

1 Einleitung

Motivation der Arbeit Die SV SparkassenVersicherung (SV) ist bestrebt ihre Versicherungsprozesse und den Kontakt mit den Kunden durch digitale Kanäle ständig zu verbessern. Die Digitalisierung ist ein wichtiger Bestandteil der SV-Strategie: So ist sie Mitbegründerin der „id-fabrik“¹ und Mitglied im „InsurLab Germany“.² Um den Kunden der SV immer besseren Service bieten zu können, möchte sie die neuen Technologien nutzen. Die Nachfrage nach ständiger Verfügbarkeit der Systemen stellt die IT-Dienstleisterin (SV Informatik GmbH (SVI)) der SV vor neue Herausforderungen, denn die Verteilungsprozesse von neuen Versionen der Anwendungen können derzeit nicht während des laufenden Betriebes durchgeführt werden. So müssen neue Funktionen warten, bis das Wochenende des „Release“ sie veröffentlicht, d. h. im schlechtesten Fall ist die neue Funktion drei Monate auf der Warteliste. Dieses Vorgehen passt nicht zu den digitalen Bestrebungen der SV. Als Lösungsansatz sind die Eigenschaften der „Cloud“-Technologie zu nennen. Jedoch muss aus IT-Sicht hier eine Anmerkung gemacht werden:

Solved all your problems. You're welcome.



Abbildung 1.1: Dilbert Comic zu KUBERNETES

Quelle: *Dilbert on Kubernetes* 2017

Redaktionelle Anmerkung: Abbildung nur als komprimiertes Format verfügbar (Qualitätseinbuße)

Die, in Abbildung 1.1, dargestellte Aussage ist absichtlich überspitzt, um das Verständnis der Lösungsmöglichkeiten der „Cloud“-Technologie zu schärfen. Der Hinweis der IT stellt klar, dass das oben beschriebene Problem nur mit einer ganzheitlichen Lösung zu erreichen ist. Es führt nicht zum gewünschten Ergebnis, Anwendungen, die nicht für die „Cloud“ gedacht sind, mit Zwang in diese zu bewegen. Diese Arbeit beschreibt den Prozess zur Verteilung von Container-Anwendungen.

¹Ein Start-up, das federführend Innovationen im Bereich der S-Finanzgruppe erzeugt.

²vgl. SV SparkassenVersicherung 2019, S. 30.

2 Methodologie: Beschreibung des Vorgehens

Forschungsfrage eins Die Anforderungsanalyse, die in dieser Forschungsfrage benutzt wurde, verwendet das Vorgehen von Hull, Jackson und Dick 2011 und Partsch 2010. Vor allem in Hull, Jackson und Dick 2011 wurde die methodische Vorgehensweise stark fokussiert, um eine hohe Qualität und Aussagekraft der Analyse zu gewährleisten. Deswegen nutzte diese Arbeit die dort illustrierte Methodik, da, erstens, es ein de-facto-Standard darstellt⁹ und, zweitens, das Vorgehen passend für die Problemstellung dieser Arbeit erschien. Die „stakeholder“ wurden anhand des vorhandenen „Deployment“-Prozesses abgeleitet, d. h. es war beim aktuellen Prozess Projektbeteiligte in Form von Rollen definiert. Diese wurden für jede Ausprägung des Prozess mit bestimmten „stakeholder“ gefüllt. Dadurch war der relevante Personenkreis aus den aktuellen Prozessen auf den zu implementierenden übertragen. Das „statement of needs“ war durch den Fachbereich „Entwicklung“ konkretisiert. Des Weiteren sollte ein Fragebogen die Wünsche der beteiligten Fachbereiche („stakeholder“) aufnehmen, um daraus die präzisen Anforderungen abzuleiten. Dabei ward absichtlich eine offene Fragestellung an den Entwicklungs-Fachbereich (namentlich: „SQUAD1“) gesendet, da es keine Beeinflussung durch die Fragestellung geben sollte. Ziel war es, die wirklichen Wünsche des Fachbereichs herauszufinden. Aus den Wünschen wurden nach dem Muster von vgl. Hull, Jackson und Dick 2011, S. 28 (siehe dazu auch Kapitel 3.1.1 auf Seite 9) spezifische funktionale und nicht-funktionale Anforderungen formuliert. Als Formulierungshilfe, um die Verständlichkeit und die Messbarkeit zu wahren, dienten die Regeln nach Rupp 2020.

Die Modellierung einer „OPENS SHIFT“-Labor-Umgebung diente dem Zweck der Anwendungsevaluierung, so wurde die Funktionsweise dieses System unter eingeschränkten Bedingungen getestet. Einschränkend wirkten die Limitierung auf eine „worker node“, die beschränkte Leistung der virtuellen Maschine, die Version von OPENS SHIFT und die Entkopplung von der internen AWL. Hier wurde die „open source“-Variante genutzt, da so gewährleistet war, dass alle Tests und Verprobungen ohne Einschränkungen der Produktivumgebung durchgeführt wurden. Durch die Labor-Umgebung konnte die allgemeine Funktionsweise des Clusters getestet werden. Ziel dieser Umgebung war es, Tests zu ermöglichen und das Zusammenspiel der einzelnen Komponenten zu überprüfen. Aus diesem Grund wurde die Methodik der Labor-Umgebung gewählt, da sie einige Vorteile für diese Arbeit bot: Es konnte die Funktionsweise der Anwendungen getestet werden, die Konfigurationsdateien für das Deployment wurden unter realisti-

⁹Die Monographie wurde 1200 mal laut GOOGLE SCHOLAR – Google LLC 2020a – zitiert.

Dies ist eine Anforderung an die Vollständigkeit der Konfigurationsdatei, die die benötigten Komponenten beschreibt. Wichtig dabei ist es, dass die Konfigurationsdatei nicht nur vollständig ist, sondern auch semantisch korrekt ist. Ein zentrales Akzeptanzkriterium ist, dass das System automatisiert funktioniert und bei Normalbetrieb keine menschliche Aufsicht benötigt.

A₁₀: Das System muss den gesetzlichen sowie sicherheitstechnischen Vorgaben entsprechen.

