

Für das Verständnis des Buchs werden zunächst einige Begriffe definiert. Auf diese zentralen Begriffe wird im weiteren Verlauf immer wieder zurückgegriffen. Einige Definitionen zeigen neue Aspekte, die zu anderen Denkweisen in der Softwareentwicklung anregen sollen.

---

## 2.1 IT-Vorhaben

Das substantivierte Verb *vorhaben* beschreibt, dass „etwas verwirklicht werden soll“. In der einschlägigen deutschen Literatur zu den Themen Projektmanagement und Software Engineering wird der Begriff Vorhaben häufig mit Projekt gleichgesetzt. Korrekt ist, dass die Normenreihe DIN 69901 ein Projekt als Vorhaben beschreibt, dass sich durch die Einmaligkeit seiner Bedingungen auszeichnet. Diese Bedingungen werden als zeitliche, finanzielle und personelle Beschränkungen angesehen.<sup>1</sup>

Es gibt allerdings auch Vorhaben in der Softwareentwicklung, die durch wiederkehrende Tätigkeiten gekennzeichnet sind. Dazu gehört beispielsweise das Customizing von Formularen oder die Anpassung von Verfahren an neue gesetzliche Vorgaben.

Dieses Buch nutzt den Begriff IT-Vorhaben stellvertretend für Projekte in der Softwareentwicklung, aber auch für Kleinmaßnahmen im täglichen Geschäftsablauf. Beide Arten unterliegen beim Vorgehen identischen Prozessschritten.

Darum werden die Begriffe Vorhaben und IT-Vorhaben synonym verwendet.

---

<sup>1</sup> Vgl. Kütz (2009).

**Tab. 2.1** Klassifikation eines Vorhabens anhand der Codezeilen

Umfang (LOC)	Codezeilen (pro PM)	Aufwand (in PM)	Klassifikation
10.000	2000–25.000	< 1	Klein
100.000	1000–20.000	5	Mittel
1000.000	700–10.000	100	Groß
10.000.000	300–5000	2000	Sehr groß

## 2.2 Stakeholder

Im klassischen Verständnis der Betriebswirtschaftslehre setzt sich die Gruppe der Stakeholder einer Organisation aus Arbeitnehmern, Lieferanten, Kunden und der Öffentlichkeit zusammen, also aus allen internen und externen Personengruppen, die am Handeln der Organisation direkt oder indirekt ein Interesse haben<sup>2</sup>. Im Kontext eines IT-Vorhabens verhält sich der Stakeholder-Ansatz genauso. In einem IT-Vorhaben gehören zu den Stakeholdern alle internen und externen Personengruppen, die von dem Vorhaben direkt oder indirekt betroffen sind. Dazu gehören bspw. verschiedene Fachabteilungen, die Informationstechnik (IT), die Unternehmensleitung, das Entwicklungsteam oder die Projektleitung.

## 2.3 Geläufige Umfänge für Entwicklungsvorhaben

In der Literatur werden kleine, mittlere und große Entwicklungsvorhaben unterschieden. Die Größenangabe für ein Entwicklungsvorhaben ist eine abstrakte Größe, denn sie wird durch verschiedene Indikatoren bestimmt. Dies können Umsatz, Personentage (PT) oder Personenmonate (PM), Dauer, Anzahl Codezeilen u. v. a. sein. Interessant ist hierbei die Betrachtung der Entwicklungsvorhaben anhand der Codezeilen. Der Laie geht davon aus, dass ein Vorhaben mit 10-mal mehr Codezeilen auch 10-mal mehr Aufwand bedeutet. Tatsächlich nimmt der Aufwand eher exponentiell zu. Größere Vorhaben benötigen nämlich mehr Koordination zwischen größeren Personengruppen, womit gleichzeitig der Kommunikationsaufwand steigt.<sup>3</sup>

In der Softwareentwicklung herrscht somit der sog. Größennachteil vor: „... [Je] größer ein System wird, desto größer werden auch die Kosten je Einheit.“<sup>4</sup> McConnell<sup>5</sup> hat in

