith (2009))*

1 Abbildung 12 zeigt die Veränderung der als fix definierten Betrachtungsobjekte in agilen
2 Projekten. Um die Anforderungsdefinition als wiederkehrenden Schritt innerhalb der
3 Wertschöpfungskette zu integrieren, ist es erforderlich, dass das Betrachtungsobjekt
4 Funktionalität, welches mithilfe von Anforderungen beschrieben wird, als nicht fixer Bestandteil
5 des Projekts angesehen wird. Die fixen Komponenten des agilen Projektdreiecks sind somit die
6 Durchlaufzeit und der Ressourceneinsatz. Der fixierte Zeit- und Ressourceneinsatz bei variablem
7 Umfang wird timeboxing genannt.

8 **5.3 Anwendungsfall Test und Deployment**

9 Durch die Verwendung einer dreistufigen Architektur bestehend aus Test-, Qualitäts- und
10 Produktivsystem sind mehrstufige Deployments und Tests auf allen drei Systemen erforderlich.
11 Die Tests sind aufgrund der Unterschiede in den Schnittstellen nicht ident und erfordern bei
12 manueller Durchführung Zeit. Der hohe Aufwand für Test und Deployment führt dazu, dass
13 versucht wird, die Anzahl an Deployments gering zu halten.

14 **5.3.1 Problembeschreibung**

15 Mit 88 von 462 Kalendertagen nimmt der Bereich Test und Inbetriebnahme nahezu 19% der
16 Gesamtdurchlaufzeit des beobachteten Arbeitsablaufs (Kapitel 2) ein. Die beiden Arbeitsschritte,
17 die die Durchlaufzeit am meisten beeinflusst haben (14 und 16 Tage), sind auf
18 Kommunikationsprobleme zurückzuführen. Einerseits handelte es sich um ein Missverständnis
19 zwischen den Entwicklern über die genaue Verteilung der Verantwortung und andererseits gab
20 es ein Verständigungsproblem betreffend implizite Anforderungen. Neben diesen beiden
21 Arbeitsschritten fällt auf, dass eine Änderung mit geringem Umfang in etwa zwei Kalenderwochen
22 in Anspruch nimmt, um in das Test- und das Qualitätssystem überführt und getestet zu werden.
23 Die Überführung der fertig getesteten Lösung benötigt im beobachteten Fall knapp drei Wochen,
24 wobei der Großteil davon in die Vorbereitung und das Durchlaufen des Change Management
25 Prozesses fließt.

26 **5.3.2 Lösungsansatz**

27 Das in Kapitel 5.2 erforderte Umdenken in Richtung eines variablen Funktionsumfangs ist fixer
28 Bestandteil des Lösungsansatzes, Kommunikationsproblemen entgegenzuwirken. Der
29 beschriebene Lösungsansatz im Anwendungsfall Anforderungsdefinition sieht eine frühe
30 Integration des Auftraggebers und häufige Abstimmungen über den Entwicklungsfortschritt vor.
31 Die Anwendung iterativer anstatt sequenzieller Vorgehensweisen verringert die negativen
32 Auswirkungen der beiden beobachteten Problemfelder. Durch die frühe Integration und häufige
33 Interaktion mit dem Auftraggeber, werden anfänglich nicht dokumentierte Anforderungen früher
34 im Entwicklungsprozess erkannt. Eine früh erkannte implizite Anforderung kann in weiterer Folge
35 analog zu jeder anderen Anforderung aufgenommen und abgearbeitet werden. Die Problematik
36 der nicht klar definierten Verantwortung erübrigt sich einerseits durch die Integration des DevOps

1 Prinzips der End-to-End Responsibility. Hierdurch trägt das gesamte Team die Verantwortung für
2 die Funktionalität der Lösung und somit das Vorhandensein aller nötigen Schnittstellen.
3 Andererseits ermöglichen häufige Deployments das frühe Erkennen von möglicherweise
4 fehlenden Schnittstellen. Durch das frühe Erkennen wird zusätzlicher Aufwand (durch mehrfache
5 Integration/Testungen/Deployments) vermieden.

6 Dieser Lösungsansatz und auch die Lösungsansätze der zuvor beschriebenen Anwendungsfälle
7 erfordern einen agilen Ansatz bei der Identifikation und der Umsetzung von Anforderungen. Agile
8 Vorgehensmodelle verfolgen das Prinzip von frühzeitiger und kontinuierlicher Auslieferung
9 wertvoller Software. Um eine Beschleunigung der Leistungserbringung zu erreichen, ist somit
10 eine Verringerung der Cycle Time erforderlich. Die Cycle Time beantwortet nach Mary und Tom
11 Poppendieck (2006) die Frage, wie lange es in einem Unternehmen dauert, eine Änderung in den
12 Produktivbetrieb zu überführen, wenn sich lediglich eine Zeile Programmcode ändert.

13 Mit der Cycle Time werden die Aufwände für Test, Deployment, Dokumentation und Freigabe der
14 Änderung zusammengefasst.

15 Nachfolgend wird auf die unterschiedlichen Aspekte der Cycle Time eingegangen. Im Anschluss
16 werden Lösungsansätze beschrieben, die auf die in Kapitel 2 beobachteten Szenarien aufbauen.

17 **5.3.2.1. Test und Deployment**

18 Eines der Prinzipien von DevOps sieht vor, so viel wie möglich zu automatisieren. Durch die
19 Bereitstellung einer Deployment Pipeline je Service, werden jegliche Änderungen am Service
20 durch einen standardisierten Weg in die Produktivumgebung überführt. Änderungen durchlaufen
21 unterschiedliche Stadien, in denen standardisierte Komponenten- und Integrationstests
22 durchgeführt werden. Jede Änderung am Service erweitert mit den eigenen Funktions- und
23 Komponententests das Testspektrum der Deployment Pipeline. Um manuelle
24 Integrationsaufwände zu verringern, wird nach dem DevOps Prinzip Sharing auf eine
25 gemeinsame Codebasis gesetzt, in die alle Entwickler des Projektteams ihre Änderungen
26 integrieren können.

27 Als Praktik kommt Continuous Integration zum Einsatz, welche eine schnelle Bereitstellung des
28 aktuellen Entwicklungsstands in unterschiedlichen Umgebungen ermöglicht. Um Continuous
29 Integration und eine Deployment Pipeline zum Einsatz bringen zu können, ist eine testgetriebene
30 Entwicklung erforderlich. Genau spezifizierte Anforderungen, wie beispielsweise Userstories im
31 Product Backlog in Scrum, enthalten grundsätzlich eine Definition of Done, die als Grundlage für
32 die Spezifikation von Testfällen der jeweiligen Anforderung dient.

33 *ITIL (2013, S. 132) fordert im Kontext von Deployments Folgendes: „Alle Changes und Releases*
34 *müssen vor dem Deployment unter realistischen Arbeitsbedingungen vollständig getestet*
35 *werden“.*

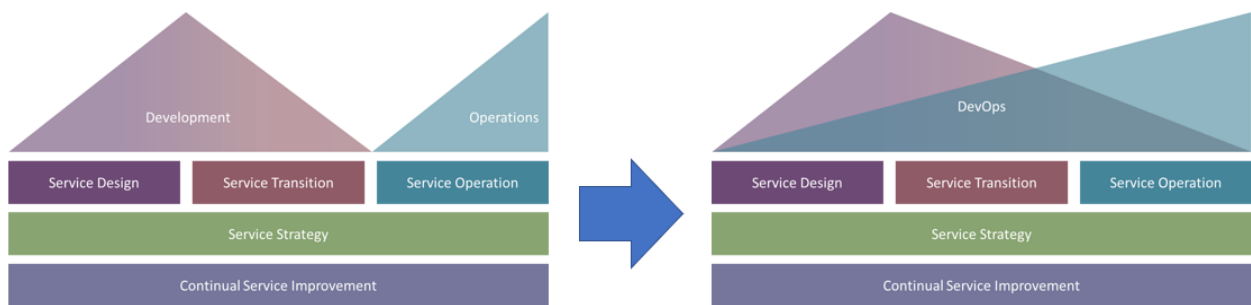
36 Durch die Verwendung der Deployment Pipeline und automatisierten Tests wird ein weiterer
37 Schritt zur Erfüllung der in ITIL geforderten vollständigen Testung vollzogen.

1 **5.3.2.2. Dokumentation**

2 Der Teil der Dokumentation gliedert sich in verschiedene Abschnitte. Nachfolgend werden die
 3 Bereiche Testdokumentation und Supportdokumentation betrachtet. Andere Arten der
 4 Dokumentation – Benutzerdokumentation und/oder Projektdokumentation und/oder
 5 Programmierdokumentation – werden in diesem Lösungsansatz nicht betrachtet.

6 **Testdokumentation:** Durch den Einsatz von Deployment Pipelines und das Vorhandensein einer
 7 Vielzahl an automatisierten Testfällen, sind auch die Schnittstellen der unterschiedlichen Services
 8 zu anderen Services klar. Um Aufwand auf Seiten der Service Owner zu sparen, werden die
 9 Informationen aus den automatisierten Testfällen dazu verwendet, den Service Ownern
 10 automatisiert Änderungsvorschläge an der Configuration Management Database vorzuschlagen.

11 **Supportdokumentation:** Der Ansatz von DevOps sieht vor, Mitarbeiter des IT Operations Teams
 12 in die Projektteams zu integrieren. Das führt dazu, dass die Erfahrungen der Teammitglieder aus
 13 IT Operations in die User-Experience einfließen. Es wird während der gesamten Wertschöpfung
 14 darauf geachtet, dass die Lösung auch den Anforderungen des IT Applikationsbetriebs entspricht.
 15 Das DevOps Prinzip der End-to-End Responsibility führt dazu, dass das Verständnis für die
 16 Lösung im gesamten Team und somit auch im Bereich der IT Operations vorhanden ist. Durch
 17 die Mitarbeit der Entwickler bei Lösungsbehebungen entsteht zudem wertvoller Wissenstransfer.



18
 19 *Abbildung 13: DevOps in ITIL*

20 Abbildung 13 illustriert das Auflösen der Grenze zwischen Entwicklung und Operations. Durch
 21 das Auflösen der Grenze zwischen Lösungsentwicklung und Lösungsbetrieb ist es nicht mehr
 22 notwendig, eine gesonderte Supportdokumentation zu erstellen. Ebenfalls wird hierdurch die
 23 Wahrscheinlichkeit vermindert, dass aus dem Applikationsbetrieb heraus, wie in Kapitel 2
 24 beobachtet, fehlerhafte Informationen an die Benutzer verteilt werden.

25 **5.3.2.3. Change Management Prozess**

26 ITIL sieht in der Beschreibung des Change Managements eine Unterteilung in drei Arten von
 27 Changes vor: der Normal-Change, der Standard-Change und der Emergency-Change.

28 Software Deployments sind in der Regel als Normal-Changes klassifiziert und durchlaufen daher
 29 den gesamten Change Management Prozess. Im beobachteten Fall, dokumentiert in Kapitel 2,
 30 sind über 80% der Durchlaufzeit für die Freigabe des Changes inklusive
 31 Entscheidungsvorbereitung erforderlich.

1 Der an die ITIL Best Practices angelehnte Change Management Prozess des beobachteten
2 Unternehmens (siehe Kapitel 2) sieht ebenfalls die Unterscheidung der Change Typen vor. Der
3 Change Management Prozess des Unternehmens kennt den pre-approved Change (entspricht
4 dem Standard Change), den Normal-Change und den Emergency-Change. Analog zu den
5 Empfehlungen durchlaufen der Normal-Change und der Emergency-Change den gesamten
6 Change Management Prozess. Der pre-approved Change hingegen verzichtet auf die
7 Risikoanalyse, die Planung und Dokumentation eines Roll-Back Szenarios, sowie die genaue
8 Dokumentation der auszuführenden Schritte. Der pre-approved Change erfordert, dass ein
9 Change ein geringes Risiko und eine standardisierte Methode zur Umsetzung hat.