A₁₀ beschreibt eine nicht-funktionale Anforderungen, die zwingend erforderlich ist und deswegen an dieser Stelle erwähnt wird. Im Bereich der Finanzdienstleistungen, welche das Versicherungswesen inkludieren, sind strenge Sicherheitsvorschriften und gesetzliche Rahmenbedingung einzuhalten (vgl. dazu die Forschungsfrage drei im Kapitel 5 auf Seite 47). Diese nicht-funktionale Anforderung ist an alle Vorhaben/Projekte gerichtet und wird von dem Projektbeteiligten „Geschäftsführung“ eingebracht. Diese Anforderung muss zwingend umgesetzt werden.

3.3 Konzeption eines container-basierten, automatisierten „Deployments“

Dieses Teilkapitel beschreibt den Aufbau einer OPENSHIFT-Labor-Umgebung, um das Verhalten von OPENSHIFT zu testen. Des Weiteren wird eine Prozessmodellierung des automatisierten „Deployments“ anhand einer Beispielanwendung CAMUNDA erläutert. Dies ist gleichzeitig die Testapplikation, die in der SVI benutzt wird, um das automatisierte „Deployment“ von Container-Anwendungen zu untersuchen. Abschließend wird eine generische Konfigurationsdatei entwickelt, um Applikationen ins OPENSHIFT-Cluster zu verteilen.

3.3.1 Aufbau und „Deployment“ eines OpenShift-Labors

Nachfolgend wird der Aufbau einer OPENSHIFT-Labor-Umgebung mit „single node“-Architektur beschrieben, d.h. es ist nur eine „worker node“ im OPENSHIFT-Cluster vorhanden. Dieses wird auf einem UBUNTU-System in der Version 18.04 installiert. Der folgende Quelltext 3.1 beschreibt die Terminal-Eingaben:

```
1 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
   ↪ apt-key add -
2 sudo add-apt-repository "deb [arch=amd64]
   ↪ https://download.docker.com/linux/ubuntu $(lsb_release
   ↪ -cs) stable"
```

```
3 sudo apt update && sudo apt -y install docker-ce
4 docker version
5 sudo usermod -aG docker $USER
6 wget $LINK
7 tar xvf openshift-origin-client-tools*.tar.gz
8 cd openshift-origin-client*/
9 sudo mv oc kubectl /usr/local/bin/
10 oc version
11 cat << EOF | sudo tee /etc/docker/daemon.json
12 {
13 "insecure-registries" : [ "172.30.0.0/16" ]
14 }
15 EOF
16 sudo systemctl restart docker
17 oc cluster up
```

Quelltext 3.1: Installation des OPENSIFT-Clusters

Das Konzept dieses „Shell“-Skripts 3.1 auf der vorherigen Seite ist es, zuerst die notwendigen Abhängigkeiten und danach die OPENSIFT-Anwendungen zu installieren. So wird in den Zeilen eins bis drei die notwendigen Voraussetzungen geschaffen, um die DOCKER-Umgebung zu installieren. Zuerst wird der GPG⁵⁶-Schlüssel zum lokalen Schlüsselbund hinzugefügt, danach kann die Anwendung DOCKER mit dem „Advanced Packaging Tool“ (APT)-Programm heruntergeladen und installiert werden. Danach wird getestet, ob die Installation von DOCKER erfolgreich war und der eigene Benutzer wird der Gruppe „docker“ hinzugefügt. Dies wird benötigt um dem eigenen Benutzer die Steuerung der Software DOCKER zu erlauben. Die Zeilen sechs bis neun installieren KUBERNETES und OPENSIFT (in der „Community“-Variante). Sehr wichtig sind die Anpassungen, die in den Zeilen elf bis fünfzehn vorgenommen werden: Sie fügen einen Eintrag "insecure-registries": ["172.30.0.0/16"] in die Datei /etc/docker/daemon.json ein. Dies erlaubt es DOCKER auf unsichere Verzeichnisse zuzugreifen; dadurch kann OPENSIFT lokal Container-„Images“ im Cache speichern, um diese schneller wiederzuverwenden. Schließlich wird ein Cluster in „default“-Konfiguration zur Kontrolle der Funktionsfähigkeit mit `oc cluster up` erstellt. Dies ist unter <http://127.0.0.1:8443/console> als Web-Konsole erreichbar.

Um das Verhalten des OPENSIFT-Clusters während eines „Deployments“ zu testen, wird ein Test-Projekt erstellt. Grundsätzlich sind die meisten Kommandos der CLI nach dem Muster `oc <action> <object type> <obj name or id>`⁵⁷ aufgebaut.

⁵⁶ „GnuPG is a complete and free implementation of the OpenPGP standard as defined by RFC4880 (also known as PGP). GnuPG allows you to encrypt and sign your data and communications; it features a versatile key management system, along with access modules for all kinds of public key directories.“ Quelle: The People of the GnuPG Project 2020

⁵⁷ vgl. Red Hat, Inc. 2020a.

Ein Projekt ist in OPENSHIFT ein privater Bereich, indem Applikationen laufen können und nur die Erstellerin administrativen Zugriff besitzt. Dieser kann nach außen veröffentlicht werden, wenn dies erforderlich ist. Nachdem das Projekt mit `oc new-project <projectName> --display-name '<displayName>'`⁵⁸ erstellt wurde, wird dies automatisch als „default“-Projekt ausgewählt, d. h. darin werden nun alle folgenden Schritte ausgeführt. Des Weiteren ist die Überlegung zu treffen, ob noch weitere Benutzer für das angelegte Projekt eine Berechtigung erhalten (`oc adm policy add-role-to-user <admin/edit/view> <collabUser>`).

Als Test-Applikation soll eine Webseite auf Basis von DJANGO⁵⁹ in Cluster verteilt werden. Es wird ein vorgefertigtes „Image“ für diesen Test genutzt. Der Quellcode 3.2 zeigt die benötigten Kommandos, um eine neue Applikation zu erstellen. Des Weiteren illustriert dieser, wie die Anwendung nach außen freigegeben wird, d. h. sie ist außerhalb des Projektes erreichbar. Danach wird in Zeile 7 der Quellcode-Abbildung 3.2 die Verbindung zur Webseite mittels der Ermittlung des HTTP-Status-Codes überprüft.⁶⁰

```

1 oc new-app openshiftkatacoda/blog-django-py --name blog
2 oc describe svc/blog # print config
3 oc expose svc/blog
4 oc status | grep 'to pod port' # see all exposed services on
   ↪ this project
5
6 # test_connection is listed at A.1 auf Seite XXVIII
7 test_connection blog-lab.127.0.0.1.nip.io
```

Quelltext 3.2: Test-„Deployment“ ins OPENSHIFT-Cluster

Nach Abschluss des Tests wird die Applikation mit `oc delete all --selector app=blog` wieder gelöscht.

