

Product-Service Systems Engineering for Food Retail

**Systematische Entwicklung softwarebasierter hybrider Leistungsbündel
in der Lebensmittelindustrie**

Inauguraldissertation

zur Erlangung des akademischen Grades eines Doktors der
Wirtschaftswissenschaften des Fachbereichs
Wirtschaftswissenschaften der Universität Osnabrück

vorgelegt von

Alexander Fuchs
Diplom-Wirtschaftsinformatiker

Osnabrück, Januar 2023

Dekan: Prof. Dr. Frank Teuteberg
Referenten: Prof. Dr. Oliver Thomas
Prof. Dr. Frank Teuteberg

Inhaltsverzeichnis

I. Dachbeitrag	5
1 Ausgangssituation	6
2 Motivation und Zielsetzung	7
3 Einordnung	7
4 Methodik	10
4.1. Forschungsfragen	11
4.2. Methodenspektrum	12
4.3. Forschungsplan	14
5 Ergebnisse	14
5.1. Überblick	16
5.2. Implikationen	28
5.3. Limitationen	30
Literatur	33
II. Einzelbeiträge	38
1 Wertorientiertes Vendor Managed Inventory durch IT-gestützte Steuerung	39
2 Von der klassischen zur agilen Softwareentwicklung – Evolution der Methoden am Beispiel eines Anwendungssystems	40
3 Designing Customer-Specific Product-Service Systems in B2B Markets	41
4 DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter	42
5 Machine Learning for Demand Forecasting – A Case Study in the German Food Retail Sector	43

Abbildungsverzeichnis

3.1. Epistemologische Positionierung der Dissertationsbeiträge	10
4.1. Forschungsplan	15
5.1. Rolle der Einzelbeiträge im Erkenntnisprozess	16
5.2. Ordnungsrahmen für Vendor Managed Inventory	19
5.3. Agile Softwareentwicklung und Wasserfallmodell im Vergleich	21
5.4. Durch DevOps-Praktiken geprägte IT-Architektur	25
5.5. Zuordnung von DevOps-Teams zu Services	25
5.6. Was-wäre-wenn-Simulation für definierbare Parameter	29
5.7. KPI-Simulation für verschiedene Konfigurationen des Sicherheitsbestands	29

Tabellenverzeichnis

3.1. Erkenntnistheoretische Paradigmen der Wirtschaftsinformatik	9
4.1. Erkenntnisinteressen in der Wissenschaft	12
5.1. Publikationshistorie der Einzelbeiträge	17
5.2. Servicekonfiguration	18
5.3. Methodenvergleich anhand verschiedener Zielkriterien	22
5.4. Kennzahlen zur Dynamik der Anwendungsentwicklung	27
5.5. Werkzeuge zur Anwendung von DevOps-Praktiken	27

Teil I.

Dachbeitrag

1. Ausgangssituation

Die digitale Transformation bezeichnet einen andauernden Veränderungsprozess in unserer Gesellschaft, der von Informationstechnologien getrieben wird. Die technologische Entwicklung ist rasant und verändert die Art, wie wir kommunizieren, arbeiten, Entscheidungen treffen und alltägliche Dinge bewältigen. In wirtschaftlicher Hinsicht sind vor allem Unternehmen davon betroffen, die ihre Leistungen in immer kürzer werdenden Abständen auf veränderte Gewohnheiten und Erwartungen ihrer Kunden ausrichten müssen (Bosch 2016). Die disruptive Kraft der digitalen Transformation ermöglicht es, branchenfremden Wettbewerbern, neue Märkte zu betreten und die Marktanteile etablierter Unternehmen zu bedrohen (Streibich 2018). Die Art, wie Unternehmen untereinander konkurrieren, Werte schaffen und mit ihren Geschäftspartnern und Kunden interagieren, verändert sich dabei grundlegend.

Die hohe Konkurrenz auf globalisierten Märkten sowie die zunehmende Standardisierung und Qualitätsangleichung erschwert die Umsetzung von Differenzierungsstrategien über Produkte. Der Wettbewerb um Käufer für industriell gefertigte Produkte wird vorrangig über den Preis ausgetragen (Thomas, Loos und Nüttgens 2010). Unternehmen stehen vor der Herausforderung, sich diesen Veränderungen entgegenzustellen und ihre eigene Innovationskraft zu beweisen (Oswald und Krcmar 2018). In diesem Zuge durchlaufen Unternehmen einen strukturellen Wandel und erweitern ihr Leistungsangebot hin zu ganzheitlichen, kundenzentrischen Lösungen. Produzierende Unternehmen ergreifen die Chance, das eigene Angebot an Produkten um komplementäre Dienstleistungen zu einem hybriden Leistungsbündel zu erweitern. Sie erwirtschaften somit immer größer werdende Anteile ihrer Umsätze und Erträge mit Dienstleistungen (Thomas, Loos und Nüttgens 2010).