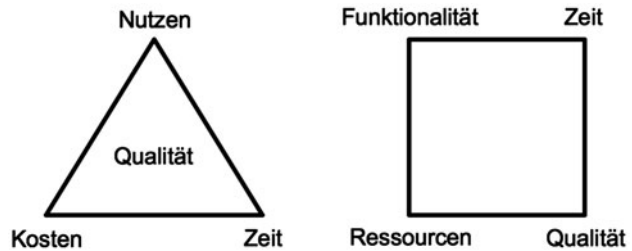
<sup>2</sup> Vgl. Wöhe und Döring (2008), S. 29 und S. 55.

<sup>3</sup> Vgl. Brooks (1995), S. 13–26; vgl. McConnell (2006), S. 91.

<sup>4</sup> Vgl. ebd.

<sup>5</sup> Vgl. McConnell (2006), S. 94.

**Abb. 2.1** Iron  
Projectmanagement Triangle.  
(In Anlehnung an: Appelo  
(2011), S. 185)



einer Untersuchung die generelle Beziehung zwischen der Größe eines Vorhabens und der Produktivität dargestellt. Daraus kann in Kombination mit den Ausführungen von Blum und Wörsdörfer<sup>6</sup> die in Tab. 2.1 pauschale Klassifizierung der Größe abgeleitet werden. Die Klassifizierung findet unabhängig von der Größe des Unternehmens und der Anzahl der verfügbaren Mitarbeiter statt, um so eine neutrale Bewertung anhand der Codebasis zu erstellen.

## 2.4 Projektdimensionen

Im Bereich des Projektmanagements wird häufig der Begriff *Iron Triangle* genutzt. Dieses Konzept zeigt, in welchen Dimensionen bzw. Grenzen sich ein Projekt bewegt. Die drei Größen oder Dimensionen Zeit, Kosten und Nutzen stehen in Konflikt zueinander und haben direkten Einfluss auf die daraus resultierende Größe Qualität.

Im Folgenden wird ein modifiziertes Konzept nach Appelo verwendet (vgl. Abb. 2.1). Das Dreieck wird zu einem Quadrat erweitert und der Qualitätsaspekt als eine direkt beeinflussbare Größe berücksichtigt. Dadurch sollen die in Kap. 3.4 ff. vorgestellten agilen Methoden in den Fokus gerückt werden, bei denen Qualität eine besondere Rolle spielt. Die Grundidee hinter dem veränderten Gedankenmodell: Verändert sich eine Größe in einer Ecke, so hat dies Einfluss auf die anderen Ecke. Zum Beispiel benötigen zusätzliche Funktionen entweder mehr Zeit oder mehr Ressourcen oder bewirken eine niedrigere Qualität. Ein Verlust von Ressourcen führt zu weniger Funktionen, einer geringeren Qualität oder einem verlängerten Zeitplan.<sup>7</sup> Dieses erweiterte Modell zur Darstellung der Dimensionen gilt auch für die Wartungsphase.

<sup>6</sup> Vgl. Blum und Wörsdörfer (2013), S. 55; bezogen auf ein mittelständisches Unternehmen.

<sup>7</sup> Vgl. Appelo (2011), S. 185 f.

## 2.5 Arten von IT-Vorhaben

In der Softwareentwicklung gibt es nicht nur eine Art von IT-Vorhaben. Das Buch soll zeigen, dass sich für die nachfolgend gelisteten Arten von Vorhaben ein hybrides Modell eignet<sup>8</sup>:

- **Neu-Entwicklung:** In einem Projekt zur Neu-Entwicklung geht es um die Schaffung von Programmen und Datenbanken. Es ist das klassische Projekt in der Softwareentwicklung und wird auch häufig als Individualentwicklung bezeichnet.
- **Wartung:** Ziel von IT-Vorhaben in der Wartungsphase ist die kontinuierliche Weiterentwicklung bestehender Anwendungen und Systeme.
- **Migration:** Bei Vorhaben zur Migration werden Programme und Datenbanken von einer Umgebung in eine andere versetzt. Das Ziel ist erreicht, wenn die Anwendungen mit gleicher Funktionalität in der neuen Umgebung laufen.
- **Sanierung:** Werden Programme und Datenbanken in der gleichen Umgebung in einen besseren Zustand versetzt, so wird diese Art des Vorhabens als Sanierungs- bzw. Reengineeringprojekt bezeichnet.
- **Einführung:** Soll Standardsoftware in einer Organisation eingeführt werden, so wird dies als Einführungsprojekt bezeichnet. Die Techniken und Methoden beschränken sich auf die Migration der alten Daten und die Schaffung von Schnittstellen zu benachbarten Systemen.
- **Integration:** Ziel eines Integrationsvorhabens ist, bestehende Anwendungssysteme miteinander zu verknüpfen. Läuft eine Transaktion ohne Medienbruch über alle Systeme hinweg, so ist das Ziel des Projekts erreicht.

In der Praxis sind Sanierung- oder Migrationsprojekte eher selten. Migrationen sind typische Herausforderungen von Herstellern einer Standardsoftware, denn Migrationen werden meist dann durchgeführt, wenn der Anwender sich von bestehender Hardware oder Software trennt. Sanierungsprojekte waren in der Softwareindustrie nie sonderlich populär, da sich ihr Nutzen kaum quantifizieren lässt<sup>9</sup>. Integrationsprojekte sind so komplex, dass sie sich in drei Projekte teilen lassen: ein Aufklärungsprojekt, ein Realisierungsprojekt und ein Evolutionsprojekt. Diese Unterprojekte nutzen teilweise eigene Vorgehensmodelle (siehe Kap. 3).

---

<sup>8</sup> Vgl. Kütz (2009), S. 299; vgl. Sneed (2004); vgl. Sneed (2007c), S. 59–60.

<sup>9</sup> Vgl. Sneed (2007c), S.60.

## 2.6 Prozessmodell und Vorgehensmodell

In der Literatur wird häufig als Abgrenzung zwischen Prozess- und Vorgehensmodell angeführt, dass Vorgehensmodelle keine Aussagen zur Organisation und zu den Verantwortlichkeiten treffen. Diese Abgrenzung zwischen Vorgehensmodell und Prozessmodell wird in den nachfolgenden Ausführungen nicht vorgenommen, da im Hybriden Vorgehensmodell auch Empfehlungen zur Organisation genannt sind. Für die in der Fallstudie zur DEVK genutzten Prozesskostenrechnung wird jedoch zwischen Prozess und Prozessmodell unterschieden.

### 2.6.1 Prozess

Für Becker ist ein „Prozess .. die inhaltlich abgeschlossene, zeitliche und sachlogische Folge von Aktivitäten, die zur Bearbeitung eines betriebswirtschaftlich relevanten Objektes notwendig sind.“<sup>10</sup> Bei Womack handelt es sich bei einem Prozess um eine „Reihe von einzelnen Operationen, die zur Entwicklung einer Konstruktion, zur Erledigung eines Auftrags oder der Herstellung eines Produkts benötigt werden.“<sup>11</sup>

Auch bei Schweitzer findet man eine ähnliche Definition: „Ein Prozess ist dadurch gekennzeichnet, dass er eine Folge von Aktivitäten (Vorgängen, Tätigkeiten, Arbeitsgängen) umfasst, die sich auf ein bestimmtes Arbeitsobjekt beziehen und bei erneutem Arbeitsvollzug an einem Arbeitsobjekt identisch wiederholt werden“<sup>12</sup> Neu an dieser Definition ist der in den anderen Definitionen fehlende Aspekt, dass ein Prozess identisch wiederholt wird.