3.3.2 Modellierung einer „Deployment“-Konfigurationsdatei

Um eine Verteilung der Container-Anwendung auf dem OPENSHIFT-Cluster durchzuführen, benötigt das System eine Konfigurationsdatei, die als API-Objekt an OPENSHIFT übergeben wird. Dabei definiert diese den gewünschten Zustand einer bestimmten Komponente der Anwendung als „pod“⁶¹-Vorlage.⁶² Diese Datei stellt, als alleinige

⁵⁸ „<Platzhalter>“, die mit Werten vor der Ausführung ersetzt werden müssen

⁵⁹ „Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.“ Quelle: Django Software Foundation 2020

⁶⁰ Der Quelltext dieser Funktion ist im Anhang A.1 auf Seite XXVIII einzusehen

⁶¹ siehe Tabelle 3.1 auf Seite 19

⁶² vgl. Red Hat, Inc. 2019, Application → Deployments.

Eine Anforderung der Abteilung IE2 ist es, die Konfigurationsdatei über eine Generierung zu erstellen, um mögliche Syntaxfehler ausschließen zu können. Des Weiteren soll so eine Standardisierung der Konfigurationsdatei unternehmensweit erzielt werden. Das Vorgehen dazu soll folgende Stufen enthalten:

1. Eine einheitliche Übergabedatei soll von den Entwicklungsteams in einer Versionsverwaltung abgelegt werden. Die Datei ist nach Vorgaben der Abteilung IE2 aufgebaut.
2. Danach soll aus dieser Datei die Konfigurationsdatei erzeugt werden.
3. Diese wird durch OPENSIFT validiert, bevor sie ins Cluster geladen wird.
4. Das Ergebnis ist die vollständige Konfigurationsdatei, die die Komponenten beschreibt, die verteilt werden sollen.

Für die Generierung ist es wichtig ein Grundgerüst der Konfigurationsdatei zu haben. Dieses ist im Anhang A.3 auf Seite XXXI mit beispielhaften Werten zu sehen. Zweck des Grundaufbaus ist es, bei der Generierung die fehlenden Informationen möglichst leicht zu integrieren. Dabei hilft es, wenn eine Grundstruktur des Ergebnisses vorhanden ist. Der Aufbau der zulässigen Wortfolgen in einer Zeile der Übergabedatei ist nachfolgend in der Quellcode-Darstellung 3.7 als Backus-Naur-Form (BNF)⁷³ beschrieben.

```

1      <Zeile> ::= <Schluessel> <Zuweisung> <Wert> <EOL> | <
           ↳ Gruppe> <EOL> | λ <EOL>
2      <Schluessel> ::= <Grossbuchstaben> | <Grossbuchstaben> <
           ↳ Schluessel>
3      <Zuweisung> ::= "="
4      <Wert> ::= <Zeichenkette>
5      <Gruppe> ::= "[" <Zeichenfolge> "]"
6      <Zeichenfolge> ::= <Grossbuchstaben> | <Grossbuchstaben> <
           ↳ Zeichenfolge>
7      <Zeichenkette> ::= λ | <Zeichen> | <Zeichen> <
           ↳ Zeichenkette>
8      <Grossbuchstaben> ::= "A" | "B" | "C" | "D" | "E" | "F"
           ↳ | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N"
           ↳ | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V"
           ↳ | "W" | "X" | "Y" | "Z" |
9      <Zeichen> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" |
           ↳ "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" |
           ↳ "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" |

```

⁷³ „Using BNF it is possible to specify which sequences of symbols constitute a syntactically valid program in a given language. (The question of semantics—i.e, what such valid strings of symbols mean—must be specified separately.)“ (McCracken und Reilly 2003)

```

↪ "x" | "y" | "z" | "_" | "/" | "-" | ":" | "." |
↪ "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |
↪ "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" |
↪ "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
↪ "y" | "z"

```

Quelltext 3.7: BNF der Übergabedatei

Ein Beispiel einer validen Übergabedatei ist im Anhang A.4 zur Veranschaulichung des Konzeptes abgebildet. Nachfolgend beschreibt der Algorithmus 1 die Logik zur Generierung einer Konfigurationsdatei. Dieser ist absichtlich in Pseudocode verfasst, um den Fokus auf die Logik zu setzen und nicht auf Besonderheiten einer Programmiersprache. Die Idee dieses ist es, die Übergabedatei einzulesen, auszuwerten und auf Basis dieser eine valide Konfigurationsdatei zu erstellen.

Algorithmus 1 Generierung der Konfigurationsdatei aus einer K-V-Datei

```

procedure GENERATEDEPLOYMENTCONFIG(pathToKVFile)
  path ← pathToKVFile
  map[ ][ ]      ▷ Map with a key and a value, e.g. a array of lines as value
  lines[ ] ← READFILEBYLINES(path)
  currentKey = null
  yamlTemplate ← READFILE(pathToYamlTemplate)

  for all lines[ ] as line do
    if line like REGEX("\{1\}([A-Z]+)\{1\}") then      ▷ New group element?
      keyword ← TRIM(line, "[", "]")
      currentKey ← keyword
      map.ADDKEY(currentKey)
    else                                              ▷ line contains a k-v-pair
      map[currentKey].ADD(line)
    end if
  end for      ▷ All lines of file are stored in k-v-pairs in var map now

  allKeys[ ] ← map.GETALLKEYS
  for i ← 0, i ≤ LENGTH(allKeys[ ]), i ++ do
    for all lines of map[allKeys[i]] as line do
      ADDTOYAML(yamlFile, lineToBeAdded, key)
    end for
  end for
end procedure

```

Ziel des Algorithmus 1 ist es, aus einer definierten Übergabedatei (siehe dazu Quellcode 3.7 auf der vorherigen Seite) die Konfigurationsdateivorlage („yamlTemplate“)