Im Zuge der digitalen Transformation nimmt Software als Baustein für innovative Produkte und Dienstleistungen eine immer wichtiger werdende Schlüsselrolle ein. Maschinenbauer wenden zunehmend größere Teile ihrer Entwicklungsbudgets für Softwareentwicklung auf (Vyatkin 2013). In der Automobilindustrie ist Software längst zum entscheidenden Wettbewerbsfaktor geworden (Ebert und Favaro 2017). Die Erbringung moderner, bedarfsgerechter Dienstleistungen ist ohne IT- und Softwareunterstützung nicht denkbar (Thomas, Nüttgens und Fellmann 2017). Die Wettbewerbsfähigkeit der deutschen Wirtschaft beruht somit zunehmend auf der Fähigkeit, softwareintensive Produkte und Dienstleistungen mit höchster Qualität zu erstellen (Broy u. a. 2006). Neben der Erhöhung des Kundennutzens ist die Verkürzung von Markteinführungszeiten das erklärte Ziel. Für Unternehmen ist es daher entscheidend, Software – und damit Informationssysteme – so zu gestalten, dass sie als wirksames Instrument zur Verbesserung der für den Kunden erbrachten Leistungen genutzt werden können. Software wird somit zum Enabler für die Erbringung innovativer, bedarfsgerechter Dienstleistungen mit

kundenzentrischem Werteversprechen.

Die vorliegende Arbeit veranschaulicht diese Entwicklung am Beispiel der Lebensmittelindustrie. Als Anwendungsszenario dienen Produkt-Service-Systeme für den Lebensmitteleinzelhandel, welche anhand von Fallstudien untersucht und weiterentwickelt werden.

2. Motivation und Zielsetzung

Im industriellen Umfeld beruhen Differenzierungsstrategien von Unternehmen seit Langem auf dem immateriellen Dienstleistungsanteil (Thomas und Scheer 2003, Goedkoop u. a. 1999). „Hybride Produkte“ oder „Produkt-Service-Systeme“ (PSS) sind gesamtheitliche Leistungsbündel, bei denen der Kunde nicht zwischen dem materiellen und immateriellen Anteil unterscheiden können muss (Blinn u. a. 2010). Die hybride Wertschöpfung erfolgt durch die Bereitstellung von Produkt-Service-Systemen mit dem Ziel, Kundenbedürfnisse ganzheitlich zu erfüllen (Tukker und Tischner 2006, Goedkoop u. a. 1999). Produktbegleitende Dienstleistungen sind in der Regel darauf ausgerichtet, die über den Lebenszyklus des betreffenden Sachobjekts auftretenden Anforderungen und Bedarfe zu erfüllen. Dabei ist es von zunehmender Bedeutung, inwiefern Dienstleistungen auf der Grundlage einer breiten Informationsbasis weiterentwickelt werden können. In diesem Sinne gelten *Smart Services* als kontextsensitive, an die Bedürfnisse des Kunden angepasste Dienstleistungen (Thomas, Nüttgens und Fellmann 2017). Bullinger, Ganz und Neuhüttler (2017) beschreiben Smart Services als „datenbasierte, individuell konfigurierbare Angebote aus Dienstleistungen, digitalen Diensten und Produkten [...]“. In der Regel basieren derartige Dienste in hohem Maße auf IT- und Softwareunterstützung.

Der Einsatz von PSS in der Lebensmittelbranche ist weitgehend unerforscht und daher ein relevanter Untersuchungsgegenstand. Diese Arbeit verfolgt das Ziel, Gestaltungswissen für die Konzeption und Implementierung hybrider Leistungsbündel und dazugehöriger Informationssysteme im Kontext der Lebensmittelindustrie zu generieren. Auf Basis von Fallstudien wird die Perspektive eines Lebensmittelproduzenten eingenommen, der den deutschen Lebensmitteleinzelhandel bedient. Dabei werden Schwerpunkte auf die Modellierung der Dienstleistungen sowie auf zielgerichtete Softwareentwicklung unter Anwendung moderner Technologien und Verfahrensweisen gelegt. Hauptziel ist es, die in diesem Rahmen auftretenden Untersuchungsgegenstände zu erfassen, zu erklären und weiterzuentwickeln. Es sollen somit Erkenntnisse über die Art der vorliegenden Problemstellungen sowie praktikable Lösungsansätze aufgezeigt werden, um einen Beitrag zur Erweiterung der vorhandenen Wissensbasis aus wissenschaftlicher und praktischer Perspektive zu leisten.

3. Einordnung

Die vorrangigen Untersuchungsgegenstände der Wirtschaftsinformatik sind betriebliche Informationssysteme sowie die Rahmenbedingungen für ihre Entwicklung, Einführung und Nutzung (Becker, Holten u. a. 2003). Die Ursprünge der Wirtschaftsinformatik gehen auf die Wirtschaftswissenschaften sowie auf die Informatik zurück, was ihre interdisziplinäre Ausrichtung begründet. Sie fungiert als „anwendungsorientierter Vermittler zwischen der Betriebswirtschaftslehre und der Informatik [...]“ (Thomas 2006). Die Forschung in der Wirtschaftsinformatik orientiert sich an den unterschiedlichen Forschungsmethoden der Mutterdisziplinen. Während vor Allem die Betriebswirtschaftslehre stark durch empirische und verhaltenswissenschaftliche Forschungsstile gekennzeichnet ist, werden in der Informatik überwiegend formalwissenschaftliche Methoden verfolgt, u.a. zur Konstruktion technischer Systeme. Die wirtschaftsinformatische Forschung ist bis heute geprägt durch den in diesem Spannungsfeld entstandenen Methodenpluralismus (Becker, Holten u. a. 2003).