10 Die geforderte standardisierte Methode ist durch den Einsatz der Deployment Pipeline gegeben,
11 die per Definition automatisierte Komponenten- und Integrationstests durchführt.

12 Die bereits beschriebenen Lösungsansätze erfordern eine hohe Anzahl an Deployments, welche
13 einen hohen Grad an Automatisierung erfordern. Der hohe Grad an Automatisierung entsteht
14 durch die Verwendung von automatisierten Komponenten- und Integrationstests und dem Einsatz
15 einer Deployment Pipeline. Das vom pre-approved Change geforderte geringe Risiko kann durch
16 eine weitere Erhöhung der Automatisierung erreicht werden. Durch das Automatisieren von User
17 Acceptance Tests und die Integration dieser Tests in die unterschiedlichen Stadien der
18 Deployment Pipeline, wird neben der Warranty des Service auch die Utility des Service getestet.
19 Die hohe Anzahl an erfolgreichen Deployments und die Vielzahl an automatisierten Tests
20 verringern das Risiko eines Deployments in die Produktivumgebung.

21 Durch diese Ansätze können Deployments in die Produktivumgebung mittels „Standard-Change“
22 bzw. „pre-approved Change“ anstatt mit „Normal-Change“ durchgeführt werden.

23 **5.3.3 Erfordernisse und Voraussetzungen**

24 Die Umsetzung dieses Lösungsansatzes erfordert die Umsetzung der DevOps Prinzipien „cross-
25 functional Autonomous Teams“ und „End-to-End Responsibility“, wodurch das Entwicklungsteam
26 um Personen aus dem Operations Team erweitert wird. Mit der tiefen Einbindung des Kunden
27 wird das agile Prinzip zur engen Integration des Fachbereichs erfüllt. Um einen derart hohen Grad
28 an Automatisierung zu erreichen, ist es erforderlich, dass für jede Userstory eines Service (nicht
29 nur der Änderung) ein Testfall existiert. Nur dadurch kann bei der Integration einer Änderung
30 bereits im Testsystem geprüft werden, ob diese Änderung negative Auswirkungen auf die
31 Stabilität des Service hat. Eine gut gepflegte Configuration Management Database ist
32 erforderlich, damit automatisierte Tests zu richtigen Ergebnissen führen. Sofern die Schnittstellen
33 und Verbindungen zwischen den Services dokumentiert sind, kann deren Funktion verifiziert
34 werden. Damit durch automatisierte Tests und automatisierte Deployments in die
35 Qualitätsumgebung verwertbare und belastbare Testergebnisse entstehen, ist es erforderlich,
36 dass eine hohe Übereinstimmung zwischen Qualitäts- und Produktivsystem existiert.

1 5.4 Anpassungen des Arbeitsablaufs

2 Der beobachtete Arbeitsablauf aus Kapitel 2 enthält eine Vielzahl an Verzögerungen. Die
3 nachfolgenden Punkte integrieren Ansätze aus DevOps, Agile und Lean, um die Durchlaufzeit
4 des beobachteten Szenarios unter Beibehaltung der Systemstabilität zu verringern. Als Methode
5 wird das Gedankenexperiment gewählt.

6 Ein Gedankenexperiment ist nach der Definition eines lexikalischen Standardwerks (Brockhaus)
7 „[...] ein lediglich gedanklich vollzogenes Experiment, das real entweder so nicht durchführbar ist
8 beziehungsweise mangels technischer Möglichkeiten oder wegen ethischer, pragmatischer oder
9 prinzipieller Gründe nicht durchgeführt werden kann.“ Sehr ähnlich wird es auch von der
10 Enzyklopädie Philosophie und Wissenschaftstheorie (1980, S. 712) definiert: „[...]Experiment,
11 dessen Ablauf methodisch entwickelt ist, das jedoch aus technisch-kontingenten (Kosten des
12 technischen Aufwands etc.) oder prinzipiellen (Dimension der gedachten Versuchsinstrumente,
13 moralische Grenzen beim Umgang mit Versuchspersonen etc.) Gründen nicht ausgeführt wird.“
14 Nach der Definition von Kühne (2005, S. 9) ist mit einem Gedankenexperiment ein
15 Erkenntnisgewinn ohne empirische Arbeit gemeint. Er beschreibt Gedankenexperimente als „[...]
16 eine Wissenschaftsmethode gemeint, die man in Gedanken ausführt. [...] Es bereichert unser
17 Wissen und läßt uns also etwas entdecken, was wir vorher noch nicht wußten.“

18 Nachfolgend beschriebenes Gedankenexperiment betrachtet die Arbeitsschritte nach der
19 finanziellen Freigabe der Projektumsetzung. Jene Abläufe, die nicht in den vorangegangenen
20 Anwendungsfällen behandelt wurden, werden nach Möglichkeit gleich gehalten, um die
21 Vergleichbarkeit zu erleichtern. Der Ablauf geht davon aus, dass sowohl für den Service
22 *SharePoint Intranet Services* als auch für den Service *Salesforce CRM Deployment Pipelines*
23 existieren. Die Deployment Pipelines prüfen mit automatisierten Tests die Funktion des jeweiligen
24 Service samt aller bisherigen Erweiterungen/Anpassungen und stellen einen geordneten Weg in
25 das Test-, Qualitäts- und Produktivsystem bereit. Des Weiteren beruht das Gedankenexperiment
26 auf der Annahme, dass die gleiche Menge an Personentagen pro Woche aufgewendet wird und
27 beliebig auf die Personen verteilt werden kann. Die nachfolgende Tabelle zeigt den adaptierten
28 Arbeitsablauf. Zusätzlich zu den Spalten ID und Inhalt besitzt die Tabelle die Spalte „Ref. ID“.
29 Diese Spalte beinhaltet eine Referenz ID, welche auf den inhaltlich ähnlichen Schritt des
30 beobachteten Arbeitsablaufs aus Kapitel 2 verweist. Die Spalte ID beginnt bei 101, um auf den
31 ersten Blick eine eindeutige Unterscheidung zwischen Beobachtung und Gedankenexperiment
32 zu ermöglichen.

ID	Ref. ID	Inhalt	Dauer
101	14	Nach der finanziellen Freigabe startet der Projektleiter den Projektstartprozess gemäß interner Projektabwicklungsrichtlinie. Diese sieht im Startprozess eine detailliertere Planung des Projekts vor. Neben Zielen, Aufgaben und Kosten werden auch Termine und personelle Ressourcen in einem höheren Detailgrad geplant. Der Projektleiter überträgt Punkte aus dem erstellten Projektantrag und dem Business Case in seine Planungsdokumentation. Der Projektleiter entschließt sich	7

ID	Ref. ID	Inhalt	Dauer
		<p>für einen hybriden Ansatz innerhalb des Projektmanagements. Projektplanung und Reporting folgen den Vorgaben der Projektmanagement Richtlinie, die Umsetzung setzt auf den Scrum Prozess. Das vom Projektleiter nominierte Projektteam beinhaltet unter anderem eine Person mit Expertise im IT Applikationsbetrieb und das Scrum-Team. Er selbst übernimmt die Rolle des Product Owners und nominiert neben einem Scrum Master ein fünfköpfiges Entwicklungsteam (kurz: Team). Alle notwendigen Kompetenzen und das notwendige Know-How für die Umsetzung der Anforderungen sind im Team vorhanden, wodurch keine Projektmitarbeiter, die Projektkernteammitgliedern zuarbeiten, erforderlich sind.</p> <p>Der Projektleiter vereinbart einen Termin, um Arbeitspakete zu identifizieren und diese in den Projektstrukturplan zu übertragen.</p>	
102	15	<p>Der Projektleiter (PL), in seiner Rolle als Product Owner (PO), nutzt den Termin, um beim Team ein Verständnis für die Arbeit des Kunden und dessen Problemfall zu schaffen. Er stellt die zu schaffende Lösung in den Mittelpunkt und präsentiert den Beitrag dieser Lösung zum Unternehmenserfolg.</p> <p>Der Projektleiter / Product Owner (PO/PL) kommuniziert die Anforderungen des Kunden und versucht gemeinsam mit dem Scrum Master und dem Team eine geeignete Sprint Dauer festzulegen.</p> <p>Das Team einigt sich auf eine Sprint Dauer von 3 Wochen.</p> <p>Mit der Sicherheit, dass belastbare Schätzungen auf Userstory Ebene im späteren Verlauf des Projekts noch erstellt werden können, grenzt das Team den Aufwand je User Story auf Basis von Erfahrungswerten ein.</p>	7
103	16	<p>Der Projektleiter bespricht den Ressourcenbedarf mit den Vorgesetzten der Teammitglieder und vereinbart eine gleichbleibende, fixe wöchentliche Zuordnung zu diesem Projekt.</p> <p>Auf Basis der zugesicherten wöchentlichen Ressourcen schätzt das Team die Dauer der Umsetzung auf 8-9 Sprints.</p> <p>Der PO/PL überträgt 9 Sprints in die Projektplanung und hinterlegt nach jedem Sprintende einen Meilenstein. Es resultiert ein Umsetzungszeitraum von 24 bis 27 Wochen.</p>	7
104	17	<p>Der PO/PL befüllt den Product Backlog mit vagen User Stories und spezifiziert einige Userstories gemeinsam mit dem Auftraggeber genauer. Zu diesem Zeitpunkt werden jene Userstories spezifiziert, um den Anforderungen an das minimum viable product gerecht zu werden.</p>	28

ID	Ref. ID	Inhalt	Dauer
		Die Spezifikationen dieser Anforderungen enthalten bereits Akzeptanzkriterien, Testfälle und eine Definition of Done (DoD).	
105	18	Auf Basis der bislang vorliegenden Informationen werden Softwarearchitektur- und Infrastrukturkonzepte erstellt. Diese Konzepte enthalten auch die informationssicherheitsbezogene Relevanz in den Punkten Vertraulichkeit, Integrität und Verfügbarkeit und erfüllen die Vorgaben des Projektabwicklungsprozesses.	7
106	19	Aufgrund der ISO 27001 Relevanz der geplanten Umsetzung ist die Abteilung „global IT Security“ zu involvieren. Der Projektleiter präsentiert das Softwarearchitektur- und das Infrastrukturkonzept der unternehmensinternen Abteilung für IT Sicherheit, die aufgrund des klaren Umsetzungskonzepts, welches ebenfalls IT sicherheitsrelevante Punkte betrachtet, grünes Licht für die weitere Umsetzung gibt.	7
107	20,21	Der Projektleiter integriert die Konzepte in die Projektplanungsdokumentation und informiert das Team, dass die Umsetzung gestartet werden kann. Die Vorbereitung der Lösungsentwicklung ist dadurch abgeschlossen. Das Team trifft sich, um die Elemente im Product Backlog zu diskutieren und das Sprint Planning Meeting vorzubereiten. Der PO/PL erarbeitet vor dem Start des ersten Sprints die Spezifikation der höchstpriorisierten Einträge im Product Backlog. Die Einträge beinhalten Benutzerakzeptanzkriterien, Testfälle, eine Definition of Done (DoD) und eine initiale Aufwandsschätzung des Teams.	7
108		Der erste Sprint startet mit dem Sprint Planning Meeting. Das Team wählt die identifizierten Arbeitspakete für das minimum viable product und weitere hochpriorisierte Userstories aus dem Product Backlog aus und präzisiert die Aufwandsschätzung. Nachdem sich das Team über den zu erwartenden Aufwand der Arbeitspakete und Userstories einig ist, übernimmt es diese in den Sprint Backlog. Das Sprint Planning Meeting ist abgeschlossen.	1
109		Direkt im Anschluss an das Sprint Planning Meeting beginnt das Team, die Features und User-Stories in geeignete Tasks aufzuteilen. Die Teammitglieder starten mit der Entwicklung von Unit-Tests für ihre Tasks und beginnen im Anschluss daran, ihre Tasks zu entwickeln. Des Weiteren werden Integrations- und Akzeptanztests entwickelt und in die Deployment Pipeline der beiden Services hinzugefügt. Bei jedem Check-In eines Entwicklers laufen automatisiert Unit Tests. Nach erfolgreichem Durchlaufen der Unit Tests werden am Testsystem Integrations- und	19