Eine sehr detaillierte Definition findet man bei Hoffmann: „Formal wird jeder Prozess über eine Reihe von Aktionen definiert, die aus einer gegebenen Menge von Eingangsgrößen schrittweise eine Menge von Ausgangsgrößen bilden. . . Für die Durchführung einer einzelnen Aktion werden unterschiedliche Ressourcen benötigt, die sich auf der obersten Ebene in drei Kategorien einteilen lassen. Der ersten Kategorie werden alle produktspezifischen Ressourcen wie Quelltexte und Planungsdokumente zugeordnet. Die zweite Kategorie umfasst die Software- und Hardware-Infrastruktur und die dritte Kategorie wird durch das zur Verfügung stehende Personal gebildet.“<sup>13</sup> Abbildung 2.2 zeigt die wesentlichen Merkmale der Definition in einer schematischen Übersicht.

Gemeinsam haben alle genannten Definitionen die folgenden Merkmale:

- Eine Abfolge von Aktivitäten
- Veränderung eines Objekts oder Erstellung einer Leistung
- Definierten Input und definierten Output

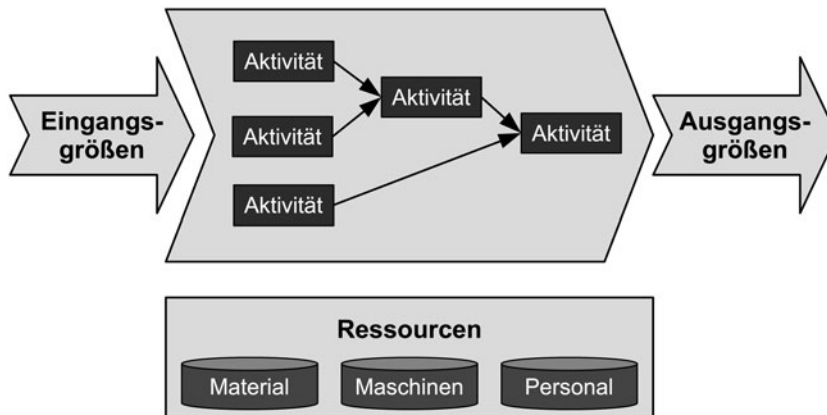
---

<sup>10</sup> Becker (2005), S. 6.

<sup>11</sup> Womack (2013), S. 408.

<sup>12</sup> Schweitzer (2008) S. 353.

<sup>13</sup> Hoffmann (2013), S. 491.



**Abb. 2.2** Definition eines Prozesses. (In Anlehnung an: Hofmann (2013), S. 492)

Da die Hoffmann'sche Definition alle Aspekte der anderen abdeckt und teilweise um weitere wesentliche Aspekte ergänzt, wird im Folgenden diese als geltende Definition verwendet.

## 2.6.2 Softwareentwicklungsprozess

In der englischen Literatur werden die Begriffe „software development process“ und „software process“ häufig synonym verwendet.<sup>14</sup> In diesem Buch wird dies ebenfalls getan.

Sommerville definiert einen Softwareentwicklungsprozess als „a set of activities that leads to the production of a softwareproduct.“<sup>15</sup> – also als eine Menge von Tätigkeiten, die zur Entwicklung eines Softwareprodukts führt. Nach seiner Aussage umfasst dies neben Erstellen einer völlig neuen Software auch das Erweitern und Anpassen von bestehenden Systemen und die Konfiguration und Integration von Standardsoftware oder Systemkomponenten.<sup>16</sup>

Münch definiert einen Softwareentwicklungsprozess hingegen als „a goal-oriented activity in the context of engineering-style software development.“<sup>17</sup>

Bei O'Regan findet sich die folgende Definition: „A software process is a set of activities, methods, practices, and transformations that people use to develop and maintain the software and the associated work products.“<sup>18</sup>

Neben Aktivitäten machen für ihn also auch Methoden, Praktiken und Transformationen einen Softwareentwicklungsprozess aus. Auch er betont, dass sowohl die Erstellung als

<sup>14</sup> Münch (2012), S. 8.

<sup>15</sup> Sommerville (2007), S. 64.

<sup>16</sup> Vgl. ebd.

<sup>17</sup> Münch (2012), S. 8.