Die Wirtschaftsinformatik ist traditionell eine Disziplin, in der die praktische Anwendbarkeit von Erkenntnissen von Beginn an wichtig war. Robra-Bissantz und Strahringer (2020) unterteilen die klassische deutsche Wirtschaftsinformatik in zwei Ausprägungen. Zum einen bediente sie sich analytischer Optimierungsmethoden, wie beispielsweise aus dem Bereich des Operations Research, mit dem Ziel, IT-Systeme auf die Erfüllung bestimmter Zielsetzungen auszurichten. Typische Anwendungsfälle können den Bereichen Produktion und Logistik entnommen werden, wie etwa die Optimierung einer Zielfunktion (z. B. eine Kostenfunktion) unter Berücksichtigung bestimmter Nebenbedingungen (z. B. begrenzte Kapazitäten). Zum anderen bildete das Software Engineering einen ingenieurwissenschaftlich geprägten Ansatz für sämtliche Prozesse für die Entwicklung und den Betrieb von Software. Robra-Bissantz und Strahringer (2020) führen an, dass weder die reine Anwendung von Optimierungsmethoden noch die Entwicklung von Software mit starkem Anwendungsbezug dafür bekannt waren, wissenschaftliche Strenge (rigor) im Sinne von Hevner (2007) oder Gregor (2002) zur Maßgabe zu machen. Darüber hinaus entwickelte sich die Erkenntnis, dass Informationstechnologie nicht nur isoliert, sondern mit ihren Beziehungen und ihrem Wirken auf den Mensch untersucht werden muss (Jones und Gregor 2007). Die Beurteilung, wann ein Anwendungssystem im Sinne der Interaktion von Mensch und Maschine in einem betrieblichen Kontext, als nützlich oder gut gestaltet gilt, verlangt eine soziotechnische Betrachtung. Es sind demnach nicht alleine Effizienzkriterien für Informationssysteme von Belang, sondern auch Fragen der Effektivität im Hinblick auf Zufriedenheit der Nutzer (Usability), auf die Erreichung von Unternehmenszielen sowie auf die Verfolgung gesellschaftlicher Ziele (Robra-Bissantz und Strahringer 2020). Dieser Einsicht folgend, zeichnet sich die wirtschaftsinformatische Forschung durch ihre ganzheitliche und multidisziplinäre Betrachtungsweise aus.

Als Orientierung zur Auswahl geeigneter Forschungsmethoden ist die Unterscheidung von Erkenntnis- und Gestaltungszielen maßgeblich. In dieser Hinsicht haben sich *Behaviorismus* und *Design Science* als epistemologische Paradigmen der Wirtschaftsinformatik herausgebildet (Hevner u. a. 2004, Tabelle 3.1). Ein wesentliches Merkmal der

3. Einordnung

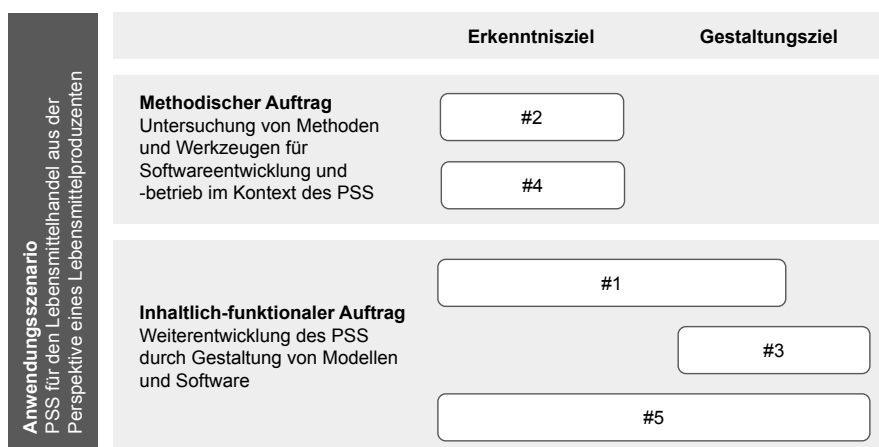
Tabelle 3.1. Erkenntnistheoretische Paradigmen der Wirtschaftsinformatik

	Behaviorismus	Design Science
Forschungsfrage	Wie und wieso?	Wie gut?
Forschungsergebnis	Theorien	IT-Artefakte
Forschungsaktivitäten	Theoriebildung Theorieüberprüfung	Konstruktion von Artefakten Bewertung von Artefakten
Forschungsziel	Wahrheit	Nützlichkeit