ID	Ref. ID	Inhalt	Dauer
		<p>Akzeptanztests automatisiert durchgeführt. Mehrfach während des Sprints werden auf Knopfdruck automatisierte Deployments in das Qualitätssystem gestartet, welche das Durchlaufen von weiteren Integrations- und Akzeptanztests initiieren.</p> <p>Parallel zu den Arbeiten des Teams erweitert der PO/PL in Zusammenarbeit mit der Vertretung des Auftraggebers die detailliertere Spezifikation der wichtigsten verbleibenden Userstories des Product Backlogs.</p> <p>Jeweils zu Beginn des Tages trifft sich das Team, um einander von den Ergebnissen des vergangenen Tages und dem Plan für den aktuellen Tag zu berichten (Daily Stand-Up). Der PO/PL nimmt schweigend teil und wird über den Fortschritt im Sprint informiert. Der Scrum Master nimmt moderierend teil, sorgt für die Einhaltung der Zeitspanne und notiert bei Bedarf Probleme, die das Team bei der Arbeit behindern.</p>	
110		<p>Am letzten Werktag des Sprints treffen sich das Team, der PO/PL, der Scrum Master, Vertreter des Auftraggebers und weitere Interessensvertreter.</p> <p>Der PO/PL stellt vor, welche Funktionalitäten im aktuellen Sprint umgesetzt wurden und welche noch ausständig sind. Das Team demonstriert die fertiggestellten Funktionalitäten (das Produktinkrement). Feedback vom PO/PL und vom Auftraggeber wird entgegengenommen und entsprechende Einträge im Product Backlog werden hinzugefügt und/oder adaptiert.</p> <p>Ohne Beisein von Vertretern des Auftraggebers und anderen Interessensvertretern reflektiert das Team, der PO/PL und der Scrum Master den Ablauf des vergangenen Sprints (Sprint Retrospektive). Gemeinsam werden Maßnahmen abgeleitet, um den Ablauf der folgenden Sprints zu verbessern.</p>	1
111		<p>Der zweite Sprint startet erneut mit dem Sprint Planning Meeting. Das Team wählt detailliert spezifizierte Stories mit hoher Priorität für den Sprint aus und startet im Anschluss mit der Entwicklung der dazugehörigen Testfälle. Analog zu den Abläufen im ersten Sprint werden die Entwicklungen nach jedem Check-In automatisiert geprüft.</p> <p>Aufgrund einer fehlenden Schnittstelle schlägt ein Integrationstest im Qualitätssystem fehl. Das Team löst das Problem umgehend und setzt die Entwicklungen fort. PO/PL und Auftraggeber spezifizieren parallel zu den Entwicklungen weitere User Stories genauer. Da der Vertreter des Auftraggebers bereits inoffizielle Informationen zu einer potentiellen</p>	21

ID	Ref. ID	Inhalt	Dauer
		<p>Änderung in den Anforderungen gehört hat, bittet er den PO/PL die Priorität der eventuell betroffenen Userstory zu verringern und in einem späteren Sprint durchzuführen. Im Laufe des Sprints finden daily Stand-Up Meetings statt, die für den Informationsfluss im Team sorgen.</p> <p>Beim Sprint Review am Ende des Sprints zeigen sich der PO/PL, der Vertreter des Auftraggebers und die Interessensvertreter zufrieden. Der Service Owner des Service Salesforce CRM nimmt beim Sprint Review teil und nutzt die Möglichkeit sich ein Bild über die geplanten Änderungen sowie deren Auswirkungen auf die Konsumenten seines Service zu verschaffen.</p> <p>In der Sprint Retrospektive wird die Problematik der fehlenden Schnittstelle offen diskutiert und das Team vereinbart Abläufe, die ein erneutes Auftreten verhindern.</p>	
112	24,25	<p>Der dritte Sprint startet erneut mit einem Sprint Planning Meeting und ist von daily Stand-Up Meetings geprägt. Vor dem Start der Implementierung werden abermals die dazugehörigen Testfälle erstellt und in die Deployment Pipeline integriert. Während des Sprints kommuniziert der Auftraggeber eine weitere wichtige Anforderung. Der PO/PL prüft die Anforderung auf Auswirkungen auf die IT-Sicherheit und/oder die strategische Ausrichtung der Lösung und fügt sie als weiteren Eintrag im Product Backlog hinzu. Bei der gemeinsamen Spezifikation mit dem Vertreter des Auftraggebers fällt auf, dass der Eintrag im Product Backlog Anforderungen einer zurückgestellten bestehende Userstory substituiert. Die betroffene Userstory wird entsprechend inhaltlich angepasst. Weitere Userstories werden in Abhängigkeit zur Priorität des Auftraggebers spezifiziert. Die Einträge im Sprint Backlog werden nicht verändert.</p> <p>Im Sprint Review wird der Fortschritt der Entwicklung bekanntgegeben und die fertiggestellten Funktionalitäten werden demonstriert. Auf Basis des Feedbacks werden die Einträge im Product Backlog adaptiert.</p> <p>Zeitgleich wird die Aufwandsschätzung der geänderten User Story adaptiert. Der Vertreter des Auftraggebers sieht den entstehenden Mehraufwand. Er wünscht sich, dass diese Änderung anstatt einer Nice-to-Have Funktionalität mit gleichem Aufwand umgesetzt wird, wodurch keine Änderung des Gesamtaufwands, der Durchlaufzeit oder des Budgets entsteht. Da hierdurch ein aufwändiger Project Change Request verhindert wird, stimmt der PO/PL dem zu.</p> <p>Die Sprint Retrospektive fällt kurz aus.</p>	21

ID	Ref. ID	Inhalt	Dauer
113		Sprint vier arbeitet nach denselben Prinzipien und findet ohne nennenswerte Zwischenfälle statt. Während des Sprints demonstriert der PO/PL dem Projektkernteammitglied aus dem Umfeld des IT-Applikationsbetriebs (Operations Engineer) den aktuellen Stand der Entwicklungen. Der PO/PL lädt ihn ein, beim nächsten Sprint Review dabei zu sein und sein Feedback zu geben. Das Feedback umfasst Verbesserungsvorschläge hinsichtlich der Benutzerführung und Tipps zur Verbesserung der intuitiven Bedienung. Dieses Feedback wird in den Product Backlog als eigener Punkt aufgenommen.	21
114		Sprint fünf, sechs und sieben arbeiten nach denselben Prinzipien und finden ohne nennenswerte Zwischenfälle statt.	63
115	26	Während des achten typischen Sprintablaufs werden erstmals systemübergreifende Funktionalitäten (SharePoint und Salesforce) implementiert. Die Informationen aus den entwickelten Testfällen werden dazu verwendet, um Änderungsvorschläge an der Configuration Management Database automatisiert zu erzeugen. Der Kunde testet im Verlauf des Sprints den aktuellen Stand am Qualitätssystem und bemerkt, dass eine implizite Anforderung nicht umgesetzt ist. Parallel dazu führt das Projektkernteammitglied aus dem Umfeld des IT-Applikationsbetriebs (Operations Engineer) manuelle Tests durch und entdeckt einige kleinere Fehler, die nicht durch die automatisierten Tests entdeckt wurden. Sowohl die fehlende Anforderung als auch die entdeckten Fehler werden im Product Backlog hinterlegt. Wie für neue Elemente im Product Backlog üblich, werden die Elemente genauer spezifiziert, priorisiert, mit Abnahmekriterien versehen, Testszenarien beschrieben und vom Team aufwandstechnisch bewertet. Das Team beschließt aufgrund des bisherigen guten Projektverlaufs, die zusätzliche Anforderung ohne Verlängerung der Gesamtlaufzeit umsetzen zu können.	21
116	27, 29	Zu Beginn des neunten Sprints sind alle im Product Backlog befindlichen Einträge genau spezifiziert und werden vollständig in den Sprint Backlog übernommen. Aufgrund der noch verfügbaren Kapazitäten wird zusätzlich das substituierte Nice-To-Have Feature in den Sprint Backlog übernommen. Nach 15 Tagen ist das Team mit der Umsetzung der Einträge im Sprint Backlog fertig und nutzt die verbleibende Zeit, um die Funktionalitäten manuell zu testen und daraus resultierende kleinere Fehler zu beheben. Der PO/PL nimmt im Sprint Review die letzten User Stories ab.	21

ID	Ref. ID	Inhalt	Dauer
117	31-41	Durch das mehrmalige Deployment in das Qualitätssystem während der Sprints, ist ein gesondertes Deployment nicht erforderlich. Die zu verwendenden Schnittstellen sind bereits im Vorfeld je Zielsystem definiert, wodurch kein manuelles Ersetzen von Stub-Objekten notwendig ist. Ein Teil der manuellen Testaktivitäten des Auftraggebers wurde parallel zur Fertigstellung der letzten Userstories durchgeführt. Nach abschließenden Tests des Auftraggebers wird die inhaltliche Freigabe zur Überführung in das Produktivsystem erteilt.	2
118	42	Der Projektleiter informiert das Team über die positive Nachricht, initiiert einen Change Management Prozess, der für jede Änderung an der IT Infrastruktur erforderlich ist und startet mit der Hauptaktivität „register and plan change“ des Change Management Prozesses. Die Vorbereitung des Deployments ist dadurch abgeschlossen. Das Risiko wird aufgrund der Vielzahl an automatisierten Tests als gering eingestuft. Als standardisierte Methode zur Überführung in die Produktivumgebung wird die Nutzung der Deployment Pipeline angegeben und als Change Typ wird „pre-approved Change“ ausgewählt. Unter Zuhilfenahme der automatisiert aufbereiteten Vorschläge zur Anpassung der jeweiligen CMDBs beschreibt der PO/PL die Auswirkungen des geplanten Changes.	1
119	43, 50, 51	Das Projektteam wartet aufgrund des Umfangs der Änderung mit dem Start des Deployments auf das Wochenende und startet den Automatisierten Prozess zur Überführung in die Produktivumgebung. Nach dem Deployment laufen automatisierte Funktionstests ab, die den Erfolg des Deployments bestätigen. Der Prozessschritt „implement change“ und „close change“ werden durch die Automatisierung in einem Arbeitsschritt erledigt.	3
120	52	Die Service Owner bestätigen den erfolgreichen Ablauf des Changes. Die automatisiert aufbereiteten Vorschläge zur Anpassung der CMDB werden dem Service Owner des jeweiligen Service zur Verfügung gestellt. Die Service Owner aktualisieren ihre CMDB anhand dieser Informationen. Der Projektleiter wird über den positiven Abschluss des Prozesses informiert, wodurch der Change Management Prozess abgeschlossen ist. Der Projektleiter bittet den Ansprechpartner des Kunden, das Deployment zu überprüfen und das Projekt abzunehmen.	2
121	53	Durch die Integration des Operations Engineers in das Projekt Team ist der Wissenstransfer über den Aufbau und die Funktionsweise der Lösung bereits während des Projektverlaufs erfolgt. Die weitere Verteilung des Wissens innerhalb des Teams des IT-	0

ID	Ref. ID	Inhalt	Dauer
		Applikationsbetriebs erfolgt informell und ohne gesondertes Handover Meeting.	
122	54	Der Ansprechpartner des Kunden bestätigt das positive Deployment. Die durch die Auftraggeberseite geschulten Mitarbeiter verwenden den angepassten Angebotsfreigabeprozess produktiv.	1
123	57	Der Ansprechpartner des Kunden führt gemeinsam mit dem Projektleiter die formelle Abnahme des Projekts durch.	1

