Sascha Block

Large-Scale Agile Frameworks

Agile Frameworks, agile Infrastruktur und pragmatische Lösungen zur digitalen Transformation

EBOOK INSIDE

Springer Vieweg

Large-Scale Agile Frameworks

Sascha Block

Large-Scale Agile Frameworks

Agile Frameworks, agile Infrastruktur und pragmatische Lösungen zur digitalen Transformation

Springer Vieweg

Sascha Block
Hamburg, Deutschland

ISBN 978-3-662-62047-2 ISBN 978-3-662-62048-9 (eBook)
https://doi.org/10.1007/978-3-662-62048-9

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg

Planung: David Imgrund
Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature.
Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Vorwort

Mein Entschluss, dieses Buch zu schreiben, reifte im Zusammenhang mit meiner Masterarbeit an der Universität Hamburg, die sich mit der Eignung der Large-Scale Agile Frameworks als Organisationsmodell für Software-Hersteller auseinandersetzt. Schon der Entschluss, noch in „fortgeschrittenem Alter" einen Masterstudiengang zu absolvieren, war maßgeblich davon geprägt, dass ich in meiner Berufspraxis in vielen Belangen agile Organisationsformen vermisst habe. Dabei waren es nie die einzelnen Teams oder Kollegen, die solcher Methodik entgegenstanden – sogar ganz im Gegenteil!

Regelmäßig war festzustellen, dass Best Practices und ausgereifte agile Organisationsmodelle fehlen oder schlicht unbekannt waren. Ohnehin war ich fest entschlossen, mich genau mit diesem Themenkomplex der Large-Scale Agile Frameworks auseinanderzusetzen. Bis heute nehme ich wahr, wie schwer es Unternehmen fällt, agile Arbeitsweisen ganzheitlich in ihrer Organisation zu verankern.

Ich habe dieses Buch entwickelt, um dabei zu unterstützen, Agilität als festen Bestandteil unserer modernen Arbeitskultur in Unternehmen und Organisationen jeder Art zu etablieren. Losgelöst davon, in welchem Reifegrad Sie sich oder Ihre Kolleg*innen und Mitarbeiter*innen aktuell befinden, soll dieses Buch als Leitfaden dabei unterstützen, agiles Mindset und agile Arbeitsweisen in Ihrem Berufsalltag zu verankern. Unabhängig davon, ob Sie als Führungskraft oder Teammitglied mehr über agile Methodik erfahren wollen, soll Ihnen dieses Werk als Kompass dienen und navigiert Sie entlang an den zurzeit brandaktuellen Themen wie Cloud-Technologien, DevOps und IT-Security. Ein weiteres Ziel ist es, das Bewusstsein für IT-Sicherheit zu schärfen und praktikable Strategien und Lösungen zur Umsetzung aufzuzeigen. Es ist mir ein Anliegen, dabei zu helfen, Ihre Kenntnisse über die Grundlagen von Agilität hinaus zu erweitern. Das vorliegende Fachbuch wurde auf der Grundlage jahrelanger Erfahrung in unzähligen IT-Projekten und vor dem Hintergrund verschiedenster Unternehmensumgebungen erstellt. Es gibt viele Bücher, die in Agilität und die dazu gehörigen Grundlagen einführen. Mein Plan war es, ein Buch zu schreiben, das darüber hinausgeht. Dieses Buch soll Sie und Ihre Organisation effektiv dabei unterstützen, agil durchzustarten und eine neuartige Organisationsform zu etablieren, damit Ihnen gemeinsam mit Ihren Kolleg*innen und Teams – und vor allem mit Spaß – eine erfolgreiche digitale Transformation gelingt!

Der Zeitraum, in dem dieses Buch entstanden und gewachsen ist, wurde von der Corona-Pandemie geprägt; nicht nur die immense Bedeutung, sondern vor allem auch die Dringlichkeit agiler Methodik wurde noch einmal verschärft. Als IT-Architekt im Kontext der Telematikinfrastruktur sind die Ergebnisse meiner täglichen Arbeit stark von vielen Einzelpersonen und Teams abhängig, wobei externe Organisationen, rechtliche Rahmenbedingungen und neue Themenfelder – in und abseits von Software-Architekturen – eine maßgebliche Rolle spielen.

In diesem Zusammenhang durfte ich in dieser prägenden Zeit die Einführung zahlreicher Collaboration Tools mitgestalten und erleben, wie sich diese positiv auswirken; zudem konnte ich lernen wie wichtig das Feedback des Einzelnen und der Gemeinschaft ist und wie sehr ein agiles Mindset geeignet ist, Herausforderungen im Team zu meistern.

Weil ich durch meine Arbeit als IT-Architekt weiß, wie wichtig visuelle Darstellungen für die Entwicklung eines gemeinsamen Verständnisses – gerade auch über einzelne Teams hinaus – sind und wie wirksam grafische Artefakte für eine erfolgreiche Gestaltung akzeptierter Lösungen sind, habe ich dieses Buch um zahlreiche Darstellungen erweitert. Bekanntlich sagt ein Bild oft mehr als 1000 Worte …

Dieses Buch ist das Ergebnis unzähliger Iterationen, beeinflusst durch immer neue Ideen, sei es während meiner täglichen Arbeit als IT-Architekt und im Dialog mit Kolleg*innen oder mit Freund*innen und anderen IT-Spezialist*innen auf zahlreichen Onlineveranstaltungen und Meetups.