behavioristischen Forschung, deren Wurzeln auf den naturwissenschaftlichen Bereich zurückgehen, liegt in der Formulierung und Überprüfung von Theorien über den Einsatz von Informationssystemen. Mit dem Ziel der Wahrheitsfindung werden empirische Methoden zur Analyse des Verhaltens und der Auswirkungen von Informationssystemen auf Organisationen angewandt (Wilde und Hess 2006, S. 3, Becker und Pfeiffer 2006, S. 2). Forschung, die höchsten Ansprüchen an wissenschaftliche Strenge (research rigor) genügen soll, und nicht notwendigerweise auf die Lösung von Problemen der Praxis fokussiert sein muss, findet in der englischsprachigen, verhaltensorientierten Forschung des Information Systems viele Anhänger (Robra-Bissantz und Strahringer 2020, S. 164). Unterdessen legt Design Science Research (DSR) als Zweig des Information Systems Research überwiegend ingenieurwissenschaftliche Prinzipien für die Gestaltung und Entwicklung von Informationssystemen zugrunde. Peffers, Tuunanen und Niehaves (2018) zählen die gestaltungsorientierte Wirtschaftsinformatik nach Österle u. a. (2010) als eine Ausprägung von DSR. Ziel ist es, methodisch fundiert (rigor) zu handlungsleitenden Aussagen mit hohem Praxisbezug (relevance) zu gelangen. Der Erkenntnisgewinn durch Konstruktion und Evaluation von IT-Lösungen steht dabei im Mittelpunkt. Letztere sind Artefakte, die im Rahmen der Analyse, des Entwurfs sowie der Implementierung von Informationssystemen entstehen oder eingesetzt werden (Becker und Pfeiffer 2006). IT-Artefakte werden in verschiedene Typen unterteilt, darunter u.a. Sprachen, Modelle (Unternehmensmodelle, Referenzmodelle), Methoden (Verfahren und Algorithmen) sowie Instanzen bzw. Implementierungen (Vaishnavi und Kuechler 2015). Vielfach werden prototypische Umsetzungen als Mittel zur Überprüfung der Gültigkeit von Forschungsergebnissen eingesetzt (Becker, Niehaves u. a. 2009).

Das behavioristische und das konstruktionsorientierte Paradigma können als unterschiedliche, aber komplementäre Vorgänge der Information Systems Forschung verstanden werden. Während die konstruktionsorientierte Forschung IT-Artefakte erstellt, versucht die behavioristische Forschung Theorien darüber zu bilden und deren Wahrheitsgehalt zu überprüfen. Empirisch evidente Theorien liefern der konstruktionsorientierten Forschung Ansatzpunkte für die Schaffung neuer IT-Artefakte (Becker und Pfeiffer 2006, S. 3). Über die Vereinigung von Relevanz und Rigorosität leistet DSR mit seinen

Abbildung 3.1. Epistemologische Positionierung der Dissertationsbeiträge



zahlreichen Ausprägungen einen Beitrag zur Verbindung von praktischer Nützlichkeit mit wissenschaftlichem Erkenntnisgewinn. Ungeachtet dessen existiert auch jenseits von DSR Raum für praxisorientierte Forschung in der Wirtschaftsinformatik, so z. B. über die klassische Fallstudienforschung (Robra-Bissantz und Strahringer 2020, S. 165).

Im Hinblick auf den Forschungsschwerpunkt liegt ein maßgebliches Unterscheidungskriterium in der Abgrenzung des methodischen Auftrags vom inhaltlich-funktionalen Auftrag (Becker, Holten u. a. 2003). Der methodische Auftrag hat das Verständnis und die Entwicklung von Methoden und Techniken zur Beschreibung, Entwicklung, Einführung und Nutzung von Informationssystemen zum Ziel, während der inhaltlich-funktionale Auftrag auf das Verständnis und der Gestaltung von Informationssystemen für betriebswirtschaftliche Branchen, einzelne Betriebe oder Anwendungsfälle gerichtet ist.

Ziel dieser Forschungsarbeit ist es, Erkenntnisse über real existierende Problemstellungen zu erlangen, um daraus Gestaltungsempfehlungen für praxistaugliche Lösungsansätze abzuleiten. Es wird angestrebt, ein realwissenschaftliches Verständnis über die Problemstellung, die Art und Weise der Lösungsfindung sowie zur Lösung selbst zu erlangen. Im Forschungsprozess werden die Generierung von Erkenntnissen auf Basis von Beobachtungen des Untersuchungsgegenstandes im Rahmen von Fallstudien sowie die Konstruktion und Evaluation von IT-Artefakten nach dem DSR-Paradigma (Thomas 2006, S. 15) eine zentrale Rolle einnehmen. Die epistemologische Einordnung der im Forschungsprozess erstellen Beiträge mit den Bezeichnungen #1 bis #5 erfolgt anhand Becker, Holten u. a. (2004) in eine Auftrags- und Zielmatrix (Abbildung 3.1). Über die Analyse fachlich motivierter Problemstellungen und die Gestaltung von Lösungsansätzen im Rahmen der gegebenen Fachdomäne wird dem inhaltlich-funktionalen Auftrag nachgegangen (Beiträge #1, #3 und #5). Dem methodischen Auftrag wird über die Untersuchung von Methoden und Werkzeugen im Hinblick auf Entwicklung und Betrieb der zugrundeliegenden Informationssysteme sowie individueller Software entsprochen (#2 und #4).

4. Methodik

4.1. Forschungsfragen

Die in der Ausgangssituation beschriebene Problemstellung wird in eine zusammenfassende Hauptforschungsfrage überführt und wie folgt formuliert.

FF. *Wie können Produkt-Service-Systeme für den Lebensmittelhandel entwickelt und betrieben werden?*

Die durch die Forschungsfrage adressierte Problemstellung ist durch ein mehrdimensionales Erkenntnisinteresse geprägt. Eberhard (1999) unterscheidet phänomenale, kausale und aktionale Erkenntnisinteressen (Tabelle 4.1). Die Behandlung der Problemstellung erfordert vorrangig die Untersuchung realer Gegebenheiten (phänomenales Erkenntnisinteresse) und deren Ursachen (kausales Erkenntnisinteresse) sowie die Gestaltung von Lösungsansätzen zu deren Verbesserung (aktionales Erkenntnisinteresse). Um die Behandlung der kompakten Hauptforschungsfrage zu ermöglichen, wird eine Zerlegung in Teilforschungsfragen vorgenommen. Die Zerlegung erfolgt derart, dass die Teilforschungsfragen weitgehend abgegrenzt voneinander untersucht werden können.