Ein „Business Case“ kann intern oder extern erstellt werden. Es sind noch weitere Kombinationsmöglichkeiten denkbar, die in der Praxis jedoch kaum eine Rolle spielen.⁹⁹ Es gibt für beide Möglichkeiten, intern oder extern erstellt, Vor- und Nachteile, die im Anhang B.2 auf Seite XXXV abgebildet sind. Die beteiligten Einheiten des Unternehmens entstammen der Informatik, einer „Business Unit“, der Finanzabteilung (meist „Controlling“) und der Personalabteilung. Entscheidungen, die das Projekt und somit das Geschäftsszenario betreffen, werden durch die höheren Führungsebenen in Verbindung mit den projektanforderten Bereich getroffen. Die Erstellung teilt sich in drei Ebenen auf: Initialisierung („Business Case Definition“), Entwicklung („Business Case Development“) und Prüfung („Business Case Quality Check“).¹⁰⁰ Während der Initialisierungsphase werden die Teams definiert, eine Eingrenzung der beteiligten Abteilungen vereinbart, die Faktoren/Parameter für die Wirtschaftlichkeitsrechnung festgelegt und die Kalkulationsmethoden für die Ermittlung der Kennzahlen gewählt. In der Entwicklungsphase werden folgende Arbeitsschritte durchgeführt: Projektplanung/Systemkonzeption, Erhebung und Analyse der Kosten/des Nutzens, Aufbau des Wirtschaftlichkeitsmodells; Auswertung der Ergebnisse, die eine Sensitivitäts- (Versuch die optimale Lösung weiter zu verbessern), eine Risiko- und Strategieanalyse enthält; und die Zusammenfassung für die Führungsebene. Im Anhang B.2 auf Seite XXXV ist eine Abbildung mit der chronologischen Anordnung der Arbeitsschritte zu sehen, die nochmals auf die Abhängigkeit der Schritte hinweist. Die letzte Phase beschäftigt sich mit der Qualitätssicherung der gewonnenen Erkenntnisse, dabei werden eine Validierung der Annahmen, die Prüfung der eingegebenen Daten und eine Abstimmung mit anderen Projekten durchgeführt.

4.2 Geschäftsprozess: „Release“ von neuen Anwendungsversionen

Der Geschäftsprozess „Release“ beinhaltet den „Deployment“-Prozess, deswegen wird in der Geschäftsprozessanalyse der „Release“-Prozess als Ganzes betrachtet, um den wirtschaftlichen Effekt der Veränderung des „Deployments“ zu untersuchen. Es wird zuerst der aktuelle „Release“-Prozess ohne den Container-„Deployment“-Prozess betrachtet.

Der aktuelle „Release“-Prozess folgt in den Entwicklungsarbeiten dem Vorgehensmodell „Wasserfall“. Dies ist ein iteratives Vorgehensmodell zur Anwendungsentwicklung, dabei teilt sich das streng Modell in Konzeption und Umsetzung auf.¹⁰¹ Die SVI hat ein Meilenstein-Konzept für das „Release“ entwickelt, der das genaue Vorgehen dieses

⁹⁹vgl. Brugger 2009, S. 33.

¹⁰⁰vgl. Brugger 2009, S. 41-42.

¹⁰¹vgl. Freund und Rücker 2017, S. 24.

Geschäftsvorfalls beschreibt. Dieser definiert Aufgaben und Tätigkeiten, die zu einer Folge kombiniert werden.