Aufbau des Buchs

Das Buch strukturiert das Problem im Kontext agiler Software-Entwicklung und richtet dabei den Fokus auf Large-Scale Agile Frameworks, die den besonderen Anforderungen von Unternehmen gerecht werden, die mit einer Vielzahl agiler Teams in diversen, oft parallel verlaufenden, Digitalisierungsprojekten konfrontiert sind.

Teil I – Digitale Transformation mit Large-Scale Agile Frameworks
Praxisnahe Methodik basierend auf realer Projekterfahrung

Kap. 2 schildert die Problemstellungen mit denen Projektteilnehmer und Stakeholder bei der digitalen Transformation konfrontiert sind. Die **agile Priorisierung** ist dabei regelmäßig eine Herausforderung für alle Beteiligten.

Über klar definierte **Ziele der digitaler Transformation** und dem Wechsel in agile Arbeitsweisen wird die Bedeutung agiler Prozesse und der Large-Scale Agile Frameworks dargestellt.

Agile Konzepte und grundlegende Begriffe werden erläutert und im Kontext von Multi-Projekt-Management und Portfolio Management beleuchtet.

Software-Lizenzprodukte und **Individualsoftware** sind von einer **Software-Plattform**, der **Software-Produktfamilie** und einer **Software-Produktlinie** abzugrenzen, weil hier anforderungsseitig unterschiedliche Rahmenbedingungen zu berücksichtigen sind. Der **Produktlebenszyklus von Software** sowie **Software Releases**, das

Release-Management, die **Software-Architektur** und **Wissensmanagement** sowie DevOps haben wesentlichen Einfluss auf die agile Software-Entwicklung.

Agile Konzepte und das **agile Projektmodell** werden im Kontext von **Scaled-Agile** und **Large-Scale Agile**, Agile Teams, deren Rollen, Aufgaben und Prozesse dargestellt.

Kap. 3 führt in das Konzept der Large-Scale Agile Frameworks ein und stellt die relevantesten und in der Praxis bewährtesten Modelle vor, zeigt deren Unterschiede und Grenzen auf.

Kap. 4 bietet eine praxisorientierte Unterstützung bei der Einführung eines Large-Scale Agile Frameworks in einer Organisation. Mit der **Methode des *Action Design Research*** steht Ihnen ein moderner Ansatz zur praxisorientierten Problemlösung in Organisationen zur Verfügung.

Mit einer **faktenbasierten Zusammenfassung der relevantesten Einflussfaktoren des Cloud-Trends und der Virtualisierung** werden nicht nur hochaktuelle Technologiethemen berücksichtigt, sondern auch deren Auswirkungen in Bezug auf die Erfordernisse agiler Organisationsformen berücksichtigt.

Software-Architektur und IT-Security spielen eine zentrale Rolle für alle Aktivitäten der Digitalisierung. Vorgestellt werden nicht nur die Paradigmen der Ansätze **IT-Security-by-Design** und der **Zero-Trust-Strategie**, sondern auch effektives Secret-Management und der erweiterte Schutzbedarf virtueller Container-Umgebungen. Somit ergibt sich eine ganzheitliche Betrachtung.

Hierbei dürfen die Aspekte **von Public-Key-Infrastrukturen**, **Microservice-Architekturen**, **REST-APIs** und **RESTful-Design**, dem **Architektur-Pattern der Service Meshs** und darauf ausgerichtete Qualitätsmerkmale nicht außer Acht gelassen werden.

Auf Grundlage der **OWASP-Guidelines** lässt sich IT-Sicherheit nachweislich durch iteratives, begleitendes Pentesting verbessern. Penetration-Tests hingegen dokumentieren untersuchte Eigenschaften zu einem bestimmten Zeitpunkt unter definierten Rahmenbedingungen mitsamt vereinbarter Testkriterien und attestieren somit IT-Sicherheit in Bezug auf einen definierten Testumfang. Kombiniert ergänzen sich beide Maßnahmen wechselseitig als wirksame Mittel um Software zu härten, also die IT-Sicherheit zu erhöhen.

Mit der Methodik des **Threat Modeling** wird ein wirksamer Mechanismus erläutert, um Bedrohungen vorzeitig zu identifizieren und diese Sicherheitsrisiken dann systematisch zu eliminieren.

Sie erfahren nachvollziehbar, warum frühestmögliches, agiles Deployment und automatisierte Tests geeignete Mittel sind um robuste und **hinsichtlich der IT-Sicherheit gehärtete Software-Lösungen** zu entwickeln.

In diesem Kontext wird die relative junge Disziplin **DevSecOps** vorgestellt und mit den Arbeitsmitteln von **Code Reviews** und des **Pair Programmings** werden praxisorientierte Methoden zur Software-Entwicklung dargestellt.

Die oft außer Acht gelassene Erfordernis zum **Logging und Monitoring** hat in diesem Abschnitt den Fokus auf IT-Sicherheit und zu beachtende Datenschutzaspekte.

Zur Bewältigung dieser Herausforderungen werden die erforderlichen **agilen Teams, Rollen, Aufgaben und Prozesse** skiziert, die ein agiles Organisationsmodell erfordert. Mit der Vorstellung der jeweiligen Spezialisierungen wird eine verständliche und leicht nachvollziehbare Einordnung der einzelnen Fachdisziplinen gegeben. Gleichzeitig kann dieser Abschnitt Organisationen wirkungsvoll dabei unterstützen, eigene Teams und Rollenprofile zu schärfen oder zu ergänzen.