FF 1. *Welche aktuelle und perspektivische Bedeutung hat das Produkt-Service-System im Rahmen des gegebenen Anwendungsszenarios?*

Dieser Forschungsfrage liegt ein phänomenales Erkenntnisinteresse zu Grunde, welchem durch eine Untersuchung des Gegenstands und seiner Umwelt nachgegangen wird. Die Bearbeitung der Forschungsfrage erfolgt mit dem Zweck, eine Ausgangsbasis und Orientierung für die Behandlung der nachfolgenden, weiterführenden Forschungsfragen zu liefern.

FF 2. *Welche Methoden und Technologien eignen sich für Softwareentwicklung und -betrieb für das Produkt-Service-System auf Basis der gegebenen Anwendungslandschaft?*

Die Forschungsfrage ist primär durch ein aktionales Erkenntnisinteresse geprägt. Zweck ist es, aufzuzeigen, was zu tun ist, um eine konkrete Aufgabenstellung zu bewältigen. Die begründete Herleitung geeigneter Maßnahmen im Sinne der Anwendung nützlicher Methoden und Technologien erfordert die Untersuchung kausaler Zusammenhänge, sodass nachrangig auch ein kausales Erkenntnisinteresse besteht.

FF 3. *Wie kann das Produkt-Service-System kontextsensitiv auf Kundenbedürfnisse angepasst werden?*

Die Forschungsfrage ist primär durch ein aktionales Erkenntnisinteresse geprägt und verlangt konkrete Gestaltungsempfehlungen für eine praktische Umsetzung in einem real gegebenen Anwendungsszenario.

4. Methodik

Tabelle 4.1. Erkenntnisinteressen in der Wissenschaft

Erkenntnisinteresse	Fragestellung	Gegenstand
Phänomenal	Was ist los? Was geschieht?	Verständnis über die faktischen Gegebenheiten eines Problems
Kausal	Warum ist das so? Warum geschieht es?	Erfassung der Ursachen eines Problems
Aktional	Was ist zu tun?	Identifikation von Maßnahmen zur Bewältigung des Problems

4.2. Methodenspektrum

Mit Blick auf den praktischen Fokus der Forschungsfragen wird grundsätzlich nach dem Design-Science-Ansatz in Anlehnung an Hevner u. a. (2004) und Peffers, Tuunanen, Rothenberger u. a. (2008) gearbeitet. Unter Berücksichtigung einschlägiger Literatur werden auf Basis praktischer Erwägungen IT-Artefakte konstruiert, darunter insbesondere Modelle und Softwareimplementierungen, deren Nützlichkeit für die gegebene Aufgabenstellung im Rahmen von Fallstudien untersucht wird. Die Evaluierung der Lösungsansätze findet anhand direkter Anwendung oder Demonstration im unmittelbaren Austausch mit Fachexperten und Endanwendern statt. Das in dieser Arbeit angewandte Methodenspektrum umfasst die folgenden Elemente.

- Eine *Methode* ist im Allgemeinen eine strukturierte Herangehensweise, die sich durch eine bestimmte Auswahl von Instrumenten als Mittel der Zielerreichung auszeichnet. Ein wissenschaftlicher Anspruch ist grundsätzlich dann gegeben, wenn das Vorgehen durch intersubjektiv nachvollziehbare und nachprüfbarere Verhaltensregeln gekennzeichnet ist (Wilde und Hess 2006, S. 2). Zudem muss die Methode einen abgeschlossenen Forschungsbeitrag im Hinblick auf erkenntnistheoretischen Paradigmen der Wirtschaftsinformatik leisten können (ebd. S. 5). Sind beide Kriterien erfüllt, wird die Methode als *vollständig* angesehen. Da Methoden hierarchisch in Form einer Teil-Ganzes-Beziehung zusammengesetzt werden können, ist zu unterscheiden, ob eine Methode eigenständig im Sinne der Abgeschlossenheit ist oder eine Teilkomponente darstellt.
- *Beobachtungen, Befragungen* oder *Interviews* sind anerkannte Verfahren zur Datenerhebung. Bei großzahliger angelegten Befragungen können in der Regel qualitative oder quantitative Rückschlüsse auf die Grundgesamtheit gebildet werden. Daneben ist das Interview in Form einer direkten Expertenbefragung eine häufig angewandte, rein qualitativ-empirische Forschungsmethode, die beispielsweise zur Evaluation von IT-Artefakten eingesetzt werden kann. Im Wege direkter Kommunikation

4. Methodik

sollen Informationen gewonnen werden, die anderweitig nicht oder nur schwierig erfasst werden können. Zu unterscheiden sind strukturierte Interviews (vordefinierte Fragen mit vordefinierten Antwortmöglichkeiten), semi-strukturierte Interviews (vordefinierte Fragen, freie Antwortmöglichkeiten) sowie unstrukturierte Interviews (freie Gespräche) (Oates 2005).