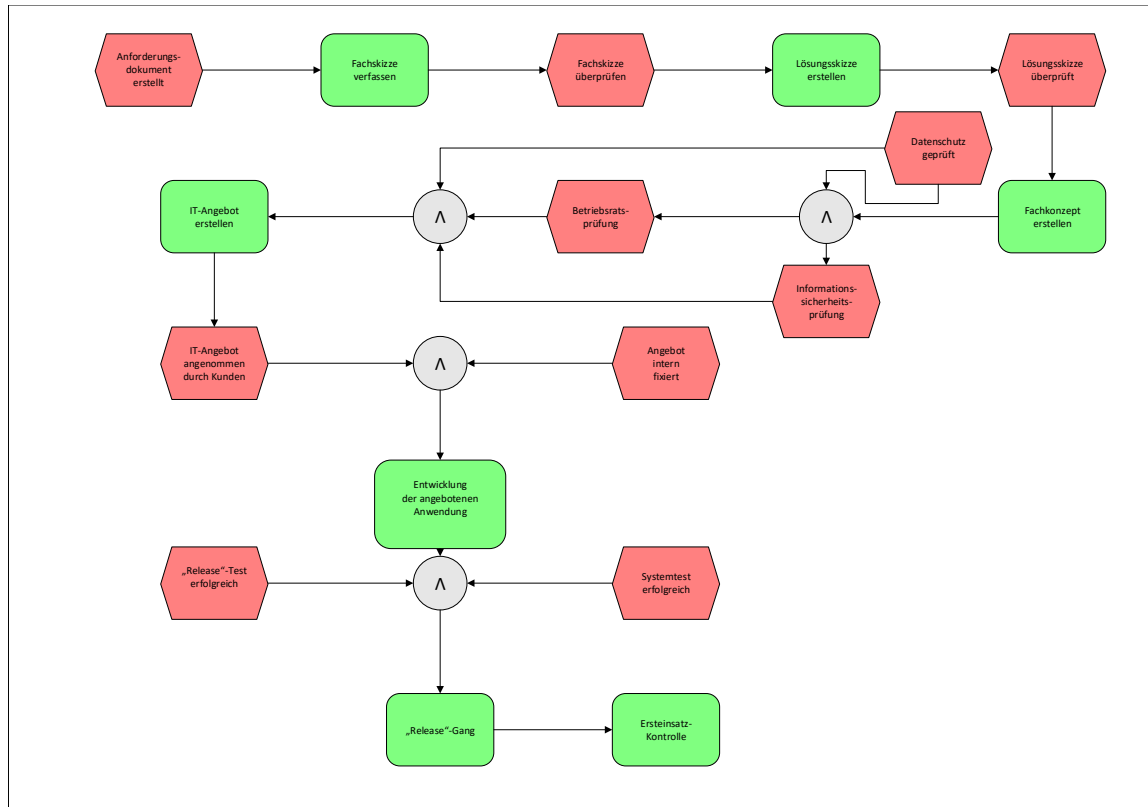


Abbildung 4.1: EPK zum Geschäftsvorfall „Release“

Quelle: in Anlehnung an unternehmensinterne Dokumente

unternehmensintern

Die Abbildung 4.1 zeigt die abgewandelte EPK des Geschäftsprozesses: Die rot gefärbten Formen sind Ereignisse, die grünen Funktionen/Arbeitspakete und das grau gefärbte Symbol beschreibt eine logische „AND“-Verknüpfung. Dieser Prozess hat verschiedene Prüfstellen, die die Qualität der erstellten Dokumente überprüfen. Erst ab der Funktion „Entwicklung der angebotenen Anwendung“ werden Anwendungen programmiert. Davor sind alle Aktivitäten zur Erstellung, Prüfung und Dokumentation der zu entwickelnden Anwendung durchzuführen. Diese sind mit erheblichen Aufwand verbunden: so muss die Kundin zusammen mit den Kundenmanagerinnen ein erstes Konzept entwickeln und beschreiben was das „statement of needs“ ist. Dieses wird in den nächsten Schritten zu einem Anforderungsdokument überarbeitet, dabei werden auch mögliche Lösungsvorschläge berücksichtigt. Schließlich ist das Fachkonzept und die Lösungsskizze erstellt. Die Kundenmanagerin kreiert ein IT-Angebot, dass der Kundin vorgelegt wird. Diese beiden Personenkreise verhandeln dann über die Kosten und Nutzen des Vorhabens. Sobald die Kundin zustimmt und alle anderen Zustimmung-

gen (Betriebsrat, IT-Sicherheit und Datenschutz) eingeholt sind, startet die Entwicklung der beschriebenen Funktion. Nun folgend Tests in verschiedenen Umgebungen und dann die Freigabe für die Produktivsetzung der neuen Funktion/Anwendung. Danach prüft die Kundin, ob alle vorgesehenen Funktionen nach ihren Vorstellungen umgesetzt wurden. Dieses Vorgehen spiegelt einen linearen Vorgang wieder, der sich schwer an sich ständig veränderten Anforderungen anpassen kann. Ist das Fachkonzept freigegeben, kann an diesem nichts mehr geändert werden. Jedoch verändern sich die Anforderungen immer schneller und die Software hat einen immer kürzer andauernden Lebenszyklus innerhalb der SV. Die Digitalisierung wird in der SV-Strategie als zentraler Bestandteil beschrieben.¹⁰² Somit wird die Forderung nach einer Anpassung des „Release“-Prozesses immer stärker.

Durch die Veränderung des „Deployment“-Prozess ändert sich der gesamte „Release“-Prozess ebenfalls: So müssen die Planung und die Nicht-Anpassbarkeit des Fachkonzeptes überdacht werden. Außerdem ist der aktuelle Prozess, wie in Abbildung 4.1 auf der vorherigen Seite dargestellt, so mit den starren Regeln nicht mehr möglich. In einer Übergangsphase, bis ein neuer Geschäftsprozess entwickelt wurde, kann dieser aktuelle in einer abgewandelten Form benutzt werden. Diese Form müsste eine Änderung der Anforderungen und somit eine Änderung des Fachkonzeptes während der Entwicklung neuer Anwendungen zulassen. Dies kann zu inkonsistenten Daten in der Dokumentation führen und deswegen mit Vorsicht zu betrachten. Des Weiteren könnte eine Art Kategorisierung für die Anforderungen gemacht werden, d. h. zwingend umzusetzende Anforderung der Anwendungen wären ersichtlich. So könnten die Fachkonzepte in mindestens zu erfüllende und optimale Anteile unterteilt werden.

Die Container-Anwendungen unterstützen die Schnelligkeit der Entwicklung, da sie direkt mit einem entsprechenden „Deployment“-Prozess produktiv gehen können. Ist die Teststruktur automatisiert und die Dokumentationspflicht erfüllt, kann der Aufwand und damit die Kosten für die Entwicklung einer Funktion einer Anwendung beziehungsweise die Entwicklung einer kompletten Anwendung reduziert werden. Dies folgt aus der Tatsache, dass weniger Arbeitsstunden in die Betreuung, Entwicklung, Kontrolle und Dokumentation investiert werden müssen. Bleibt der Preis der Anwendung gleich, erhöht sich trotzdem der erzielte Gewinn, da die Kosten sich reduzieren. Natürlich sind initial die Kosten höher, da die Entwicklung eines neuen Prozess sehr viel Aufwand darstellt.

Momentan ist die SVI in einer Übergangsphase, in der sie den Geschäftsprozess 4.1 auf der vorherigen Seite weiterhin nutzt. Jedoch wurden hier die oben genannten Veränderungen implementiert. Die illustrierte Weiterentwicklung des „Deployment“-Prozess (siehe Kapitel 3 auf Seite 9) ist eine Möglichkeit die Kosten durch Reduzierung des Betreuungsaufwandes in der Abteilung IE2 zu senken.

¹⁰²vgl. SV SparkassenVersicherung 2019.

Literaturverzeichnis

Atomic Requirement Download (19. Aug. 2019). URL: <https://www.volere.org/atomic-requirement-download/>.

AXELOS Limited und Stationery Office (Great Britain) (2019). *ITIL® Foundation, ITIL 4 edition*. v4. OCLC: 1122856407. Norwich: TSO (The Stationery Office). ISBN: 978-0-11-331607-6. URL: https://www.axelos.com/getmedia/5896d51f-ab6c-4843-992b-4f045eab0875/ITIL-4-Foundation-glossary_v0_22.aspx (besucht am 10.03.2020).

Bernstein, David (Sep. 2014). „Containers and Cloud: From LXC to Docker to Kubernetes“. In: *IEEE Cloud Computing* 1.3, S. 81–84. ISSN: 2372-2568. DOI: 10.1109/MCC.2014.51.

Brewer, Eric (Feb. 2012). „CAP twelve years later: How the "rules" have changed“. In: *Computer* 45.2, S. 23–29. ISSN: 0018-9162, 1558-0814. DOI: 10.1109/MC.2012.37. URL: <https://ieeexplore.ieee.org/document/6133253/> (besucht am 08.04.2020).