Im Zusammenhang von **Design Thinking** und **Prototyping** wird das erforderliche Grundlagenwissen zur prototypischen Vorgehensweise und Rapid Prototyping und das Prototypenphasenmodell vorgestellt.

Kap. 5: Agiles Priorisierungsmodell für Software-Hersteller reflektiert die Unternehmen in der Rolle eines Software-Herstellers anhand von drei bekannten Unternehmen und verdeutlicht somit die Tragweite und Bedeutung von Software-Entwicklung für Organisationen. In **Abschn. 5.2** werden die **Agilen Teams und Rollen die für Organisationen in der Rolle eines Software-Herstellers** typisch sind und in **Abschn. 5.3** die entsprechenden **Prozesse und Aktivitäten** präsentiert.

Auf Grundlage der Anforderungen und Ergebnisse der vorhergehenden Kapitel wird dann das *entwickelte agile Modell zur übergreifenden Priorisierung innerhalb unterschiedlicher Unternehmenskonstellationen* vorgestellt und begründet, warum dieses Modell geeignet ist, um die Anforderungspriorisierung und agile Software-Entwicklung für eine Vielzahl unterschiedlicher Software-Projekte in Einklang zu bringen.

Um Hürden wirksam zu begegnen, stellt **„Kap. 6 – Herausforderungen bei der Etablierung eines Large-Scale Agile Frameworks im Unternehmen“** entsprechende Maßnahmen vor, damit Sie auch in Ihrer Organisation die regulären Herausforderungen meistern, die einer effektiven Anwendung agiler Methodik entgegenstehen.

Mit der Etablierung von **Friendly User Tests** erfahren Sie, warum Entwicklungsstufen innerhalb einer Organisation nicht übersprungen werden können, sondern mit **agilen Reifegraden** genommen werden müssen. Außerdem werden Sie darüber lesen, wie **neuartige Entwicklungs- und Testumgebungen mit abstrahierter Hardware-Ebene** Ihnen neben zuverlässigen Tests auch solide Testergebnisse liefern. Auch die **Bedeutung transparenter Entscheidungsprozesse und eines agilen Anforderungsmanagements** werden dargestellt.

Abschn. 6.2 verdeutlicht, **wie und warum gutes Digital Leadership funktioniert**. Von der Vision zur Strategie wird das Konzept einer **Produkt-/Service-Roadmap** vorgestellt. Überdies werden die Rollen des **Chief Information Officer (CIO)** und die neue Führungsrolle des **Chief Digital Officer (CDO)** detailliert erklärt.

Ebenso sind die wesentlichen Punkte des **Change-Managements** erläutert, die gerade während der Einführung neuer Organisationsstrukturen relevant sind. Hierzu zählen insbesondere die für die Software-Entwicklung wesentlichen Verfahren des **Change-Request- und Release-Managements**, die Etablierung agiler Fähigkeiten über eine Agile Academy und die Etablierung agiler Werte.

Die Ausführungen zum **Notfallmanagement** schließen den ersten Teil mit dem Weitblick auf agile und angemessene Reaktionen in Notfallsituationen ab.

Teil II – Agile Infrastruktur

Kap. 7 – Agile Tools: Werkzeugkasten für Product Owner & Agile Teams

Der zweite Teil des Buches widmet sich den agilen Tools und dient als Werkzeugkasten für Product Owner & Agile Teams.

Neben dem **Agilen Manifest** werden **agile Ziele**, die agilen Konzepte der **Personas**, **User Stories**, **Epics**, **Tasks** und **Sub-Tasks** und des **Backlog** erläutert.

Collaboration Tools sind die agilen Werkzeuge mit denen Stakeholder und die Mitglieder der unzähligen agilen Teams miteinander kooperieren und Informationen austauschen. Sie erfahren die Bedeutung und Funktionsweise der **Git-Repositories** sowie **OpenAPI** als ein **Werkzeug zum API-Design**.

Damit **agile Architekturen das Fundament softwarebasierter Digitalisierung** bilden, lernen Sie welche wichtigen Einflussfaktoren für eine agile Infrastruktur gelten.

Mit Abschn. 7.5 erhalten Sie einen **Leitfaden zur Erstellung einer pragmatischen Software-Architektur-Dokumentation** und erfahren, wie Sie **aussagekräftige visuelle Artefakte für Ihre Software-Architektur** erstellen und welche **Standards, Kriterien und Normen** sinnvoll sind.

Beantwortet wird auch: Wer sind die **Adressaten und** was sind die **Themenfelder der Software-Architektur Dokumentation**? Wie formulieren Sie aussagekräftige Fragestellungen für die wesentlichen Software-Architektur-Entscheidungen? Auch die Relevanz der **technischen Schulden** ist dargestellt. Das **Arc42-Template** liefert Ihnen abschließend ein **direkt nutzbares Handwerkszeug für die Software-Architektur-Dokumentation**. Und die Norm der **ISO/IEC 25010** definiert die **Qualität von Software**.

In Abschn. 7.6 werden **DevOps-Methoden** und **DevOps-Tools** mit der Überleitung zu **BizDevOps** vorgestellt.