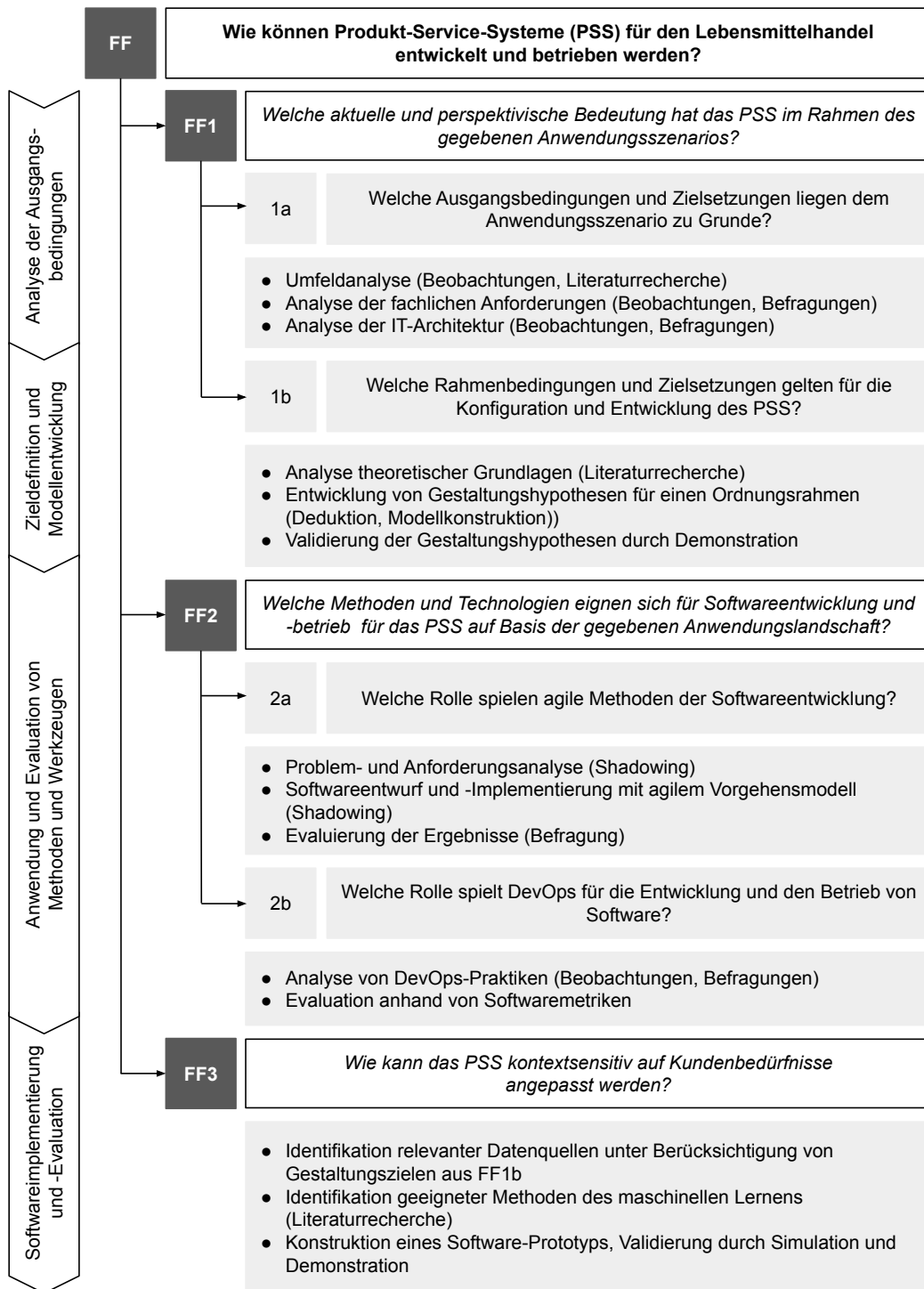
- *Shadowing*, auch *participant observation*, steht für die Beobachtung von Akteuren, deren Verhalten für die zu untersuchende Problemstellung von Bedeutung ist. Die Beobachtung des Untersuchungsgegenstandes erfolgt auf direkte Weise in Anwesenheit des Wissenschaftlers. Es bietet sich die Möglichkeit, durch spontane, kontextabhängige Interaktion mit den Beteiligten weiterführende Erkenntnisse zu erlangen, die durch eine indirekte Erhebung nicht oder nur schwierig zu erfassen sind (Myers 2009, S. 137).
- Die *Literaturrecherche* ist eine Methode, um den Stand der Forschung zu einer gegebenen Problemstellung zu ermitteln. Sie liefert eine erkenntnistheoretische und methodische Ausgangsbasis für neue Forschung. Dadurch wird es möglich, Forschungslücken aufzudecken sowie neues Wissen durch die Kombination von bereits vorhandenem Wissen zu erzeugen.
- *Deduktion* steht als Grundbegriff der Logik für die Ableitung spezieller Aussagen auf Basis von Theorien. Wilde und Hess (2006) unterscheiden je nach Grad der zugrundeliegenden Formalität sowie dem Modellbezug die formal-deduktive von der konzeptionell-deduktiven und argumentativ-deduktiven Analyse.
- Die *Fallstudie* ist eine empirische Forschungsmethode (Yin 2009), welche der Erlangung von Erkenntnissen zu einem bestimmten Untersuchungsgegenstand dient sowie zu dessen Beschreibung und Erklärung eingesetzt werden kann (Riedl 2006, S. 117ff). Sie untersucht in der Regel komplexe, schwer abgrenzbare Phänomene in ihrem realen Kontext. Unter Verzicht auf großzahlige Querschnittsanalysen werden nur einzelne oder wenige Merkmalsträger intensiv untersucht. Die Fallstudie ist sowohl geeignet, um Hypothesen zu generieren als auch um die Gültigkeit von Theorien zu testen. Die Untersuchung kann daher ebenso verhaltenswissenschaftlich wie konstruktivistisch motiviert sein (Wilde und Hess 2006, S. 9) und ist vollständig nach obiger Definition.
- Die Konstruktion von *Modellen* erfolgt zur Erklärung und Gestaltung realer Systeme. Modelle werden dabei verstanden als immaterielles Abbild der Realwelt für die Zwecke eines Subjekts (Becker, Rosemann und Schütte 1995). Sie werden eingesetzt, um Erkenntnisse über Eigenschaften und Zusammenhänge des zugrundeliegenden realen Gegenstands zu erhalten. Bei der Modellierung handelt es sich um ein konstruktivistisches Verfahren, welches entweder induktiv (ausgehend von Beobachtungen) oder deduktiv (bspw. abgeleitet aus Theorien) erfolgt. In der Wirtschaftsinformatik werden zahlreiche Typen von Modellen unterschieden, darunter u.a. Referenzmodelle, Vorgehensmodelle, Metamodelle oder Ordnungsrahmen.