<sup>18</sup> O'Regan (2011), S. 3.

auch die Wartung von Software unter diese Definition fällt. Gleiches gilt für dazugehörige Arbeitsprodukte.

Da im Rahmen dieses Buches nicht nur Entwicklungsprojekte, sondern auch Prozesse, die zur Wartung bestehender Software genutzt werden, betrachtet werden und dabei auch die zu erstellende Dokumentation analysiert wird, gibt die Definition von O'Reagan alle wesentliche Aspekte wieder und wird daher im Folgenden verwendet.

Nach Sommerville gibt es den idealen Softwareentwicklungsprozess nicht. Daher haben viele Unternehmen ihre eigene Herangehensweise an die Softwareentwicklung erarbeitet.<sup>19</sup> Trotz vieler unterschiedlicher Softwareentwicklungsprozesse gibt es einige grundlegende Aktivitäten, die alle Softwareentwicklungsprozesse gemeinsam haben<sup>20</sup>:

1. **Spezifikation:** Die Funktionalität einer Software und ihre Grenzen werden definiert.
2. **Entwurf und Implementierung:** Die Software wird gemäß ihrer Spezifikation erstellt.
3. **Validation:** Die Software wird validiert, damit sichergestellt ist, dass sie den Kundenanforderungen entspricht.
4. **Evolution:** Die Software wird weiter entwickelt, um sich ändernde Kundenanforderungen zu erfüllen.

Auch bei O'Reagan findet sich eine vergleichbare Einschätzung: „The processes employed in software development include processes to determine the requirements; processes to design and develop the software; processes to verify that the software is fit for purpose; and processes to maintain the software.“<sup>21</sup>

Wenn in diesem Buch der Begriff *Prozess* verwendet wird, ist im Allgemeinen ein Softwareentwicklungsprozess gemeint.

---

## 2.7 Flexibilität von Prozessen

Ein weiterer wesentlicher Begriff dieses Buches ist die Flexibilität. Kaluza definiert Flexibilität als „die Eigenschaft eines Systems proaktive oder reaktive sowie zielgerichtete Änderungen der Systemkonfiguration zu ermöglichen, um die Anforderungen von sich verändernden Umweltbedingungen zu erfüllen.“<sup>22</sup> Diese allgemeine Definition bezieht sich nicht nur auf Prozesse, sondern ist allgemeingültig, da sie den Begriff System verwendet.

Für Lorenz bedeutet Flexibilität hingegen „situationsadäquat Abweichungen von Standards zu ermöglichen, seien es Abweichungen vom üblichen Fachprozess, der Einsatz

---

<sup>19</sup> Vgl. Sommerville (2007), S. 64.

<sup>20</sup> Vgl. ebd.

<sup>21</sup> O'Regan (2011), S. 3.

<sup>22</sup> Kaluza (2005), S. 9.

abweichender Technologie oder die Umsetzung mit abweichender Vorgehensweise.“<sup>23</sup> Die Definition von Lorenz umfasst zwar Prozesse, beschränkt sich jedoch nicht ausschließlich auf diese. Er fasst unter dem Begriff der Flexibilität auch den Einsatz von Technologien zusammen, die vom Standard abweichen.

Die Definition von Becker konzentriert sich hingegen auf die Flexibilität von Prozessen: „Flexibilität ist ein weiteres Kennzeichen guter Prozesse, denn flexible Prozesse können auf unterschiedliche oder geänderte Anforderungen zeitnah reagieren.“<sup>24</sup> Für ihn sichert die Flexibilität „nicht nur eine Reaktionsgeschwindigkeit, sondern ermöglicht auch, sich auf unterschiedliche Situationen einzustellen.“<sup>25</sup>

Die Analysen, die in den späteren Kapiteln dieses Buches durchgeführt werden, sind unabhängig von den eingesetzten Technologien und fokussieren sich ausschließlich auf einen Prozess. Daher wird in diesem Buch die Definition von Becker verwendet.