Der Abschn. 7.7 stellt **Content** und die **Ziele einer Content-Strategie** vor und führt aus, welche Anforderungen an wertvolle Inhalte zu stellen sind und wie sich die **Wirksamkeit von Content messen** lässt. Der **Darstellung und Visualisierung von Content** widmen wir uns ebenfalls, und stellen die besonderen **Content-Formate der Infografiken und Dialogbilder** vor.

Im Abschn. 7.8 werden Möglichkeiten zur **Umsetzung eines Monitorings und Controllings** auf der Basis von **Key-Performance-Indikatoren (KPI)** erklärt und mit welchen Strategien und Werkzeugen sich ein solches Reporting umsetzen lässt.

Abschn. 7.9 stellt **Methoden und Werkzeuge zur agilen Priorisierung** vor, die für Product Owner und die agilen Teams im Umfeld der Software-Entwicklung eine echte Erleichterung bieten, um Anforderungen zu priorisieren und bestehende Abhängigkeiten zwischen Anforderungen erkennbar werden lassen. Sowohl **Timeboxing** als auch das **Feature-Driven Development** zählen hierbei zu den beliebtesten und bewährten Verfahrensweisen. Mit der Vorstellung von **Liquid Democracy** lernen Sie ein agiles Abstimmungsverfahren kennen und erfahren wie sie Transparenz innerhalb einer Organisation gemeinsam leben können.

Den Abschluss bildet das Kap. 8, das sich der Datenqualität und damit dem Lebenselixier der Digitalisierung widmet. Von den Grundlagen und Voraussetzungen für eine

exzellente Datenqualität, geht es zu Techniken der Datenbereinigung bis hin zu wirksamen Kontrollmechanismen zur Optimierung der Datenqualität.

Danksagung
Mein persönlicher Dank gilt insbesondere der mentalen Unterstützung meiner Frau Martina und der Geduld unserer Tochter Coco. Mein besonderer Dank richtet sich an meine, zur Entstehungsphase dieses Buchs, nach langer Krankheit verstorbene Mutter, die mich stets von ganzem Herzen bei meinen Zielen unterstützt und bestärkt hat. Aileen und Jean Paul soll dieses Buch auf Ihrem Weg begleiten und zeigen, dass jedes große Ziel – auch in kleinen Schritten – gelingt.

Ihre Fragen & Ihr persönliches Feedback ist gefragt!
Feedback von unseren Lesern ist immer herzlich willkommen!

Weil mir persönlich ein agiles Mindset, Transparenz und der offene Dialog sehr am Herzen liegen, bitte ich Sie als Leser unbedingt jegliches Feedback an mich als Autor dieses Werks zu richten.

Wenn Sie Fragenzu irgendeinem Aspekt dieses Buches haben, senden Sie mir bitte eine eMail an info@large-scale-agile-frameworks.com

Errata: Obwohl ich jede mir mögliche Sorgfalt darauf verwendet habe, die Richtigkeit meiner Inhalte zu gewährleisten, können Fehler passieren. Wenn Sie einen Fehler in diesem Buch gefunden haben, sind wir Ihnen dankbar, wenn Sie uns diesen melden würden.

Selbstverständlich freuen wir uns ebenso über Ihr allgemeines Feedback: Senden Sie uns eine eMail und erwähnen Sie den Titel des Buches im Betreff Ihrer Nachricht.

Aktuelle Informationen finden Sie ebenfalls auf der Website www.large-scale-agile-frameworks.com

Teilen Sie Ihre Gedanken in Form einer Buch-Rezension !
Wenn Sie Large-Scale Agile Frameworks – Agile Frameworks, agile Infrastruktur und pragmatische Lösungen zur digitalen Transformation gelesen haben, würden wir gerne Ihre Meinung dazu hören!

Ihre Rezension ist wichtig für uns und die Tech-Community und hilft uns, sicherzustellen, dass wir exzellente Qualitätsinhalte liefern.

Hamburg, Deutschland Sascha Block

Inhaltsverzeichnis

Abkürzungsverzeichnis

ADR	Action Design Research – Bitte ergänzen: API - Application Programming Interface
ART	Agile Release Train
AWS	Amazon Web Services
B2B	Business to Business
B2C	Business to Customer
BizDevOps	Business, Development & Operations
BL	Bereichsleitung
BSI	Bundesamt für Sicherheit in der Informationstechnik – bitte ergänzen: CI – Configuration Item
CR	Change Request
CWA	Closed World Assumption – bitte ergänzen: DDD – Domain Driven Design
DEV	Development (Entwicklung)
DevOps	Development & Operations
DoS	Denial of Service
DSVGO	Datenschutzgrundverordnung
EAM	Enterprise Architecture Management
FAQ	Frequently Asked Questions
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastruktur als Dienstleistung
IAM	Identity Access Management
IDLE	Infrastruktur Dienstleistungen
IoT	Internet of Things
IT	Informationstechnologie
KPI	Key-Performance-Indikatoren
LDAP	Lightweight Directory Access Protocol
MPM	Multi-Projekt-Management
MVC	Model View Controller
MVP	Minimum Viable Product
NIST	National Institute of Standards and Technology