- Die *Simulation* kommt zum Einsatz bei Problemen, die zwar formalisierbar aber analytisch nicht oder nur sehr schwierig zu lösen sind. Dabei wird das Verhalten des Untersuchungsgegenstands in einem Simulationsmodell abgebildet, Umgebungsvariablen durch die Justierung von Modellparametern und Zufallsvariablen nachgestellt und die Wirkung auf die Zielgrößen bewertet (Wilde und Hess 2006, S. 6). Die Simulation ist eine vollständige Forschungsmethode.
- *Prototypen* sind Instanziierungen von IT-Artefakten im Sinne von DSR. Sie bestehen aus Software und ggf. Daten und können Ausschnitte aus einem Gesamtsystem darstellen. Der sogenannte Relevance Cycle nach Hevner (2007) sieht vor, Prototypen im praktischen Umfeld bei Unternehmen zu testen (Gregor und Hevner 2013, S. 343). Auf diese Weise findet eine praktische Evaluation als Bestandteil der Forschung statt (Robra-Bissantz und Strahringer 2020) und die Methode gilt als vollständig. In der Praxis der Softwareentwicklung haben sich verschiedene Vorgehensmodelle durchgesetzt, die auf Prototyping setzen, um kurzfristig lauffähige Vorabversionen oder Erweiterungen eines Anwendungssystems zu entwickeln und zu testen.

4.3. Forschungsplan

Aus den vorgenannten Überlegungen leitet sich der in Abbildung 4.1 dargestellte Forschungsplan ab, der sich aus den identifizierten Forschungsfragen und den im Forschungsprozess eingesetzten Methoden zusammensetzt. Die Zerlegung der Hauptforschungsfrage ergibt drei abgegrenzte Teilforschungsfragen, die im Rahmen von Fallstudien unter Rückgriff auf das in Kapitel 4.2 angeführte Methodenspektrum behandelt werden. Teilforschungsfrage 1 untersucht die Bedeutung des PSS im Kontext des gegebenen Anwendungsszenarios. Darunter fällt die Analyse des Status Quo sowie die Identifikation von Zielsetzungen, die mit dem Einsatz des PSS verbunden sind. Um die Grundlage für gezielte Weiterentwicklungen zu schaffen, wird ein Modell zur Beschreibung der mit dem PSS verbundenen Dienstleistungen konstruiert. Teilforschungsfrage 2 widmet sich den vorherrschenden Ausgangsbedingungen für Softwareentwicklung und -Betrieb im Kontext der betrieblichen Informationssysteme, die dem PSS zugrunde liegen. Dabei wird die Modernisierung der eingesetzten Methoden und Werkzeuge untersucht, um Erkenntnisse zu gewinnen. In Teilforschungsfrage 3 wird die Weiterentwicklung des PSS unter Berücksichtigung der aus den Teilforschungsfragen 1 und 2 ermittelten Erkenntnisse adressiert. In diesem Zuge wird ein Softwareprototyp zur Erreichung der in Teilforschungsfrage 1 ermittelten Zielsetzungen unter Anwendung der in Teilforschungsfrage 2 identifizierten Methoden entwickelt und evaluiert.