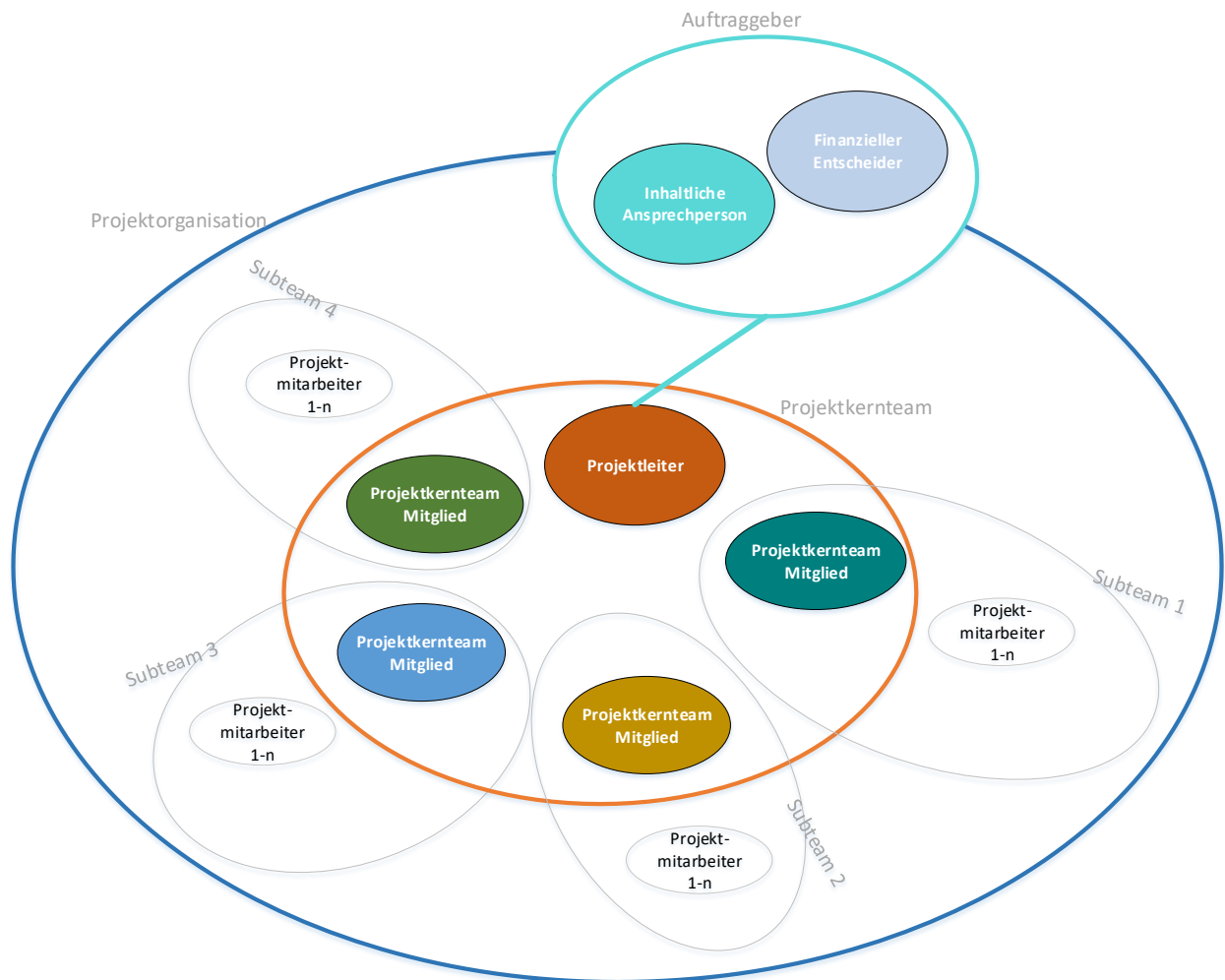
1 *Tabelle 6: angepasster Ablauf der Leistungserbringung*

2 Der in Tabelle 5 beschriebene angepasste Arbeitsablauf erfordert trotz des agilen Ansatzes eine
3 erste Phase des Requirements Engineerings. Da ausschließlich das minimum viable product
4 spezifiziert wird, wird die Zeitspanne für des Erheben der Anforderungen wesentlich verkürzt.

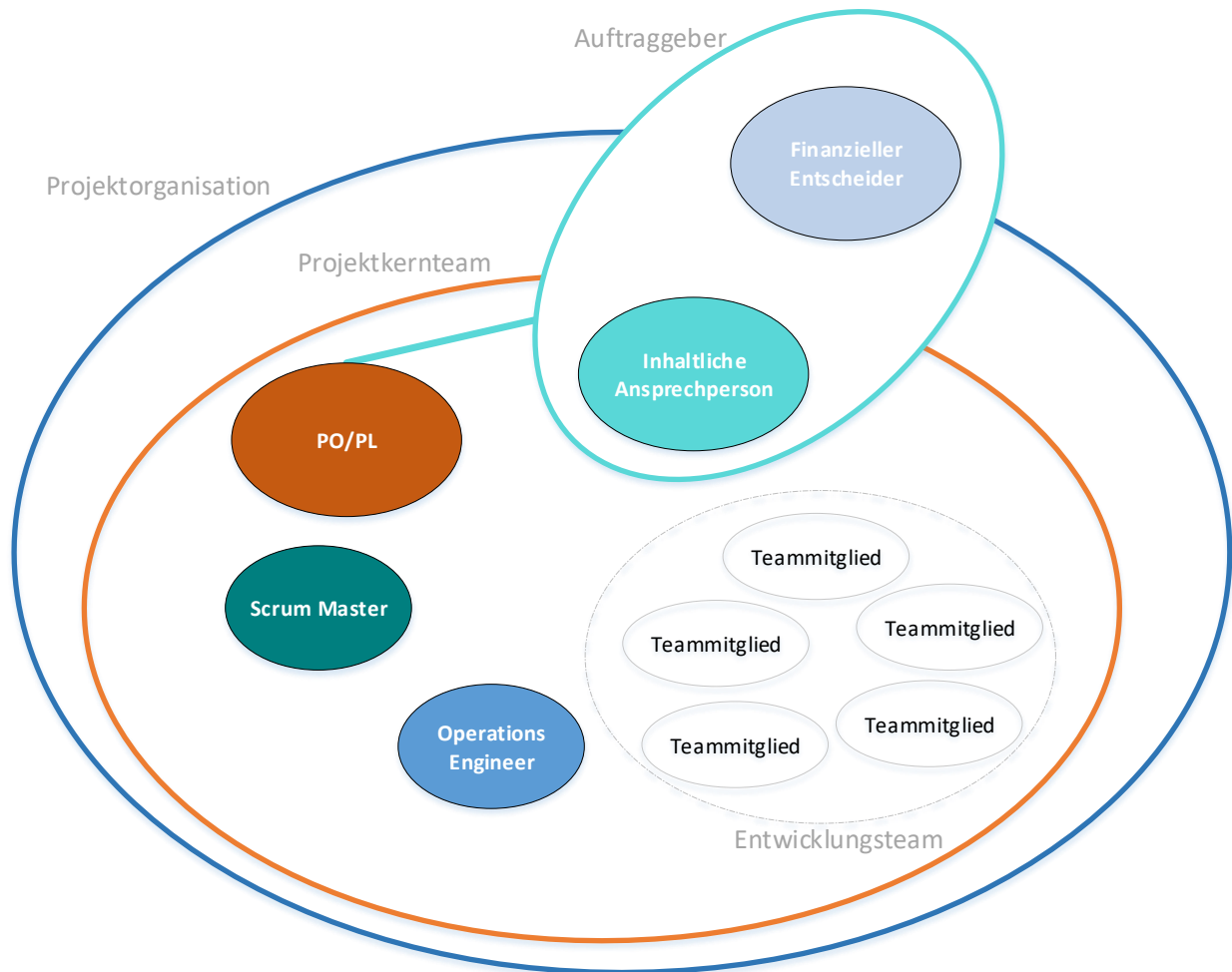
5 Die Parallelisierung von Funktionsentwicklung und Detailspezifikation der weiteren
6 Anforderungen ermöglicht einen früheren Start der Softwareentwicklung und somit einen früheren
7 Fertigstellungszeitpunkt.

8 Roland Gareis (2006) ist einer der prägenden Personen für den im Unternehmen führenden
9 Projektmanagement Ansatz IPMA. Seiner Beschreibung nach beinhaltet ein Projektorganigramm
10 alle Personen, die im Projekt aktiv tätig sind. Zusätzlich unterscheidet er zwischen
11 Projektkernteammitgliedern und meist in Subteams organisierten Projektmitarbeitern. Die
12 Kernteammmitglieder übernehmen Arbeitspaketverantwortlichkeit und steuern Projektmitarbeiter
13 aus, Projektmitarbeiter unterstützen bei der Abarbeitung der Arbeitspakete.
14 Projektkernteammmitglieder sind zudem bei den regelmäßigen Projektabstimmungen vertreten.

15 Abbildung 14 zeigt die Projektorganisation, wie sie anhand der Beobachtungen in Kapitel 2
16 aufgebaut war.



- 1
- 2 *Abbildung 14: Projektorganisation klassisch (in Anlehnung an Gareis (2006))*
- 3 In Abbildung 14 ist ersichtlich, dass die Projektorganisation aus einer Vielzahl an Personen
- 4 besteht, die einen kleinen Beitrag zum Projekterfolg beitragen. Die hohe Anzahl an Personen hat
- 5 einen hohen Abstimmungsaufwand zur Folge.
- 6 Die im Gedankenexperiment etablierte Projektorganisation sieht eine Integration des Operations
- 7 Engineers vor und integriert den inhaltlichen Verantwortlichen des Auftraggebers in das
- 8 Projektkernteam. Abbildung 15 stellt diesen Aufbau grafisch dar.