Broadcom Inc. (2020). *Automic Automation*. Library Catalog: www.broadcom.com. URL: <https://www.broadcom.com/products/software/automation/digital-business-automation/automic-automation> (besucht am 16.04.2020).

Brugger, Ralph (2009). *Der IT Business Case*. Xpert.press. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-93857-6 978-3-540-93858-3. DOI: 10.1007/978-3-540-93858-3. URL: <http://link.springer.com/10.1007/978-3-540-93858-3> (besucht am 19.03.2020).

Bundesamt für Sicherheit in der Informationstechnik (BSI) (20. Okt. 2017). *Leitfaden zur Basis-Absicherung nach IT-Grundschutz: In 3 Schritten zur Informationssicherheit*. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/Leitfaden_zur_Basis-Absicherung.html (besucht am 27.03.2020).

Bundesamt für Sicherheit in der Informationstechnik (BSI) (20. Apr. 2018). *Zuordnungstabelle ISO zum modernisierten IT-Grundschutz*. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/Zuordnung_ISO_und_modernisierter_IT_Grundschutz.pdf;jsessionid=5ADB145EF2581C0DD41F6CBA3A702_cid360?__blob=publicationFile&v=9 (besucht am 27.03.2020).

- Bundesamt für Sicherheit in der Informationstechnik (BSI) (2020). *IT-Grundschutz-Kompendium*. 2020. Aufl. OCLC: 1027470677. Bundesanzeiger Verlag GmbH. 816 S. ISBN: 978-3-8462-0906-6. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2020.pdf?__blob=publicationFile&v=6 (besucht am 09.03.2020).
- Caban, William (2019). *Architecting and Operating OpenShift Clusters OpenShift for Infrastructure and Operations Teams*. OCLC: 1138978697. Berkeley, CA: Apress : Imprint: Apress. ISBN: 978-1-4842-4985-7 978-1-4842-4984-0 978-1-4842-4986-4. URL: <https://link.springer.com/10.1007/978-1-4842-4985-7> (besucht am 24.02.2020).
- Camunda Services GmbH (2020). *Workflow und Decision Automation | Camunda BPM*. Camunda. URL: <https://camunda.com/de/> (besucht am 14.04.2020).
- Canonical Ltd (2020). *Linux Containers*. Linux Container. URL: <https://linuxcontainers.org/> (besucht am 17.03.2020).
- Combe, Theo, Antony Martin und Roberto Di Pietro (Sep. 2016). „To Docker or Not to Docker: A Security Perspective“. In: *IEEE Cloud Computing* 3.5, S. 54–62. ISSN: 2372-2568. DOI: 10.1109/MCC.2016.100.
- Dearle, Alan (Mai 2007). „Software Deployment, Past, Present and Future“. In: *Future of Software Engineering (FOSE '07)*. Future of Software Engineering (FOSE '07). Minneapolis, MN: IEEE, S. 269–284. ISBN: 978-0-7695-2829-8. DOI: 10.1109/FOSE.2007.20. URL: <https://ieeexplore.ieee.org/document/4221626/> (besucht am 18.03.2020).
- Dechange, André (2020). *Projektmanagement – Schnell erfasst*. Wirtschaft – Schnell erfasst. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-57666-3 978-3-662-57667-0. DOI: 10.1007/978-3-662-57667-0. URL: <http://link.springer.com/10.1007/978-3-662-57667-0> (besucht am 01.04.2020).
- Deutsches Institut für Normung, e.V. (10. Juni 2017). *Informationstechnik - Sicherheitsverfahren - Leitfaden für Informationssicherheitsmaßnahmen (ISO/IEC 27002:2013 einschließlich Cor 1:2014 und Cor 2:2015); Deutsche Fassung EN ISO/IEC 27002:2017*. URL: <https://perinorm-fr.reidi-bw.de/perinorm/fulltext.ashx?fulltextid=ae4372165f8d4fa1bb8bff06b0923e46&userid=96f4b659-56a0-45ff-a5be-da1774bd04e8> (besucht am 27.03.2020).
- Dilbert on Kubernetes* (11. Aug. 2017). URL: https://miro.medium.com/max/1024/1*RODEnf_7sjswuBHouioQFg.jpeg.

- DIN Deutsches Institut für Normung, e. V. (5. Juli 2019). *Informationstechnik – Sicherheitsverfahren – Informationssicherheits-Managementsysteme – Überblick und Terminologie (ISO/IEC 27000:2018); Deutsche und Englische Fassung prEN ISO/IEC 27000:2019*. URL: <https://perinorm-fr.redi-bw.de/perinorm/fulltext.ashx?fulltextid=3fb39d48521c4797907ba51ad7c443da&userid=96f4b659-56a0-45ff-a5be-da1774bd04e8> (besucht am 27.03.2020).
- DIN Deutsches Institut für Normung, e. V. (26. Feb. 2020a). *Informationstechnik - Cloud Computing - Übersicht und Vokabular (ISO/IEC 17788:2014)*. URL: <https://perinorm-fr.redi-bw.de/perinorm/fulltext.ashx?fulltextid=94c35b8fadfc4c51853726be&userid=96f4b659-56a0-45ff-a5be-da1774bd04e8>.
- DIN Deutsches Institut für Normung, e. V. (26. Feb. 2020b). *Informationstechnik - Sicherheitsverfahren - Informationssicherheitsmanagementsysteme - Anforderungen (ISO/IEC 27001:2013 einschließlich Cor 1:2014 und Cor 2:2015); Deutsche Fassung EN ISO/IEC 27001:2017*. URL: <https://perinorm-fr.redi-bw.de/perinorm/fulltext.ashx?fulltextid=b13c1f6be2f04f0298a6f7c96b1bbad1&userid=96f4b659-56a0-45ff-a5be-da1774bd04e8>.
- DIN Deutsches Institut für Normung, e. V. (26. Feb. 2020c). *Qualitätsmanagementsysteme - Grundlagen und Begriffe (ISO 9000:2015); Deutsche und Englische Fassung EN ISO 9000:2015*. URL: <https://perinorm-fr.redi-bw.de/perinorm/fulltext.ashx?fulltextid=230150cb1880449295f493e21a445a20&userid=96f4b659-56a0-45ff-a5be-da1774bd04e8>.
- Django Software Foundation (2020). *The Web framework for perfectionists with deadlines / Django*. URL: <https://www.djangoproject.com/> (besucht am 03.04.2020).
- Foster, Ian und Carl Kesselman, Hrsg. (1999). *The grid: blueprint for a new computing infrastructure*. San Francisco: Morgan Kaufmann Publishers. 677 S. ISBN: 978-1-55860-475-9.
- Freund, Jakob und Bernd Rücker (2017). *Praxishandbuch BPMN: mit Einführung in CMMN und DMN*. 5., aktualisierte Auflage. Type: Text (nur für elektronische Ressourcen). München: Hanser. ISBN: 3446450548 (Druck-Ausgabe). URL: <http://dx.doi.org/10.3139/9783446450783>.
- Gadatsch, Andreas (2010). *Grundkurs Geschäftsprozess-Management*. Wiesbaden: Vieweg+Teubner. ISBN: 978-3-8348-0762-5 978-3-8348-9346-8. DOI: 10.1007/978-3-8348-9346-8. URL: <http://link.springer.com/10.1007/978-3-8348-9346-8> (besucht am 20.03.2020).
- Gartner (2019). *Cloud Computing - Umsatz bis 2022*. Statista. Library Catalog: de.statista.com. URL: <https://de.statista.com/statistik/daten/studie/195760/umfrage/umsatz-mit-cloud-computing-weltweit/> (besucht am 13.03.2020).