OLAP	Online Analytical Processing
OLTP	Online Transcation Processing
OPS	Operations (IT-Betrieb)
OWA	Open World Assumption
PaaS	Plattform-as-a-Service
PI	Programminkremente
PJM	Projektmanager
PM	Produktmanagement
PO	Produkt Owner
RTE	Release Train Engineer
SA	Software-Architekt*in
SaaS	Software-as-a-Service
SAFe	Scaled Agile Framework
TLS	Transport Layer Security – bitte ergänzen: TOM – Technisch organisatorische Maßnahmen
UML	Unified Modeling Language
UX	User Experience – bitte ergänzen: VCS – Versioning Control System
Ver	Vertrieb
VMM	Virtual Machine Monitor
XP	Extreme Programming
XML	Extensible Markup Language

Abbildungsverzeichnis

Tabellenverzeichnis

Teil I

Digitale Transformation mit Large-Scale Agile Frameworks

Einleitung

1

Dieses Buch stellt praxisnahe Lösungen zur übergreifenden Priorisierung von Anforderungen und Dokumentation am Beispiel einer Organisation in der Rolle oder mit wesentlichen Merkmalen eines Software-Herstellers dar. Dabei wird das Zusammenspiel aktueller Technologiethemen wie des Cloud-Trends oder die organisatorischen Anforderungen im Hinblick auf Microservices reflektiert. Unter der Erfordernis kundenzentrierter und serviceorientierter Produkte und Dienstleistungen sind Organisationen zunehmend damit konfrontiert, ihre IT-Strategie eng an den Bedürfnissen ihrer Kunden auszurichten. Zudem erfordern verschärfte Anforderungen in Richtung IT-Security und im Kontext der digitalen Transformation mittels Cloud-Technologien und Containerisierung eine radikale Neuausrichtung bisheriger IT-Strategien.

Unternehmen kämpfen im Umfeld unberechenbarer Märkte unter zunehmendem Konkurrenzdruck, sich schnell ändernden Kundenanforderungen, regulatorischen Änderungen und einem rasanten technologischen Fortschritt, um den bestmöglichen Unternehmenserfolg zu erreichen. Unter solchen Bedingungen gelten langfristig ausgelegte Strategiepläne eingeschränkt und bedürfen der fortlaufenden Kontrolle und agiler Anpassungsmechanismen, um Hersteller von B2C- oder B2B-Produkten oder Dienstleister jedweder Branche in einem derart schwierigen Marktumfeld bestmöglich zu unterstützen.

Nicht nur ein schonender, sondern überdies weitgehend flexibler Umgang mit den zur Verfügung stehenden, begrenzten Ressourcen entspannt die Wettbewerbssituation maßgeblich. Wenn Software sich über für Kunden relevante digitale Services definiert und agil an den realen Bedürfnissen der Kunden orientiert, dann wandelt sich IT zum Enabler im Unternehmen und trägt maximal zum Unternehmenserfolg bei.

Die Software-Entwicklung gilt nicht nur selbst als komplexes Problemfeld, sondern auch der Wert von Software ist mitunter schwer zu verifizieren. Dabei muss Software die strategischen und wirtschaftlichen Interessen eines Unternehmens so weitreichend, einfach und kostengünstig wie möglich abdecken. Der Komplexitätsgrad von Software steigt

S. Block, *Large-Scale Agile Frameworks*,
https://doi.org/10.1007/978-3-662-62048-9_1

mit den heterogenen Interessen unterschiedlicher Kundengruppen. Somit werden individualisierte Software-Lösungen, die optimal auf spezialisierte Kundeninteressen ausgerichtet sind, und deren agiles Handling unumgänglich.

Die Wirtschaftlichkeit optimierter Verfahrensweisen steht im scheinbaren Widerspruch zur Individualisierung von Software. Je individueller ein an den Markt gerichtetes Angebot ausgestaltet ist, desto komplexer und aufwendiger werden die Anforderungen und umso relevanter wird die Notwendigkeit einer übergreifenden Priorisierung. Auf diesem Weg lassen sich erwünschte Skaleneffekte durch strukturierte digitale Services, Wirtschaftlichkeitsaspekte hinsichtlich der entsprechenden Ressourcenermittlung zur Umsetzung von Software, das Bedürfnis einer handhabbaren Wartbarkeit, höchste Ansprüche an die IT-Security und eine möglichst kurze Time-to-Market verwirklichen. Wie wichtig agiles und wissenschaftsbasierten Handeln ist, zeigt sich in den Phasen der jüngsten Corona-Krise und im Kontext nationaler und europäischer Sicherheitsbemühungen um kritische Infrastrukturen umso mehr.

Welche besonderen Anforderungen ein differenziertes Angebot von Software Services stellt und inwiefern agile Prinzipien geeignet sind, dieses Problem wirksam zu lösen, wird in diesem Buch analysiert und anhand praxisbewährter agiler Methodik nachvollziehbar erläutert. Neben der Betrachtung des Produktlebenszyklus und einer grundlegenden Klassifikation von Software werden dabei insbesondere die hierzu erforderlichen Prozesse, Aktivitäten und Rollen beleuchtet. Auf Grundlage dieser Anforderungen werden geeignete agile Frameworks ausgewählt, vorgestellt und hinsichtlich möglicher Schwächen und Stärken untersucht.