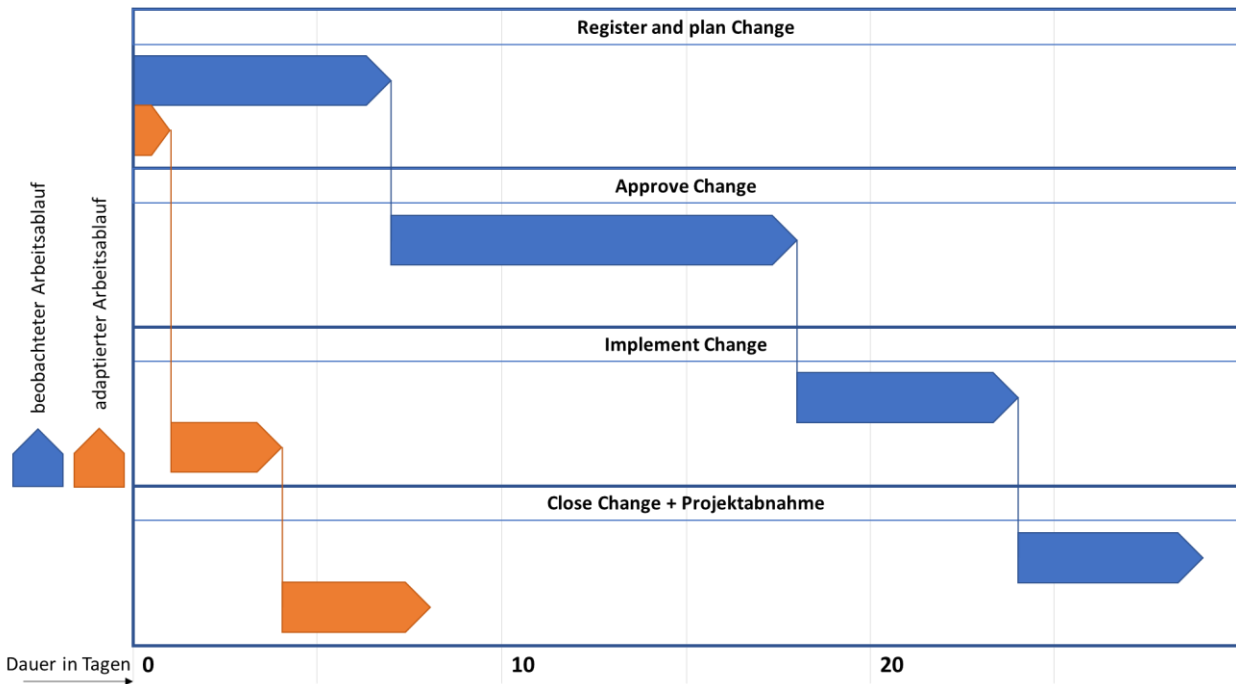
1

2 *Abbildung 15: adaptierte Projektorganisation*

3 Die adaptierte Projektorganisation, visualisiert in Abbildung 15, sieht die Aufnahme einer
 4 größeren Anzahl an Personen in das Projektkernteam vor. Durch die Integration des inhaltlichen
 5 Ansprechpartners, des Scrum Masters und des Operations Engineers, wird das
 6 Entwicklungsteam bestmöglich bei der Leistungserbringung unterstützt. Durch die Vergrößerung
 7 des Kernteams wird im beschriebenen Fall auf Subteams verzichtet. Die Integration des
 8 Operations Engineer unterstützt durch das Einbringen seiner Erfahrungen in die Konzepte des
 9 Entwicklungsteams die Beibehaltung der Gesamtsystemstabilität. Zusätzlich wird bereits in einem
 10 frühen Stadium ein Wissenstransfer in den IT-Applikationsbetrieb sichergestellt. Die Integration
 11 des Auftraggebers erhöht die Qualität der Lösung. Änderungen und Fehler können früher im
 12 Entwicklungsprozess behandelt werden, wodurch die Aufwände für die Fehlerbehebungen
 13 verringert werden. Merx (1999) beschreibt dazu, dass je später ein Fehler entdeckt wird, desto
 14 länger ist in der Regel die Dauer der Fehlerbehebung.

15 Die Verwendung der Deployment Pipeline ermöglicht eine weitgehende Automatisierung des
 16 Deployments in verschiedene Zielsysteme. Dies verringert einerseits den Zeitaufwand, der je
 17 Deployment betrieben werden muss und stellt andererseits die Beibehaltung Systemstabilität
 18 durch eine Vielzahl automatisierter Tests sicher. Aus dem verringerten Zeitaufwand je
 19 Deployment ergibt sich die Option auf kurze Deploymentzyklen, die die Basis für die Umsetzung
 20 agiler Softwareentwicklungsprozesse darstellen.

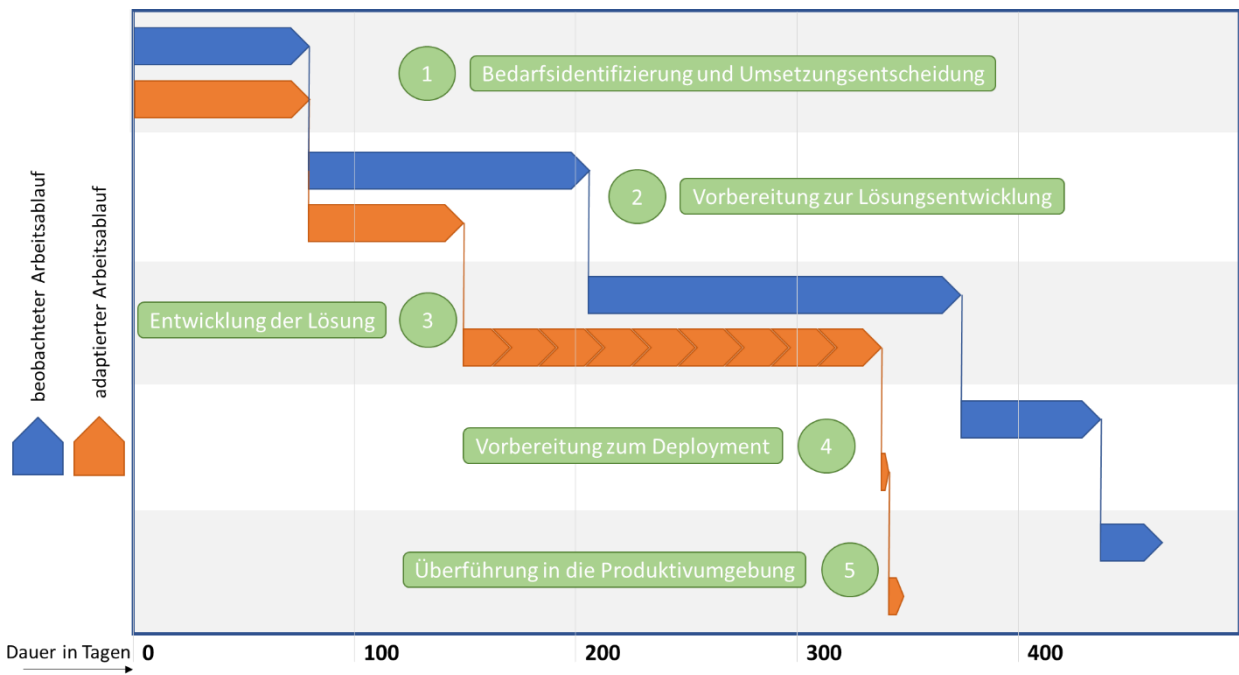
- 1 Neben der Verkürzung der Deploymentzyklen bringt die Deployment Pipeline einen weiteren
- 2 Vorteil mit sich. Im Kontext von ITIL ermöglicht der Einsatz einer Deployment Pipeline die
- 3 Kategorisierung von Produktivsystem-Deployments als Standard-Change mit geringem Risiko.
- 4 Hierdurch wird eine Verkürzung der Dauer zur Überführung in die Produktivumgebung erreicht.
- 5 Auf Basis der Beobachtungen in Kapitel zwei und des Gedankenexperiments kann die Dauer des
- 6 Change Management Prozesses wie folgt dargestellt werden.



7
8 *Abbildung 16: Gegenüberstellung Dauer des Change Management Prozesses*

9 In Abbildung 16 ist die sequenzielle Abfolge der unterschiedlichen Phasen des Change
 10 Management Prozesses ersichtlich. Die Abbildung unterscheidet zwischen dem beobachteten
 11 Arbeitsablauf und dem adaptierten Arbeitsablauf. Aufgrund des Durchlaufens des Change
 12 Management Prozesses des Typs „Standard Change“ ist die Phase „Approve Change“ nicht
 13 erforderlich, wodurch keine Zeitspanne im adaptierten Arbeitsablauf vorgesehen ist. In der
 14 Abbildung wird bewusst auf die Darstellung von Gesamtwerten verzichtet. Es soll verdeutlicht
 15 werden, inwiefern die Gesamtdauer des Change Management Prozesses durch die Adaptierung
 16 verändert werden kann.

17 In der Gesamtheit betrachtet, kann der zeitliche Vergleich zwischen beobachteten und
 18 adaptierten Arbeitsablauf wie folgt visualisiert werden.



1
2 *Abbildung 17: Gegenüberstellung des zeitlichen Verlaufs*

3 Der zeitliche Verlauf, der in Abbildung 17 dargestellt ist, zeigt die 5 zeitintensiven Abschnitte, die
 4 zwischen der Kommunikation des Bedarfs und der produktiven Nutzung der Lösung liegen. Im
 5 Gedankenexperiment wird davon ausgegangen, dass Abschnitt 1 unangetastet bleibt, wodurch
 6 auch der zeitliche Aufwand ident bleibt. Klar ersichtlich ist, dass die Dauer von Abschnitt 2,
 7 Vorbereitung zur Lösungsentwicklung, verkürzt wird. Das liegt darin begründet, dass nicht die
 8 gesamte Lösung detailliert spezifiziert wird, bevor mit der Lösungsentwicklung begonnen wird.
 9 Hierdurch wird die Detailspezifikation der Anforderungen teilweise parallel zur Entwicklung der
 10 Lösung durchgeführt. Im Abschnitt 3, Entwicklung der Lösung, ist in der Abbildung ersichtlich,
 11 dass im angepassten Arbeitsablauf eine Unterteilung in 9 Unterbereiche erfolgt. Diese
 12 Unterteilung bildet die Sprints ab, in denen die Lösung entwickelt wird. Der zeitliche Aufwand, der
 13 für die Entwicklung der Lösung im adaptierten Arbeitsablauf erforderlich ist, ist größer als im
 14 beobachteten Arbeitsablauf. Einer der Gründe für die Verlängerung ist der erhöhte
 15 Entwicklungsaufwand, der durch die Entwicklung von automatisierten Testfällen entsteht. Einen
 16 weiteren Grund bilden die kontinuierlichen Anpassungen, die das Team aufgrund des schnellen
 17 Feedbacks, welches sie bei jedem Commit und jedem Deployment auf Basis der Testfälle
 18 erhalten, durchführen. Aus der Abbildung geht klar hervor, dass die zeitlichen Aufwände für die
 19 Vorbereitung des Deployments (Abschnitt 4) und die Überführung in die Produktivumgebung
 20 (Abschnitt 5) sehr stark zurückgegangen sind. Die Ursache für den Rückgang ist der hohe Grad
 21 an Automatisierung und die starke Verringerung der bereits beschriebenen Dauer des Change
 22 Management Prozesses. Zusammenfassend kann aus der Abbildung entnommen werden, dass
 23 aufgrund der Integration von Ansätzen aus DevOps, Agile und Lean IT unter Beibehaltung der
 24 Systemstabilität eine Beschleunigung der Leistungserbringung erreicht werden kann.

25 Auf Basis des Gedankenexperiments und der beschriebenen Erkenntnisse resultiert folgende
 26 Aussage: „Durch die Verwendung einer Deployment-Pipeline pro ITIL-Service und der
 27 Automatisierung von Test- und Deploymenttätigkeiten kann in Verbindung mit agilen

- 1 Softwareentwicklungsmethoden und der Integration von Personen aus dem Applikationsbetrieb
- 2 in das Projektteam die Geschwindigkeit der Leistungserbringung ohne negativer
- 3 Beeinträchtigung der Systemstabilität erreicht werden."

1 **6 CONCLUSIO**

2 Die sich stets verändernden Rahmenbedingungen und Leistungsanforderungen im
3 geschäftlichen Umfeld zwingen IT Organisationen dazu, ihre Flexibilität und Geschwindigkeit der
4 Leistungserbringungen anzupassen. Unternehmen mit etablierten Prozessen, die an das
5 Rahmenwerk ITIL angelehnt sind, müssen Wege finden, diesen Anforderungen gerecht zu
6 werden, ohne die durch ITIL gewonnene Stabilität zu verlieren. In dieser Arbeit wurde diese
7 Problemstellung anhand eines beobachteten Szenarios näher betrachtet und potentielle
8 Lösungsansätze entwickelt. Die Lösungsansätze und Anpassungen sehen vor, dass durch die
9 Automatisierungen im Bereich von Test- und Deploymenttätigkeiten, der Deploymentprozess
10 zwischen Entwicklungs-, Test-, Qualitäts- und Produktivumgebung automatisiert werden kann.
11 Der automatisierte Deploymentprozess wird in einer Deployment Pipeline pro Service abgebildet
12 und dient für weitere Anpassungen am Service als Grundlage. Dies ermöglicht eine
13 Transformation von Bereitstellung von Software in Hauptversionen (Major Releases) zur
14 kontinuierlichen Bereitstellung neuer Software. Im Bereich dieser Arbeit endet die
15 Automatisierung der Bereitstellung im Qualitätssystem, die Überführung in das Produktivsystem
16 wird manuell initiiert. Die Arbeit sieht als Basis einen beobachteten Arbeitsablauf, der als Projekt
17 umgesetzt wurde, vor. Aufgrund der Einmaligkeit der Bedingungen, die ein Projekt
18 charakterisieren, kann kein Experiment durchgeführt werden, welches auf Basis des
19 beobachteten Arbeitsablaufs die Wirkung der Lösungsansätze beweist. Um die Lösungsansätze
20 weiter zu untermauern, wird anstelle des Experiments ein Gedankenexperiment gewählt.