---

## 2.8 Durchlaufzeit von Prozessen

Best stellt den Begriff Durchlaufzeit sehr allgemein als „den gesamten Zeitbedarf dar – vom Start- bis zum Endpunkt des Prozesses.“<sup>26</sup>

Eine konkretere Definition findet man bei Womack. Er definiert die Durchlaufzeit als die „benötigte Zeit zwischen Konzept und Einführung, Auftrag und Auslieferung, oder vom Rohmaterial bis in die Hände des Kunden.“<sup>27</sup>

Da diese Definition nicht nur die Umsetzung, sondern auch die relevanten Aspekte der Erstellung von Konzepten und der Auslieferung beinhaltet, wird sie in diesem Buch als gültige Definition verwendet.

Sowohl Best<sup>28</sup> als auch Jammernegg<sup>29</sup> unterteilen die Durchlaufzeit in die Warte- und die Bearbeitungszeit. Als Bearbeitungszeit bezeichnet Best „jene Zeitabschnitte der Durchlaufzeit, in der an der Lösung des Problems oder Fertigung des Produkts tatsächlich gearbeitet wird. Das Produkt wird sozusagen seiner Verkaufsfähigkeit näher gebracht.“<sup>30</sup> Das bedeutet, dass in den Bearbeitungszeiten wertschöpfende Aktivitäten stattfinden. Ihre Summe ist die Mindestdurchlaufzeit des Prozesses.<sup>31</sup>

---

<sup>23</sup> Lorenz (2012), S. 49.

<sup>24</sup> Becker (2008), S. 14.

<sup>25</sup> Ebd., S. 14.

<sup>26</sup> Best (2009), S. 79.

<sup>27</sup> Womack (2013), S. 405.

<sup>28</sup> Vgl. Best (2009), S. 80.

<sup>29</sup> Vgl. Jammernegg (2009), S. 220.

<sup>30</sup> Best (2009), S. 80.

<sup>31</sup> Vgl. Jammernegg (2009), S. 220.



Der zweite wesentliche Bestandteil der Durchlaufzeit ist die Wartezeit. „Wartezeiten ergeben sich immer dann, wenn der Vorgang unterbrochen wird, weil der jeweilige Mitarbeiter z. B. auf eine Information vom Kunden wartet oder das Produkt in der Fertigung bis zum nächsten Bearbeitungsschritt zwischengelagert wird, ohne dass dabei irgendein Fortschritt zu erkennen ist.“<sup>32</sup> Durch eine Reduzierung der Wartezeiten erreicht man kürzere Durchlaufzeiten. Dies führt nach Koch zu einer Erhöhung der Kundenzufriedenheit.<sup>33</sup> Nach Best entspricht die Summe aus Warte- und Bearbeitungszeit der Durchlaufzeit:

$$\text{Durchlaufzeit} = \text{Wartezeit} + \text{Bearbeitungszeit}$$

**Formel 1: Berechnung der Durchlaufzeit**

Vgl. Best (2009), S. 80

Daher ist es möglich, aus der Differenz von gemessenen Durchlauf- und Wartezeiten die Bearbeitungszeit zu berechnen.<sup>34</sup>

---

## 2.9 Die Prozesskostenrechnung

Die Prozesskostenrechnung ist eine Form der Kostenrechnung, die berücksichtigt, dass in vielen Unternehmen der Umfang an indirekten Leistungen, wie z. B. planende, steuernde und überwachende Tätigkeiten, zunimmt.<sup>35</sup>

Eine der Grundannahmen der Prozesskostenrechnung ist es, dass die Gemeinkosten der indirekten Leistungsbereiche nicht unmittelbar durch Produkte, sondern durch stellenübergreifende Prozesse verursacht werden.<sup>36</sup> Daher versucht die Prozesskostenrechnung, die Kosten aller betrieblichen Aktivitäten zu ermitteln und diese für die Kalkulation der Produkte nutzbar zu machen.<sup>37</sup>

Laut Jórasz sind die Ziele der Prozesskostenrechnung eine verursachungsgerechte Kalkulation, eine effiziente Planung und Kontrolle der Gemeinkosten sowie eine Erhöhung der Kostentransparenz in den indirekten Bereichen.<sup>38</sup> Sie kann sowohl auf Ist-, Normal- und/oder Plankosten basieren.<sup>39</sup>

---

<sup>32</sup> Best (2009), S. 80.