Dabei stellt das vorliegende Buch den engen Zusammenhang und das Optimierungspotenzial entlang von Software-Produktlinien dar. Eine Software-Produktlinie ist eine Gruppe von Software-Produkten, die Varianten eines Basisprodukts darstellen und über eine gemeinsame Software-Architektur verfügen.

Damit steht die Betrachtung im Kontext des sogenannten Large-Scale Agile Development. Large-Scale Agile Development wird relevant, sobald es darum geht, eine agile Vorgehensweise auf die Strukturen größerer Organisationen auszuweiten. Auf diese Weise können Projekte mit großen Teams und einer Vielzahl unterschiedlichster Projekte entsprechend den Prinzipien der agilen Entwicklung – über die Grenzen der einzelnen agilen Teams – auf die ganze Organisation ausgeweitet werden.

Motivation und Ziele

Das Buch soll Organisationen bei einer effektiven digitalen Transformation unterstützen. Hierzu werden geeignete Vorgehensmodelle, Methoden und agile Tools beschrieben und erläutert, so dass Unternehmen ihre Organisationsstruktur nachhaltig in Richtung einer übergreifenden, agilen Priorisierung verändern können. Mit solchen agilen Organisationsstrukturen, auch als Large-Scale Agile Frameworks bekannt, wird eine jederzeit anpassungsfähige Umsetzung von beispielsweise cloudbasierten Microservices und die Priorisierung der in agilen Teams organisierten Unternehmenseinheiten bestmöglich unterstützt.

Im Rahmen von – an Beispielen und Anwendungsfällen orientierten Empfehlungen – erfolgt eine praxis- und wissenschaftsorientierte Analyse hinsichtlich der übergreifenden Anforderungspriorisierung zwischen produktspezifischen Entwicklerteams und cross-

funktionalen Teams mit übergreifenden Unterstützungsfunktionen sowie produktübergreifenden Expertenteams mit spezialisierten Aufgaben.

Neuartig ist die Reflexion brandaktueller technischer Themen und die Berücksichtigung des Erfordernisses bestmöglicher IT-Sicherheit in Bezug auf die beschriebenen Technologien und hinsichtlich der Auswirkung auf agile Organisationsmodelle und Priorisierungsmodelle.

Im Ergebnis wird ein geeignetes agiles Organisations- und Priorisierungsmodell vorgestellt, das Organisationen, die sich am Beginn oder bereits mitten im Prozess der agilen Transformation befinden, effektiv unterstützt, um die serviceübergreifende Organisation und Priorisierung zu verbessern.

Mit Ausrichtung auf die effiziente Gestaltung von agilen Software Services stehen die Optimierung, Dimensionen der technischen Rahmenbedingung, der fachlichen Dokumentation und der Interprozesskommunikation im Fokus eines solchen Modells.

Für wen ist dieses Buch?

Ich habe dieses Buch entwickelt, um dabei zu unterstützen, Agilität als festen Bestandteil unserer modernen Arbeitskultur in Unternehmen und Organisationen jeder Art zu etablieren. Losgelöst davon in welchem Reifegrad Sie, Ihre Kolleg*innen und Mitarbeiter*innen sich aktuell befinden, soll dieses Buch als Leitfaden dabei unterstützen, agiles Mindset und agile Arbeitsweisen in Ihrem Berufsalltag zu verankern. Losgelöst davon, ob Sie als Führungskraft oder Teammitglied mehr über agile Methodik erfahren wollen, soll Ihnen dieses Werk als Kompass dienen und navigiert dabei entlang der zurzeit brandaktuellen Themen wie Cloud-Technologien, DevOps und IT-Security. Ein weiteres Ziel ist es, das Bewusstsein für IT-Sicherheit zu schärfen und praktikable Strategien und Lösungen zur Umsetzung aufzeigen. Es ist mir ein Anliegen, dabei zu helfen, ihre Kenntnisse über die Grundlagen der Agilität hinaus zu erweitern. Es wurde auf der Grundlage jahrelanger Erfahrung in unzähligen Projekten und vor dem Hintergrund verschiedenster Unternehmensumgebungen erstellt. Es gibt viele Bücher, die in Agilität und die dazugehörigen Grundlagen einführen. Mein Plan war es, ein Buch zu schreiben, das darüber hinausgeht. Dieses Buch soll Sie und Ihre Organisation effektiv dabei unterstützen, agil durchzustarten und eine neuartige Organisationsform zu etablieren, damit Ihnen gemeinsam mit Ihren Kolleg*innen und Teams – und vor allem mit Spaß – eine erfolgreiche digitale Transformation gelingt!

Vorgehensmodelle, Frameworks und Standards

Welche Vorgehensmodelle, Frameworks und Standards sind ideal geeignet, um agile Software-Projekte erfolgreich zum Abschluss oder in die nächste Prozessphase zu überführen?

Ein Vorgehensmodell beschreibt, wie ein Team innerhalb eines Projekts eine Aufgabe löst und wie das Team hierzu effektiv kooperiert. Wichtig ist dabei, dass das Vorgehen wiederholbar erfolgreich ist, so dass vergleichbare Projekte in ähnlicher Form – in Bezug auf Kosten, Qualität und Zeit – optimiert angegangen werden können.