21 Das Gedankenexperiment nennt als Voraussetzung, dass je Service bereits eine Deployment
22 Pipeline vorhanden ist und baut die Schrittabfolge sowie die Bewertung der Dauer der Schritte
23 auf dieser Voraussetzung auf. Diese Voraussetzung zu erfüllen ist für Unternehmen, die
24 hauptsächlich manuelle Testungen durchführen, mit einem hohen initialen Aufwand verbunden.
25 Eine weitere zugrunde liegende Voraussetzung ist die zeitgleiche Verfügbarkeit der notwendigen
26 personellen Ressourcen, die das Know-How in den relevanten Technologien und Plattformen
27 besitzen. Die Herausforderung besteht darin, tiefgehendes Know-How in zwei großen
28 Plattformen, die in unterschiedlichen Programmiersprachen (Java, JavaScript, C#) entwickelt
29 werden, in einem Team von fünf Personen zu vereinen.

30 Aufgrund des fehlenden empirischen Aspekts des Gedankenexperiments kann die
31 zugrundeliegende Hypothese „Die gezielte Integration von Ansätzen aus DevOps, Lean und Agile
32 in Arbeitsabläufe, die an ITIL angelehnt sind, kann die Geschwindigkeit der Leistungserbringung
33 erhöhen, ohne die Stabilität des Systems negativ zu beeinflussen.“ weder bestätigt, noch
34 falsifiziert werden. Daraus ergibt sich, dass die Forschungsfrage „Wie kann die Geschwindigkeit
35 der Leistungserbringung in an ITIL angelehnten Prozessen erhöht werden, ohne die durch ITIL
36 gewonnene Stabilität zu verringern?“ beantwortet, aber dessen Richtigkeit nicht wissenschaftlich
37 bewiesen werden kann. Als Antwort auf die Forschungsfrage resultiert aus dem
38 Gedankenexperiment das folgende Ergebnis: „Durch die Verwendung einer Deployment-Pipeline
39 pro ITIL-Service und der Automatisierung von Test- und Deploymenttätigkeiten kann in
40 Verbindung mit agilen Softwareentwicklungsmethoden und der Integration von Personen aus

1 dem Applikationsbetrieb in das Projektteam, die Geschwindigkeit der Leistungserbringung ohne
2 negativer Beeinträchtigung der Systemstabilität erreicht werden." Diese Antwort bietet die Basis
3 für weitere Forschungsarbeiten, die einerseits mit Hilfe von qualitativen Interviews von Experten
4 die empirische Komponente einbringen, oder andererseits durch die Umsetzung in Unternehmen
5 wichtige weitere Erkenntnisse ableiten können. Bei einer Pilotierung im Unternehmen kann neben
6 dem Einfluss auf Systemstabilität und Umsetzungsgeschwindigkeit auch das Augenmerk auf die
7 einhergehenden Auswirkungen auf die Mitarbeiter- und/oder Kundenzufriedenheit gelegt werden.
8 Die Pilotierung in einem Unternehmen, welches auf klassische, sequenzielle
9 Softwareentwicklung setzt, wird neben den technischen Belangen auch den Kulturwandel als
10 herausfordernde Komponente beinhalten.

11 Im Februar 2019 wurde ITIL v4 Foundation veröffentlicht, welches einen Überblick über die
12 aktualisierte ITIL Version gibt. Die neuen Auflagen der Kernpublikationen, die nähere Details
13 beinhalten, werden für das zweite Halbjahr 2019 erwartet. Dies kann zum Anlass genommen
14 werden, die Ergebnisse dieser Arbeit den sich gegebenenfalls geänderten ITIL Best Practices
15 gegenüberzustellen.

1 **ANHANG A – UMFRAGE ZUR INTEGRATION VON DEVOPS IN**
2 **ITIL**

3 Sehr geehrte Damen und Herren,

4 diese Umfrage wird von Stefan Weigl im Rahmen seiner Masterarbeit an der FH Campus 02 –
5 Fachhochschule der Wirtschaft, Studienrichtung IT & Wirtschaftsinformatik durchgeführt.

6 Ziel dieser Umfrage ist es die Relevanz der Integration von Ansätzen aus DevOps in an ITIL
7 angelehnten Prozessen der Phase Service Transition zu überprüfen.

8 Mit Ihrer Teilnahme tragen Sie dazu bei, den Grundstein für weitere Untersuchungen zu legen,
9 um in weiterer Folge die folgende Forschungsfrage zu beantworten: „Wie kann die
10 Geschwindigkeit der Leistungserbringung in an ITIL angelehnten Prozessen erhöht werden,
11 ohne die durch ITIL gewonnene Stabilität zu verringern?“.

12 Die Umfrage besteht aus 10 Fragen, wird anonym durchgeführt und wird weniger als fünf
13 Minuten Ihrer Zeit in Anspruch nehmen. Als Teilnehmer der Umfrage haben Sie die Möglichkeit
14 eine Zusammenfassung der Ergebnisse und daraus gewonnenen Erkenntnisse zu erhalten.

15 Bitte hinterlassen Sie mir hierzu eine E-Mail mit Ihren Kontaktdaten auf
16 stefan.weigl@edu.campus02.at.

17 Vielen Dank für Ihre Teilnahme!

18 Mit freundlichen Grüßen

19 Stefan Weigl

20 Student an der FH Campus 02 – Fachhochschule der Wirtschaft

21 Studienrichtung IT & Wirtschaftsinformatik

22 stefan.weigl@edu.campus02.at

1 ANHANG A1 – ERGEBNISSE DER UMFRAGE

2 Der Fragebogen ist wie folgt aufgebaut:

3 • Welche der folgenden Rollen beschreibt Ihre Tätigkeit am besten?

4 ○ IT Management

5 ○ IT Governance / Steuerung

6 ○ IT Architekt

7 ○ Anwendungsentwicklung

8 ○ IT und Anwendungssupport

9 ○ ich bin nicht innerhalb der IT Abteilung tätig

10 • Wie viele Mitarbeiter arbeiten in Ihrem Unternehmen?

11 ○ 1-9

12 ○ 10 - 49

13 ○ 50 - 249

14 ○ Über 250

15 • Gibt es in Ihrem Unternehmen einen einheitlichen Prozess, um Änderungen an einer oder
16 mehreren IT-Komponenten durchzuführen?

17 ○ Ja

18 ○ Nein

19 ○ Ich weiß nicht

20 • Gibt es einen einheitlichen Prozess, der das Management von Änderungen an IT
21 Infrastruktur oder Software regelt und über die Durchführung der Änderung entscheidet?

22 ○ Ja

23 ○ Nein

24 ○ Ich weiß nicht

25 • Gibt es einen Prozess zur Autorisierung und Inbetriebnahme neuer Softwareprodukte?

26 ○ Ja

27 ○ Nein

28 ○ Ich weiß nicht

29 • Manuelle Anwendungstests im Entwicklungs-, Qualitäts- und Produktivsystem erzeugen
30 einen hohen personellen Aufwand, wodurch tägliche Softwarereleases verhindert werden.

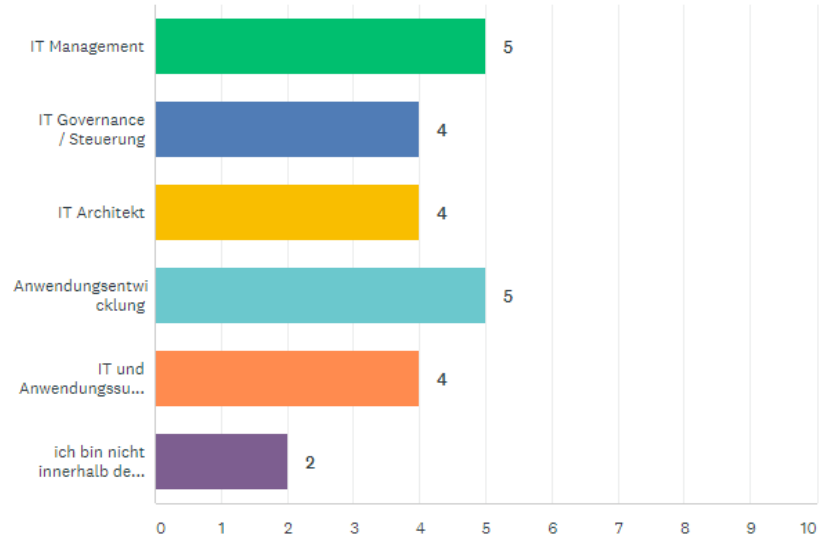
31 ○ Fünfstufige Likert-Skala beginnend mit „trifft gar nicht zu“ bis „trifft vollkommen zu“

- 1 • Aus der strikten Trennung zwischen Entwicklungsabteilung und dem IT-Betrieb entstehen
2 Kommunikationsprobleme, die langfristig erhöhte Kosten im Betrieb der Anwendung nach
3 sich ziehen.
- 4 ○ Fünfstufige Likert-Skala beginnend mit „trifft gar nicht zu“ bis „trifft vollkommen zu“
- 5 • Mehrstufige Deployment-Freigabeprozesse verhindern die schnelle Auslieferung von
6 neuen Funktionen.
- 7 ○ Fünfstufige Likert-Skala beginnend mit „trifft gar nicht zu“ bis „trifft vollkommen zu“
- 8 • Fehlendes Verständnis für den Gesamtnutzen der Anwendung im Kerngeschäft des
9 Unternehmens führt zu Fehlentscheidungen in der Konzeption, Entwicklung und im
10 Betrieb der Anwendung.
- 11 ○ Fünfstufige Likert-Skala beginnend mit „trifft gar nicht zu“ bis „trifft vollkommen zu“
- 12 • Sich schnell ändernde Marktanforderungen erfordern das Verständnis über die zu
13 unterstützenden Kerntätigkeiten des Kunden, um schnelle Änderungen an Lösungen
14 bereitstellen zu können.
- 15 ○ Fünfstufige Likert-Skala beginnend mit „trifft gar nicht zu“ bis „trifft vollkommen zu“

F1

Welche der folgenden Rollen beschreibt Ihre Tätigkeit am besten?

Answered: 24 Skipped: 0



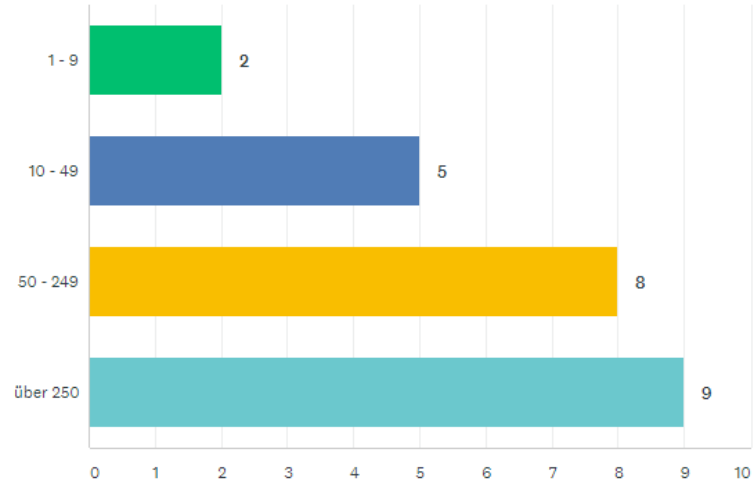
ANTWORTOPTIONEN	BEANTWORTUNGEN
IT Management	20,83% 5
IT Governance / Steuerung	16,67% 4
IT Architekt	16,67% 4
Anwendungsentwicklung	20,83% 5
IT und Anwendungssupport	16,67% 4
ich bin nicht innerhalb der IT Abteilung tätig	8,33% 2
GESAMT	24

1

F2

Wie viele Mitarbeiter arbeiten in Ihrem Unternehmen?