<sup>33</sup> Vgl. Koch (2011), S. 77.

<sup>34</sup> Best (2009), S. 80.

<sup>35</sup> Vgl. Jórasz (2008), S. 281.

<sup>36</sup> Vgl. Schweitzer (2008), S. 352 f.

<sup>37</sup> Vgl. Coenberg (2012), S. 162.

<sup>38</sup> Vgl. Jórasz (2008), S. 281.

<sup>39</sup> Vgl. Freidank (2008), S. 371.

### 2.9.1 Verwendung der Prozesskostenrechnung in diesem Buch

In diesem Buch wird keine vollständige Prozesskostenrechnung durchgeführt. Um Ist- und Soll-Prozesse vergleichen zu können, ist die Bestimmung der Prozesskostensätze ausreichend. Ein Prozesskostensatz stellt nach Jórasz die durchschnittlichen Kosten einer einmaligen Durchführung eines Prozesses dar.<sup>40</sup> Sie können für Soll-/Ist-Vergleiche genutzt werden.<sup>41</sup>

Bei einer vollständigen Prozesskostenrechnung werden nach der Berechnung des Prozesskostensatzes die Kosten der leistungsmengenneutralen Prozesse auf die Prozesskostensätze verteilt. Dies geschieht nach dem Durchschnittsprinzip.<sup>42</sup> Die geplanten Prozessoptimierungen in diesem Buch beziehen sich ausschließlich auf Softwareentwicklungsprozesse und haben daher keine Auswirkungen auf die indirekten Bereiche. Daher sind die Kosten aus diesen Bereichen vor und nach der Optimierung gleich hoch und haben bei einem Vergleich der Ist- und Soll-Situation keine Bedeutung.

Die Ermittlung der Prozesskostensätze erfolgt nach Jórasz in drei Schritten<sup>43</sup>:

- Bestimmung der Prozesse
- Festlegung der Bezugsgrößen
- Bildung von Prozesskostensätzen

Jeder dieser Schritte wird in einem der nächsten Kapitel genauer erläutert.

### 2.9.2 Bestimmung der Prozesse

Für eine Prozesskostenrechnung ist eine intensive Tätigkeitsanalyse der betroffenen Kostenbereiche notwendig.<sup>44</sup> „Zunächst werden alle wesentlichen Tätigkeiten listenartig festgestellt. Dazu kann man auf Arbeitspläne und Stellenbeschreibungen zurückgreifen oder die notwendigen Informationen durch Interviews beschaffen.“<sup>45</sup>

Die ermittelten Teilprozesse werden anschließend kostenstellenübergreifend zu Hauptprozessen zusammengefasst. Die Hauptprozesse bilden dann die Grundlage für die Prozesskostenkalkulation.<sup>46</sup> Dieser Schritt ist für die Prozesskostenrechnung von zentraler Bedeutung, weil durch ihn die Grundlage geschaffen wird, die betrieblichen Aktivitäten als

---

<sup>40</sup> Vgl. Jórasz (2008), S. 285.

<sup>41</sup> Vgl. Schmidt (2008), S. 232.

<sup>42</sup> Vgl. Jórasz (2008), S. 285.

<sup>43</sup> Vgl. ebd., S. 282.

<sup>44</sup> Vgl. ebd., S. 283.

<sup>45</sup> Schmidt (2008), S. 224.

<sup>46</sup> Vgl. Jórasz (2008), S. 283.

## Hybride Softwareentwicklung

Das Beste aus klassischen und agilen Methoden in  
einem Modell vereint

Berg, B.; Knott, P.; Sandhaus, G.

2014, XVII, 126 S. 34 Abb., Hardcover

ISBN: 978-3-642-55063-8