Dabei legt ein Vorgehensmodell drei wesentliche Elemente fest [1]:

- **Rollen**
- **Artefakte/Produkte**
- **Aktivitäten und Workflows**

Sind Anforderungen am Anfang eines Projekts vollständig bekannt, gelten Wasserfallmodelle als bewährtes Vorgehensmodell, sofern auch keine Erfordernis im weiteren Projektverlauf besteht, diese Anforderungen zu verändern [1, Seite 759].

Eine zweite wichtige, im Projektkontext nicht gegebene Voraussetzung zur Anwendung des Wasserfallmodells ist, dass ein erfahrenes Projektteam mit einer etablierten Technologie die Entwicklung übernimmt. Zusätzlich sind Änderungskosten im Projektverlauf ein relevantes Entscheidungskriterium.

Das Schaubild 1.1 „Änderungskosten im Projektverlauf“ illustriert schematisch, welche Höhe Änderungskosten im Projektverlauf haben, je später diese Anpassungserfordernisse erkannt werden [2, Seite 71/72].

Um die genannten Nachteile teilweise zu kompensieren, können Projekte iterativ umgesetzt werden, mit dem Ziel, Anpassungserfordernisse früher zu erkennen und frühzeitig vom Nutzen der Teilergebnisse zu profitieren. Auch Projekte nach dem Wasserfallmodell können – statt in einem einzigen großen Wasserfall – in Stufen durchgeführt werden. Hierbei entspricht jede Stufe wiederum einem kleinen Wasserfall. Von Stufe zu Stufe können sich Anforderungen ändern und Gelerntes kann umgesetzt werden. Auch das Feedback

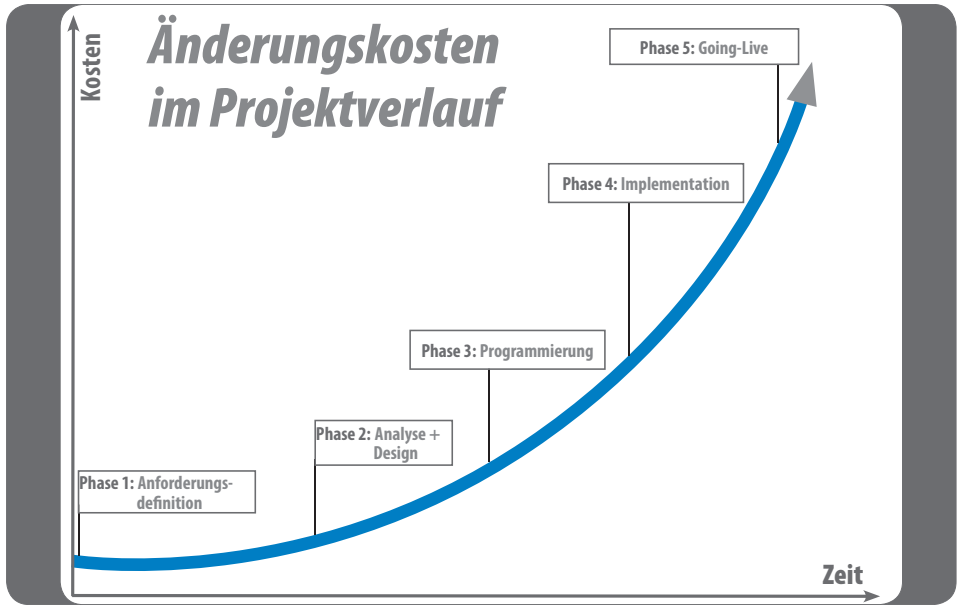

Abb. 1.1 Änderungskosten im Projektverlauf. (Quelle: Sascha Block – Eigene Darstellung)

vom Kunden erfolgt bereits nach der ersten Stufe und nicht erst am Projektende. Jede Stufe kann als eigenes Projekt zum Festpreis behandelt werden.

Mit immer kürzeren Änderungsintervallen von Anforderungen bietet sich inzwischen in den meisten Fällen ein iteratives, prototypisches Vorgehensmodell an.

Bei iterativen Vorgehensmodellen wird ein Projekt in mehreren, kleinen Iterationen organisiert, siehe Abb. 1.2 „Iteratives Projektmodell“. Nach dem iterativen Modell werden somit mit jeder Iteration vollständig alle Phasen durchlaufen.

Von Iteration zu Iteration ist Lernen im Team möglich. Auch Feedback des Kunden findet schnell Berücksichtigung, da bereits nach der ersten Iteration eine funktionierende Software zur Verfügung steht. Damit ist das Projektrisiko für alle Projektpartner geringer, da das Projekt nach jeder Iteration abgebrochen werden kann [1, Seite 760]. In jüngeren iterativen Vorgehensmodellen ist diese Vorgehensweise explizit als Build-Measure-Learn-Zyklus bzw. -Prinzip enthalten.

Zusätzlich wird bei iterativem Vorgehen zwischen evolutionärer und inkrementeller Entwicklung unterschieden. Wird **inkrementell** entwickelt, wächst die Software von Iteration zu Iteration. Auch der Funktionsumfang wird fortlaufend ergänzt, Anforderungen hingegen bleiben weitgehend unverändert.