- Google Ireland Limited (2020). *Container und ihre Vorteile*. Google Cloud. Library Catalog: cloud.google.com. URL: <https://cloud.google.com/containers?hl=de> (besucht am 17.03.2020).
- Google LLC (6. Apr. 2020a). *Google Scholar – Suche nach Hull*. URL: https://scholar.google.de/scholar?hl=de&as_sdt=0%2C5&q=hull+elizabeth+requirements+engineering&btnG=&oq=hull+elizabeth+requirements+engi.
- Google LLC (2020b). *Production-Grade Container Orchestration with K8s*. Library Catalog: kubernetes.io. URL: <https://kubernetes.io/> (besucht am 18.03.2020).
- Herman, B und J Siegelauß (2009). „Is this really worth the effort? The need for a business case“. In: *PMI Global Congress, Orlando, FL, October*. PMI Global Congress. Orlando, FL, USA: PMI. URL: <https://www.pmi.org/learning/library/need-business-case-6730> (besucht am 19.03.2020).
- Hill, Richard, Hrsg. (2013). *Guide to cloud computing: principles and practice*. Computer communications and networks. OCLC: ocn807043821. London ; New York: Springer. 278 S. ISBN: 978-1-4471-4602-5 978-1-4471-4603-2.
- Hull, Elizabeth, Ken Jackson und Jeremy Dick (2011). *Requirements engineering*. 3rd ed. London ; New York: Springer. 207 S. ISBN: 978-1-84996-404-3 978-1-84996-405-0.
- ICT/1 (26. Feb. 2020). *Information technology. Object Management Group Business Process Model and Notation*. URL: <https://www.omg.org/spec/BPMN/2.0/PDF>.
- IEEE (2005). „IEEE Standard for Application and Management of the Systems Engineering Process“. In: *IEEE Std 1220-2005 (Revision of IEEE Std 1220-1998)*, S. 1–96. DOI: 10.1109/IEEESTD.2005.96469.
- „Cloud Computing: Deployment Models, Delivery Models, Risks and Research Challenges“ (2011). In: 2011 International Conference on Computer and Management (CAMAN). Hrsg. von Institute of Electrical and Electronics Engineers. OCLC: 838686677. Wuhan, China: IEEE, S. 1–4. ISBN: 978-1-4244-9283-1 978-1-4244-9282-4. DOI: 10.1109/CAMAN.2011.5778816. URL: <https://ieeexplore.ieee.org/abstract/document/5778816>.
- International Organization for Standardization (2011). *ISO/IEC 25010:2011*. ISO. Library Catalog: www.iso.org. URL: <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/57/35733.html> (besucht am 12.04.2020).
- ITCandor (2019). *Cloud Computing - Marktanteile der führenden Unternehmen 2019*. Statista. Library Catalog: de.statista.com. URL: <https://de.statista.com/statistik/daten/studie/150979/umfrage/marktanteile-der-fuehrenden-unternehmen-im-bereich-cloud-computing/> (besucht am 13.03.2020).

- Kersten, Heinrich et al. (2020). *IT-Sicherheitsmanagement nach der neuen ISO 27001: ISMS, Risiken, Kennziffern, Controls*. Edition <kes>. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-27691-1 978-3-658-27692-8. DOI: 10.1007/978-3-658-27692-8. URL: <http://link.springer.com/10.1007/978-3-658-27692-8> (besucht am 26.03.2020).
- Kharb, Dr Latika (2016). „Automated Deployment of Software Containers Using Dockers“. In: 4.10, S. 3.
- Kumar, Vikas und R. Vidhyalakshmi (2018). *Reliability aspect of cloud computing environment*. Singapore: Springer. 170 S. ISBN: 9789811330230 9789811330223. URL: <https://doi.org/10.1007/978-981-13-3023-0>.
- Lukša, Marko (2018). *Kubernetes in Action: Anwendungen in Kubernetes-Clustern bereitstellen und verwalten*. OCLC: 1017485179. München: Hanser. 642 S. ISBN: 978-3-446-45510-8 978-3-446-45602-0.
- McCarthy, John (1983). *REMINISCENCES ON THE HISTORY OF TIME SHARING*. REMINISCENCES ON THE HISTORY OF TIME SHARING. URL: <http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html> (besucht am 13.03.2020).
- McCracken, Daniel D. und Edwin D. Reilly (2003). „Backus-Naur Form (BNF)“. In: *Encyclopedia of Computer Science*. GBR: John Wiley und Sons Ltd., S. 129–131. ISBN: 0-470-86412-5.
- Mell, Peter und Timothy Grance (28. Sep. 2011). *The NIST Definition of Cloud Computing*. Special Publication (NIST SP) 800-145. Gaithersburg, MD 20899-8930: National Institute of Standards und Technology. 7 S. URL: <https://doi.org/10.6028/NIST.SP.800-145> (besucht am 13.03.2020).
- Object Management Group (OMG) (1. Dez. 2011). *Business Process Model and Notation (BPMN)*. URL: <https://www.omg.org/spec/BPMN/2.0/PDF> (besucht am 23.03.2020).
- Object Management Group (OMG) (1. Dez. 2017). *Unified Modeling Language (UML)*. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (besucht am 06.04.2020).
- Opensource.org (2020). *The Open Source Definition*. The Open Source Definition. URL: <https://opensource.org/osd> (besucht am 10.03.2020).
- Pahl, Claus (Mai 2015). „Containerization and the PaaS Cloud“. In: *IEEE Cloud Comput.* 2.3, S. 24–31. ISSN: 2325-6095. DOI: 10.1109/MCC.2015.51. URL: <http://ieeexplore.ieee.org/document/7158965/> (besucht am 13.03.2020).
- Parkhill, Douglas Freeman (1966). *The Challenge of the computer utility: D.F. Parkhill*. OCLC: 460679364. Reading: Mass., London : Addison-Wesley Publishing C°.