Answered: 24 Skipped: 0



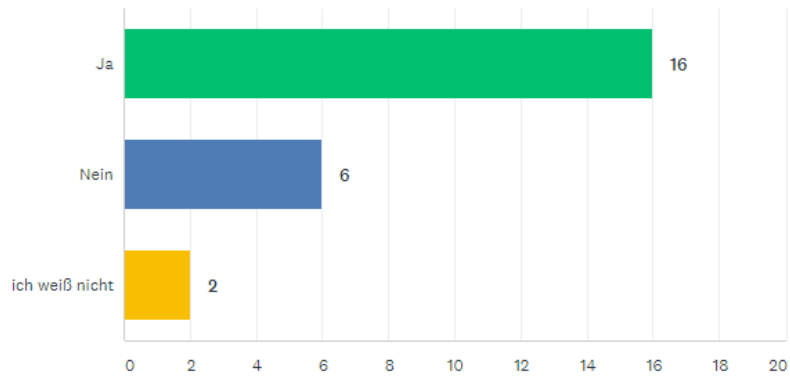
ANTWORTOPTIONEN	BEANTWORTUNGEN
▼ 1 - 9	8,33% 2
▼ 10 - 49	20,83% 5
▼ 50 - 249	33,33% 8
▼ über 250	37,50% 9
GESAMT	24

1

F3

Gibt es in Ihrem Unternehmen einen einheitlichen Prozess, um Änderungen an einer oder mehreren IT-Komponenten durchzuführen?

Answered: 24 Skipped: 0



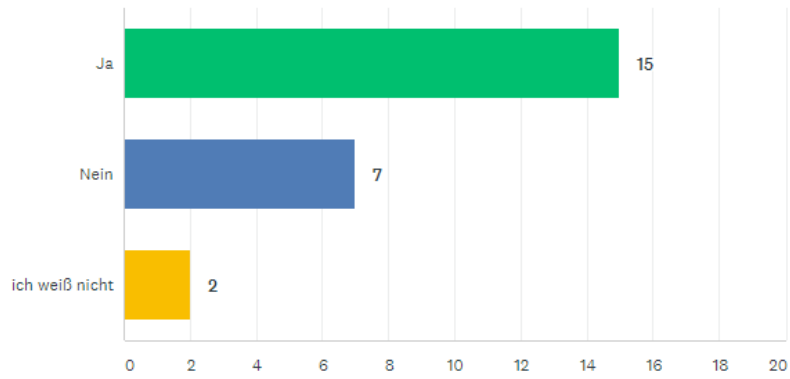
ANTWORTOPTIONEN	BEANTWORTUNGEN
▼ Ja	66,67% 16
▼ Nein	25,00% 6
▼ ich weiß nicht	8,33% 2
GESAMT	24

1

F4

Gibt es einen einheitlichen Prozess, der das Management von Änderungen an IT Infrastruktur oder Software regelt und über die Durchführung der Änderung entscheidet?

Answered: 24 Skipped: 0



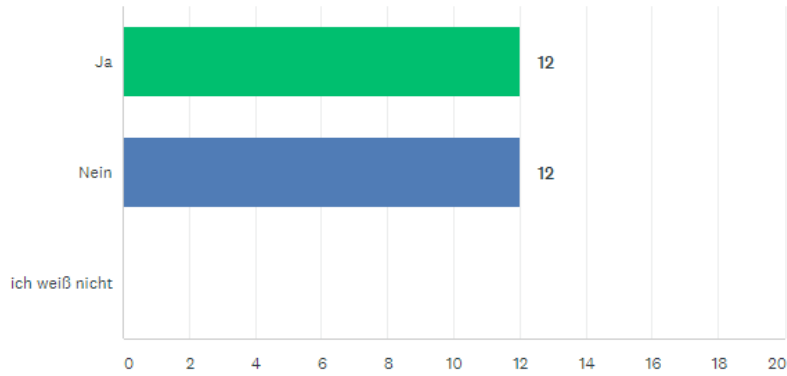
ANTWORTOPTIONEN	BEANTWORTUNGEN
▼ Ja	62,50% 15
▼ Nein	29,17% 7
▼ ich weiß nicht	8,33% 2
GESAMT	24

1

F5

Gibt es einen Prozess zur Autorisierung und Inbetriebnahme neuer Softwareprodukte?

Answered: 24 Skipped: 0



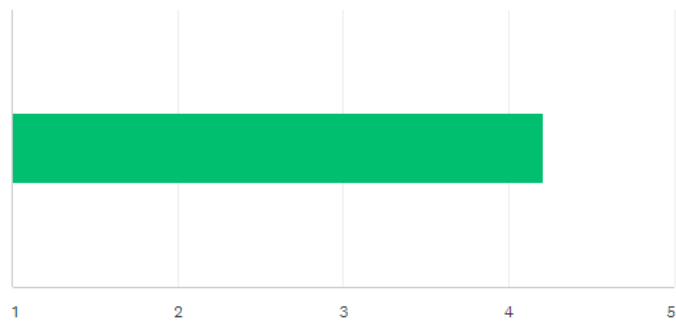
ANTWORTOPTIONEN	BEANTWORTUNGEN
Ja	50,00% 12
Nein	50,00% 12
ich weiß nicht	0,00% 0
GESAMT	24

1

F6

Manuelle Anwendungstests im Entwicklungs- Qualitäts- und Produktivsystem erzeugen einen hohen personellen Aufwand wodurch tägliche Software releases verhindert werden.

Answered: 23 Skipped: 1



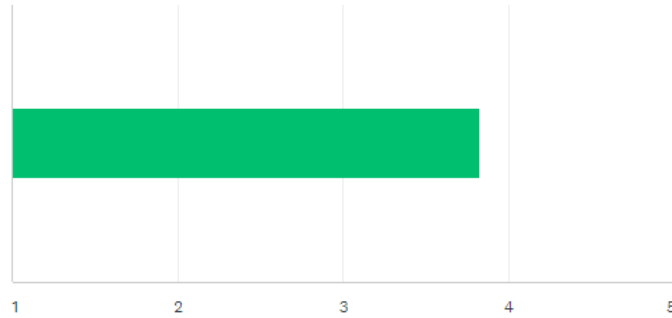
ANTWORTOPTIONEN	DURCHSCHNITTLICHE ANZAHL	GESAMTANZAHL	BEANTWORTUNGEN
Beantwortungen	4	97	23
Befragte gesamt: 23			

2

F7

Aus der strikten Trennung zwischen Entwicklungsabteilung und dem IT-Betrieb entstehen Kommunikationsprobleme, die langfristig erhöhte Kosten im Betrieb der Anwendung nach sich ziehen.

Answered: 23 Skipped: 1



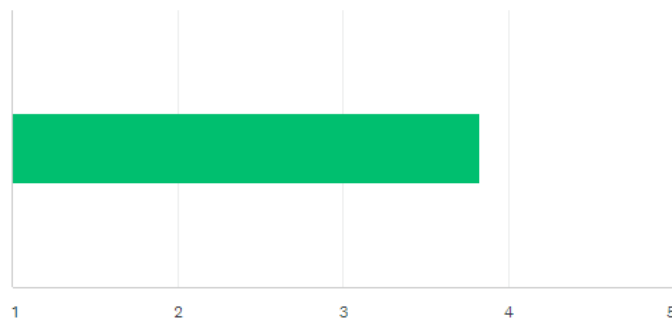
ANTWORTOPTIONEN	DURCHSCHNITTLICHE ANZAHL	GESAMTANZAHL	BEANTWORTUNGEN
Beantwortungen	4	88	23
Befragte gesamt: 23			

1

F8

Mehrstufige Deployment-Freigabeprozesse verhindern die schnelle Auslieferung von neuen Funktionen.

Answered: 23 Skipped: 1



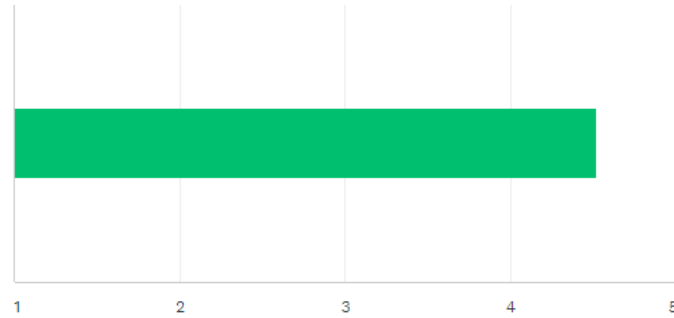
ANTWORTOPTIONEN	DURCHSCHNITTLICHE ANZAHL	GESAMTANZAHL	BEANTWORTUNGEN
Beantwortungen	4	88	23
Befragte gesamt: 23			

2

F9

Fehlendes Verständnis für den Gesamtnutzen der Anwendung im Kerngeschäft des Unternehmens führt zu Fehlentscheidungen in der Konzeption, Entwicklung und im Betrieb der Anwendung.

Answered: 23 Skipped: 1



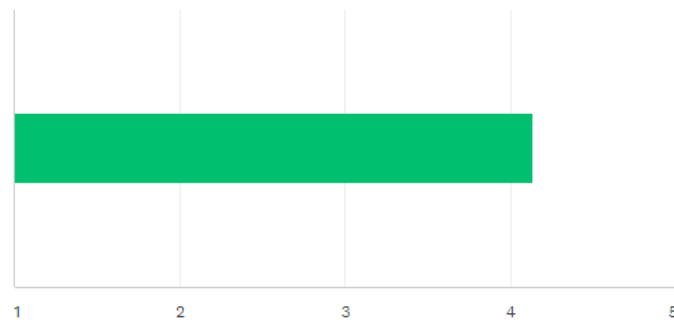
ANTWORTOPTIONEN	DURCHSCHNITTLICHE ANZAHL	GESAMTANZAHL	BEANTWORTUNGEN
Beantwortungen	5	104	23
Befragte gesamt: 23			

1

F10

Sich schnell ändernde Marktanforderungen erfordern das Verständnis über die zu unterstützenden Kerntätigkeiten des Kunden, um schnelle Änderungen an Lösungen bereitstellen zu können.