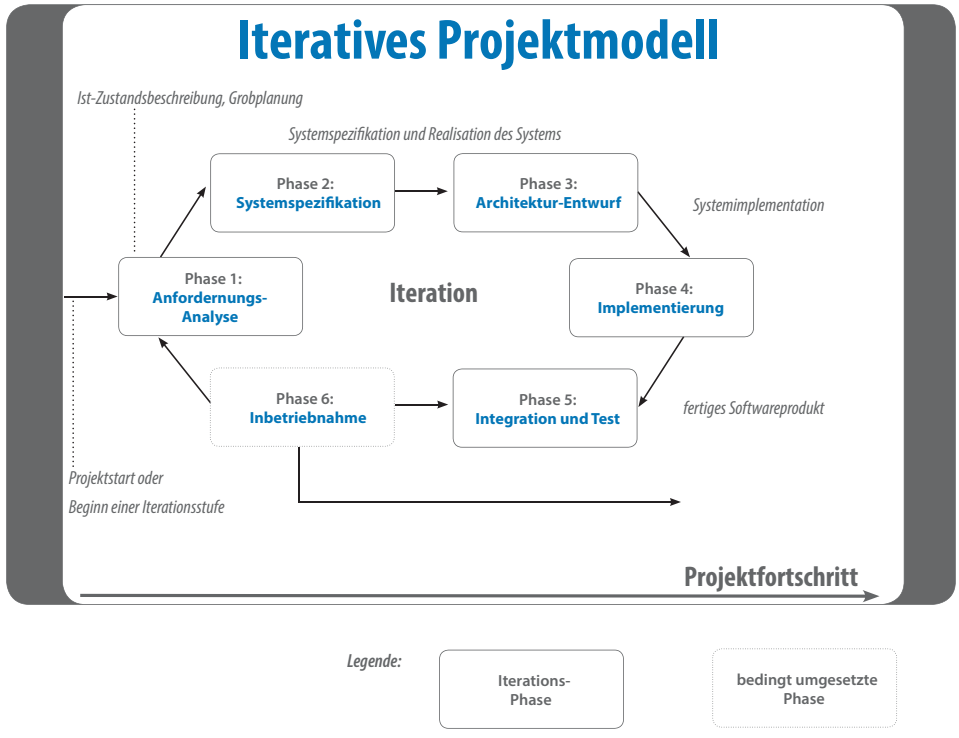

Abb. 1.2 Iteratives Projektmodell. (Quelle: Sascha Block – Eigene Darstellung)

Bei **evolutionärer Iteration** von Software-Lösungen hingegen, können sich auch Spezifikationen und Anforderungen in der Software von einer Iteration zur nächsten Iteration ändern. Somit wird mit der evolutionären Iteration – daher auch der Name dieses Iterationsmodells – auch eine komplette Restrukturierung von Software-Systemen möglich, wenn die Architektur einer Software geänderten Anforderungen nicht genügen kann.

Für Ausgangssituationen mit unklaren Projektvorgaben bieten sich agile Vorgehensmodelle wie Scrum an. Scrum nimmt als Prozessmanagement-Framework einen sehr engen Bezug auf die in einem Projekt geschaffenen Produktartefakte und gibt dafür ein Verfahren zur Verwaltung und Planung von Anforderungen vor. Dabei wird das konkrete Vorgehen im laufenden Entwicklungsprozess vom Projektteam festgelegt [3, Seiten 3 und 4].

Im Unterschied zu plangetriebenen Vorgehensmodellen steht das Team selbst in der Verantwortung für das Ergebnis und lernt somit im konkreten Vorgehen eigenständig, wie es am produktivsten arbeiten kann. Die als Sprints bezeichneten Iterationen dauern in der Regel längstenfalls 30 Tage an. Für jeden Sprint werden Anforderungen ausgewählt, die direkt umgesetzt werden. Somit ist auch die flexible und im Projektteam eigenständige Anpassung von Anforderungen von Sprint zu Sprint möglich. Anforderungen werden nicht zwingend schriftlich, sondern vielfach in persönlichen Gesprächen mit den Stakeholdern definiert. Agile Verfahren nutzen Produktinkremente oder Prototypen, die in kurzen, regelmäßigen Intervallen zur Beurteilung an die Projektkunden ausgeliefert werden. Wertvolles Feedback steht so dem Projektteam stets unmittelbar und direkt zur Verfügung.

Dieses Buch folgt der Motivation, darüber hinausgehende Mechanismen zur Synchronisierung und Priorisierung von Anforderungen zwischen agilen Teams, die dazu förderlichen Methoden, Prozesse und Tools sowie agile Konzepte der Large-Scale Agile Frameworks vorzustellen.

Agile Prozesse und Wissensmanagement
Wie lassen sich agile Prozesse zur Software-Entwicklung einfach, verständlich und transparent gestalten und pragmatisch dokumentieren?

Anforderungen an den Product Owner
Welche Anforderungen sollte ein Product Owner erfüllen und welche Methodik bewährt sich bei wiederkehrenden Aufgaben und zur agilen Dokumentation?

Literatur

1. Ernst, H., Schmidt, J., & Beneken, G. (2016). *Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis. – Eine umfassende praxisorientierte Einführung* (6. Aufl., S. 757). Springer/Vieweg. hier Kapitel 17.4 „Vorgehensmodelle“.
2. Schoeneberg, K.-P. (Hrsg.). (2014). *Komplexitätsmanagement in Unternehmen: Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern* (1. Aufl.). Springer/Gabler.
3. Schwaber, K., & Sutherland, J. (2017). „The Scrum Guide“ – Edition November 2017. http://www.scrumguides.org. Zugegriffen am 03.11.2022.