- Partsch, Helmuth (2010). *Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme*. 2., überarb. und erw. Aufl. eXamen.press. OCLC: 845656932. Berlin: Springer. 394 S. ISBN: 978-3-642-05358-0 978-3-642-05357-3. URL: <http://dx.doi.org/10.1007/978-3-642-05358-0>.
- Pawar, Kulwant S., Unny Menon und Johann C.K.H. Riedel (1. Jan. 1994). „Time to Market“. In: *Integrated Manufacturing Systems* 5.1. Publisher: MCB UP Ltd, S. 14–22. ISSN: 0957-6061. DOI: 10.1108/09576069410815765. URL: <https://doi.org/10.1108/09576069410815765> (besucht am 17.03.2020).
- Red Hat, Inc. (1. Mai 2019). *OKD Latest Documentation*. OKD Latest Documentation. URL: <https://docs.okd.io/latest/welcome/index.html> (besucht am 07.04.2020).
- Red Hat, Inc. (2020a). *CLI Operations / CLI Reference / OpenShift Enterprise 3.0*. URL: https://docs.openshift.com/enterprise/3.0/cli_reference/basic_cli_operations.html (besucht am 03.04.2020).
- Red Hat, Inc. (2020b). *OpenShift Container Platform by Red Hat, Built on Kubernetes*. OpenShift Container Platform by Red Hat, Built on Kubernetes. URL: <https://www.openshift.com/> (besucht am 11.03.2020).
- Red Hat, Inc. (2020c). *Red Hat – Wir entwickeln Open Source-Technologien für Unternehmen*. Red Hat – Wir entwickeln Open Source-Technologien für Unternehmen. URL: <https://www.redhat.com/de> (besucht am 11.03.2020).
- Reinheimer, Stefan und Springer Fachmedien Wiesbaden GmbH, Hrsg. (2018). *Cloud Computing: die Infrastruktur der Digitalisierung*. Edition HMD. OCLC: 1038769740. Wiesbaden: Springer Vieweg. 216 S. ISBN: 978-3-658-20966-7 978-3-658-20967-4. URL: <https://doi.org/10.1007/978-3-658-20967-4>.
- Rimal, Bhaskar Prasad et al. (März 2011). „Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach“. In: *J Grid Computing* 9.1, S. 3–26. ISSN: 1570-7873, 1572-9184. DOI: 10.1007/s10723-010-9171-y. URL: <http://link.springer.com/10.1007/s10723-010-9171-y> (besucht am 13.03.2020).
- Rupp, C. (3. Apr. 2020). *Formulierungsregel für die funktionalen Anforderungen*. URL: https://www.pst.ifi.lmu.de/Lehre/fruhere-semester/sose-2009/seprakt/Formulierungsregel_DE_Rupp_Schablone.pdf (besucht am 03.04.2020).
- Scheer, A.-W., M. Nüttgens und V. Zimmermann (1. Mai 1997). „Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK) – Methoden und Anwendung“. In: *IWi-Hefte* 141, S. 29. URL: <https://www.uni-saarland.de/lehrstuhl/loos/publikationen/iwi-hefte.html> (besucht am 23.03.2020).

- Sonatype Inc. (2020). *Nexus Lifecycle Product*. Nexus Lifecycle Product. URL: https://de.sonatype.com/product-nexus-lifecycle?utm_campaign=NVS&utm_source=ppc&utm_medium=adwords&ahcs_source=paid&utm_term=%2Bnexus%20%2Blifecycle&hsa_tgt=kwd-437257894053&hsa_grp=90875397990&hsa_src=s&hsa_net=adwords&hsa_mt=b&hsa_ver=3&hsa_ad=406628330148&hsa_acc=2665806879&hsa_kw=%2Bnexus%20%2Blifecycle&hsa_cam=8625747087&gclid=EAiaIQobChMIgsvQt8mP6AIVh-h3Ch29zQJJEAAYASAAEgK02fD_BwE (besucht am 10.03.2020).
- Staud, Josef L. (2006). *Geschäftsprozessanalyse: ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware*. 3. Aufl. OCLC: 180896033. Berlin: Springer. 538 S. ISBN: 978-3-540-24510-0. URL: <https://doi.org/10.1007/3-540-37976-2>.
- SV SparkassenVersicherung (2019). *SV kompakt – Profil und Positionen*. Hrsg. von Sylvia Knittel. Unter Mitarb. von Stefanie Rösch. URL: https://www.sparkassenversicherung.de/export/sites/svag/_resources/download_galerien/die_sv/geschaeftsberichte/SV_Kompakt_2020.pdf (besucht am 15.04.2020).
- The People of the GnuPG Project (7. Jan. 2020). *The GNU Privacy Guard*. Library Catalog: gnupg.org Publisher: The GnuPG Project. URL: <https://gnupg.org/> (besucht am 02.04.2020).
- Volere Requirements Specification Template* (19. Aug. 2019). URL: <https://www.volere.org/templates/volere-requirements-specification-template/>.
- Weinhardt, Christof et al. (Okt. 2009). „Cloud Computing – A Classification, Business Models, and Research Directions“. In: *Bus. Inf. Syst. Eng.* 1.5, S. 391–399. ISSN: 1867-0202. DOI: 10.1007/s12599-009-0071-2. URL: <http://link.springer.com/10.1007/s12599-009-0071-2> (besucht am 13.03.2020).