Answered: 22 Skipped: 2



ANTWORTOPTIONEN	DURCHSCHNITTLICHE ANZAHL	GESAMTANZAHL	BEANTWORTUNGEN
Beantwortungen	4	91	22
Befragte gesamt: 22			

2

1 ABKÜRZUNGSVERZEICHNIS

Abkürzung	Bedeutung
CD	Continuous Delivery
CDe	Continuous Deployment
CI	Configuration Items
CI	Continuous Integration
CMDB	Configuration Management Database
CMS	Configuration Management System
CSI	Continual Service Improvement
DIKW	Data – Information – Knowledge – Wisdom
DoD	Definition of Done
IaC	Infrastructure as a Code
IPMA	International Project Management Association
ISM	Information Security Management
ISTQB	International Software Testing Qualifications Board
ITIL	IT Infrastructure Library
ITSCM	IT Service Continuity Managements
LeSS	Large Scale Scrum
MTTR	Mean Time to Recover
MVP	Minimum Viable Product
OLA	Operational Level Agreements
PL	Projektleiter
PO	Product Owner
RfC	Request for Change
SACM	Service Asset and Configuration Management
SDP	Service-Design-Packages
SKMS	Service Knowledge Management System
SLA	Service Level Agreement
SLM	Service Level Management
SLR	Service Level Requirements
SPOC	Single Point of Contact
TDD	Test-Driven-Development
UC	Underpinning Contracts

1 **ABBILDUNGSVERZEICHNIS**

2	Abbildung 1: Aufbau der Arbeit	11
3	Abbildung 2: Überblick Change Management Prozess.....	24
4	Abbildung 3: zeitlicher Verlauf der beobachteten Leistungserbringung	29
5	Abbildung 4: ITIL Service Lifecycle (Great Britain: Cabinet Office, 2013)	31
6	Abbildung 5: Zusammenhang zwischen den Prozessen in Service Strategy (Great Britain: Cabinet Office,	
7	2013).....	32
8	Abbildung 6: Zusammenhang zwischen den Prozessen in Service Design (Great Britain: Cabinet Office,	
9	2013).....	36
10	Abbildung 7: Zusammenhang zwischen den Prozessen in Service Transition (Great Britain: Cabinet Office,	
11	2013).....	41
12	Abbildung 8: Zusammenhang zwischen den Prozessen in Service Operations (Great Britain: Cabinet	
13	Office, 2013)	47
14	Abbildung 9: Der Scrum Zyklus (DasScrumTeam AG, 2018)	59
15	Abbildung 10: Muda, Muri & Mura (Hanschke, 2015)	61
16	Abbildung 11: 8 typische Reaktionen in den 3 Phasen der Veränderung (in Anlehnung an Becker & Labucay	
17	(2012))	67
18	Abbildung 12: Agiles Projektdreieck (in Anlehnung an Highsmith (2009)).....	76
19	Abbildung 13: DevOps in ITIL	79
20	Abbildung 14: Projektorganisation klassisch (in Anlehnung an Gareis (2006))	89
21	Abbildung 15: adaptierte Projektorganisation.....	90
22	Abbildung 16: Gegenüberstellung Dauer des Change Management Prozesses.....	91
23	Abbildung 17: Gegenüberstellung des zeitlichen Verlaufs.....	92
24		

1 **TABELLENVERZEICHNIS**

2	Tabelle 1: zeitlicher Ablauf der Bedarfsidentifizierung und Umsetzungsentscheidung	15
3	Tabelle 2: zeitlicher Ablauf zur Vorbereitung der Lösungsentwicklung.....	18
4	Tabelle 3: zeitlicher Ablauf der Entwicklung der Lösung.....	19
5	Tabelle 4: zeitlicher Ablauf der Vorbereitung des Deployments	22
6	Tabelle 5: zeitlicher Ablauf der Überführung in die Produktivumgebung	28
7	Tabelle 6: angepasster Ablauf der Leistungserbringung.....	88
8		

1 LITERATURVERZEICHNIS

- 2 Ackoff, R. L. (1989). *From Data to Wisdom*. Journal of Applied Systems Analysis.
- 3 Atteslander, P. (2010). *Methoden der empirischen Sozialforschung* (13, neu bearbeitete und erweiterte
4 Auflage Ausg.). Berlin: Erich Schmidt Verlag GmbH & Co.
- 5 Beck, K. (2000). *Extreme Programming Explained*. Boston: Addison-Wesley.
- 6 Beck, K. (2003). *Test Driven Development: By Example*. Boston: Addison Wesley.
- 7 Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D.
8 (2001). *Manifesto for Agile Software Development*. Abgerufen am 16. 04 2018 von Manifesto for
9 Agile Software Development: <http://agilemanifesto.org/principles.html>
- 10 Becker, M., & Labucay, I. (2012). *Organisationsentwicklung*. Stuttgart: Schäffer-Poeschel Verlag.
- 11 Brockhaus. (2019). *Gedankenexperiment*. Abgerufen am 17. 03 2019 von Brockhaus - NE GmbH:
12 <https://brockhaus.de/ecs/enzy/article/gedankenexperiment>
- 13 Brugger, R. (2009). *Der IT Business Case: Kosten erfassen und analysieren - Nutzen erkennen und
14 quantifizieren - Wirtschaftlichkeit nachweisen und realisieren* (Zweite korrigierte und erweiterte
15 Auflage Ausg.). Heidelberg: Springer-Verlag.
- 16 DasScrumTeam AG. (24. 11 2018). *Scrum | Das Scrum Team*. Von Training und Coaching, um agil
17 erfolgreich zu sein | Das Scrum Team: <https://www.dasscrumteam.com/de/scrum> abgerufen
- 18 DevOps Agile Skills Association. (2017). *6 Principles of DevOps – DevOps Agile Skills Association (DASA)*.
19 Abgerufen am 16. 02 2018 von DevOps Agile Skills Association (DASA):
20 <https://www.devopsagileskills.org/dasa-devops-principles/>
- 21 Essmat, A. (2016). *IT Projects and IT Operations go Hand in Hand*. The EAITSM Inc.
- 22 Fitz, T. (08. 02 2009). *Continuous Deployment*. Abgerufen am 19. 03 2018 von Continuous Deployment:
23 <http://timothyfitz.com/2009/02/08/continuous-deployment/>
- 24 Fowler, M. (01. 05 2006). *Continuous Integration*. Abgerufen am 18. 03 2018 von Continuous Integration:
25 <https://www.martinfowler.com/articles/continuousIntegration.html>
- 26 Gareis, R. (2006). *Happy Projects*. Wien: MANZ'sche Verlags- und Universitätsbuchhandlung GmbH.
- 27 Gloger, B. (2016). *Scrum: Produkte zuverlässig und schnell entwickeln* (5., überarbeitete Auflage Ausg.).
28 München.

- 1 Gräf, L. (1999). Optimierung von WWW-Umfragen: Das Online Pretest-Studio. In B. Batinic (Hrsg.), *Online*
2 *Research: Methoden, Anwendungen und Ergebnisse* (S. 159 -178). Göttingen [u. a]: Hogrefe.
- 3 Great Britain: Cabinet Office. (2013). *ITIL Continual Service Improvement*. Norwich: The Stationery Office
4 Ltd.
- 5 Great Britain: Cabinet Office. (2013). *ITIL Service Design*. Norwich: The Stationery Office Ltd.
- 6 Great Britain: Cabinet Office. (2013). *ITIL Service Operations*. Norwich: The Stationery Office Ltd.
- 7 Great Britain: Cabinet Office. (2013). *ITIL Service Strategy*. Norwich: The Stationery Office Ltd.
- 8 Great Britain: Cabinet Office. (2013). *ITIL Service Transition*. Norwich: The Stationery Office Ltd.
- 9 Greenleaf, R. K., & Spears, L. C. (1998). *The Power of Servant-Leadership: Essays*. San Francisco: Berret-
10 Koehler Publishers, Inc.
- 11 Guckenheimer, S. (22. 03 2018). *Was ist Infrastruktur als Code? | DevOps*. Von Was ist Infrastruktur als
12 Code? | DevOps: <https://www.visualstudio.com/de/learn/what-is-infrastructure-as-code/> abgerufen
- 13 Hanschke, I. (17. 11 2015). Abgerufen am 18. 12 2018 von Lean IT-Management | Informatik Aktuell:
14 [https://www.informatik-aktuell.de/management-und-recht/projektmanagement/lean-it-](https://www.informatik-aktuell.de/management-und-recht/projektmanagement/lean-it-management-voraussetzung-fuer-business-agilitaet.html)
15 [management-voraussetzung-fuer-business-agilitaet.html](https://www.informatik-aktuell.de/management-und-recht/projektmanagement/lean-it-management-voraussetzung-fuer-business-agilitaet.html)
- 16 Hernstein Institut für Management und Leadership. (2017). *Fehlerkultur und Selbstorganisation in*
17 *Unternehmen*. Wien.
- 18 Hopp, W., & Spearman, M. (2011). *Factory Physics* (Third Edition Ausg.). Long Grove: Waveland Press.
- 19 Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases Through Build, Test, and*
20 *Deployment Automation*. Boston: Addison-Wesley.
- 21 Humble, J., & Molesky, J. (08 2011). Why Enterprises Must Adopt Devops to Enable. *Cutter IT Journal Vol.*
22 *24, No. 8*. Von <https://www.cutter.com/sites/default/files/itjournal/fulltext/2011/08/itj1108.pdf>
23 abgerufen
- 24 ISTQB Incorporated sciety. (11. 02 2018). *ISTQB Glossary*. Von ISTQB Glossary: <http://glossary.istqb.org/>
25 abgerufen
- 26 Kammel, A., & Groth, U. (1994). *Lean management: Konzept - Kritische Analyse - Praktische*
27 *Lösungsansätze*. Wiesbaden: Gabler.
- 28 Kleinaltenkamp, M., Plinke, W., & Geiger, I. (2013). *Auftrags- und Projektmanagement* (2.,vollständig
29 überarbeitete Auflage Ausg.). Wiesbaden: Springer Fachmedien.
- 30 Kühne, U. (2005). *Die Methode des Gedankenexperiments*. Frankfurt am Main: Suhrkamp Verlag.

- 1 Larman, C., & Vodde, B. (2010). *Practices for Scaling Lean and Agile Development: Large, Multisite, and*
2 *Offshore Product Development with Large-Scale Scrum (Agile Software Development Series).*
3 Boston: Addison-Wesley.
- 4 Mayer, T.-L., Wald, A., Gleich, R., & Wagner, R. (2008). *Advanced Project Management.* Berlin: Lit Verlag
5 Dr. W.Hopf.
- 6 Merx, O. (1999). *Qualitätssicherung bei Multimedia-Projekten.* Springer-Verlag: Berlin#.
- 7 Mittelstraß, J. (1980). *Enzyklopädie Philosophie und Wissenschaftstheorie.* Berlin: Bibliographisches
8 Institut.
- 9 Ohno, T. (1988). *Toyota Production System: Beyond Large-scale Production.* Tokyo: Diamond.
- 10 Pohl, K., & Rupp, C. (2015). *Basiswissen Requirements Engineering (4., überarbeitete Auflage Ausg.).*
11 Heidelberg: dpunkt.verlag GmbH.
- 12 Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit: An Agile*
13 *Toolkit.* Boston: Addison Wesley.
- 14 Poppendieck, M., & Poppendieck, T. (2006). *Implementing Lean Software Development: From Concept to*
15 *Cash.* Boston: Addison-Wesley.
- 16 Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create*
17 *Radically Successful Businesses.* New York: Crown Publishing Group.
- 18 Riley, C. (26. 01 2015). *Metrics of DevOps - devops.com.* Von Metrics of DevOps - devops.com:
19 <https://devops.com/metrics-devops/> abgerufen
- 20 Rother, M. (2013). *Die Kata des Weltmarktführers (2. erweiterte Auflage Ausg.).* Frankfurt/Main: Campus
21 Verlag GmbH.
- 22 Rubin, K. S. (2014). *Essential Scrum: Umfassendes Scrum-Wissen aus der Praxis.* Heidelberg: mitp.
- 23 Schwaber, K. (1997). SCRUM Development Process. In J. Sutherland, D. Patel, C. Casanave, J. Miller, &
24 G. Hollowell, *Business Object Design and Implementation.* London: Springer.
- 25 Schwaber, K., & Sutherland, J. (2017). *Der Scrum Guide.* Abgerufen am 16. 10 2018 von The Scrum Guide:
26 <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>
- 27 Vahs, D. (2007). *Organisation: Einführung in die Organisationstheorie und -praxis (6., überarb. u. erw. Aufl.*
28 *Ausg.).* Stuttgart: Schäffer-Poeschel.
- 29 Vahs, D., & Weiland, A. (2010). *Workbook Change Management: Methoden und Techniken.* Stuttgart:
30 Schäfer-Poeschel.

- 1 Vogels, W. (04. 05 2006). A Conversation with Werner Vogels. *ACM Queue*.
- 2 Westphal, F. (2006). *Testgetriebene Entwicklung - mit JUnit & FIT* (1. Auflage Ausg.). Heidelberg:
3 dpunkt.verlag GmbH.
- 4 Wiest, S. (2011). *Continuous Integration mit Hudson*. Heidelberg: dpunkt.verlag GmbH.
- 5 Womak, J. P., Jones, D. T., & Roos, D. (1992). *Die zweite Revolution in der Autoindustrie*. Frankfurt / New
6 York: Campus Verlag.
- 7 Womak, J., & Jones, D. (2013). *Lean Thinking* (3., aktualisierte und erweiterte Auflage Ausg.).
8 Frankfurt/New York: Campus Verlag.
- 9