Abbildung 4.1. Forschungsplan

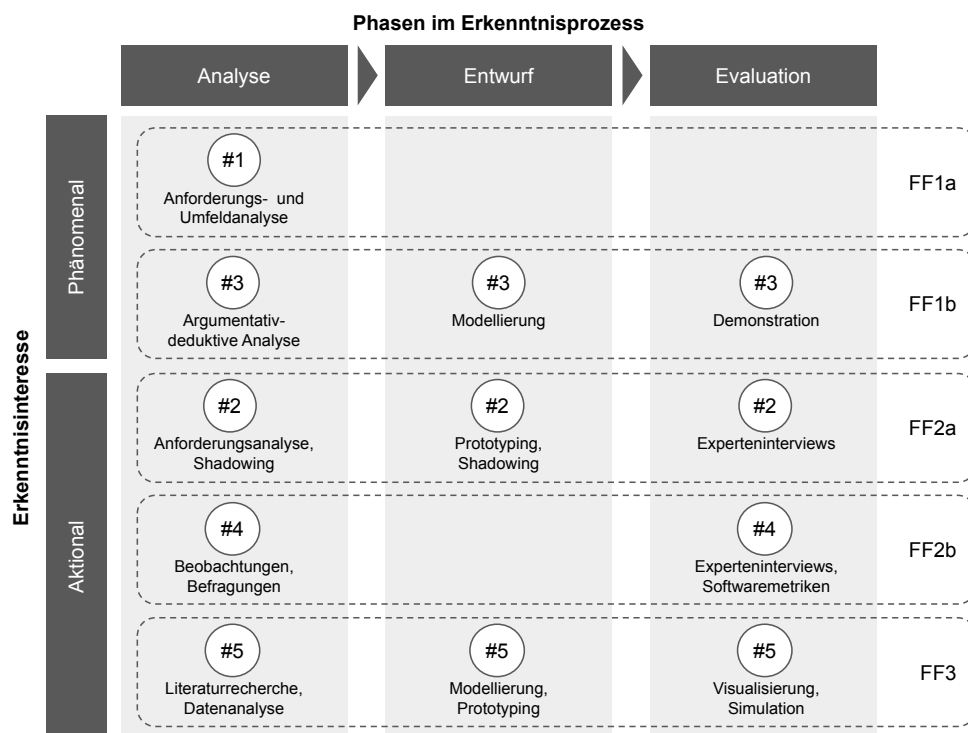


5. Ergebnisse

5.1. Überblick

Die Ergebnisse dieser Forschungsarbeit werden in fünf Einzelbeiträgen erzielt. Der Erkenntnisprozess erfolgt nach Maßgabe der gestaltungsorientierten Wirtschaftsinformatik im Rahmen der Phasen Analyse, Entwurf und Evaluation (Österle u. a. 2010). Die Rolle der einzelnen Beiträge im Erkenntnisprozess wird in Abbildung 5.1 veranschaulicht.

Abbildung 5.1. Rolle der Einzelbeiträge im Erkenntnisprozess



Beitrag #1 untersucht das gegebene Anwendungsszenario aus einer betriebswirtschaftlichen Perspektive und veranschaulicht die Rollen der beteiligten Akteure. Zentraler Untersuchungsgegenstand ist die Leistungsbeziehung eines Anbieters von Lebensmittelprodukten gegenüber dem Einzelhandel. Im Mittelpunkt steht die Bereitstellung von Verkaufsflächen seitens des Handels und der damit verbundene Auftrag an den Hersteller die Flächen möglichst erfolgreich zu bewirtschaften. Um dies zu bewerkstelligen, beauftragt und koordiniert der Hersteller verschiedene Dienstleister. Es wird aufgezeigt, welche operativen und finanziellen Kennzahlen in diesem Szenario von Bedeutung sind und in welchem Zusammenhang sie zueinander stehen. Außerdem wird sichtbar gemacht, welche Prozesse von welchem Akteur ausgeführt werden, welche Informationen dabei entstehen und welche Informationssysteme beteiligt sind. Im Ergebnis wird erkennbar, dass

5. Ergebnisse

Tabelle 5.1. Publikationshistorie der Einzelbeiträge

#	Publikationsorgan	Medium	WK-WI	VHB	Bibliographische Informationen	FF
1	HMD – Praxis der Wirtschaftsinformatik	Journal	B	D	Fuchs, A. , Stolze, C., Breitschwerdt, R., Hucke, S. und O. Thomas (2012), „Wertorientiertes Vendor Managed Inventory durch IT-gestützte Steuerung“	1a
2	HMD – Praxis der Wirtschaftsinformatik	Journal	B	D	Fuchs, A. , Stolze, C., Thomas, O. (2013), „Von der klassischen zur agilen Softwareentwicklung – Evolution der Methoden am Beispiel eines Anwendungssystems“	2a
3	Enterprise Modelling and Information Systems Architectures (EMISA)	Journal	–	C	Fuchs, A. , Bittmann S. und D. Özcan (2015), „Designing Customer-Specific Product-Service Systems in B2B Markets – A Consecutive Framework for Development and Configuration Management“	1b
4	HMD – Praxis der Wirtschaftsinformatik	Journal	B	D	Thomas, O., Varwig, A., Kammeler, F., Zobel, B. und A. Fuchs (2017), „DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter“	2b
5	Universitätsbibliothek Osnabrück	–	–	–	Fuchs, A. , Varwig, A., Kortum, H., Thomas, O. (2022): Machine Learning for Demand Forecasting – A Case Study in the German Food Retail Sector	3

5. Ergebnisse

Tabelle 5.2. Servicekonfiguration

Service	<i>Service Definition</i>		<i>Service Delivery</i>
	Service Level	Nebenbedingungen	Aktionen
Category Management	Aktualisierungsrythmus	–	Planung und Optimierung des Sortiments, Anpassung der Verkaufsmöbel
Disposition	Besuchs- und Bestellfrequenz	Zugriff auf Warenbestand am POS ¹	Bestandsoptimierung und Bestellabwicklung
Distribution	Lieferzeit	Zeitfenster für Warenannahme	Kommissionierung und Versand
Warenverräumung	Besuchsfrequenz	Physischer Zugang zu Verkaufsflächen und Warenlager	Einlagern der Waren auf Verkaufsflächen

¹ Point of Sale (POS) bezeichnet den Verkaufsort.

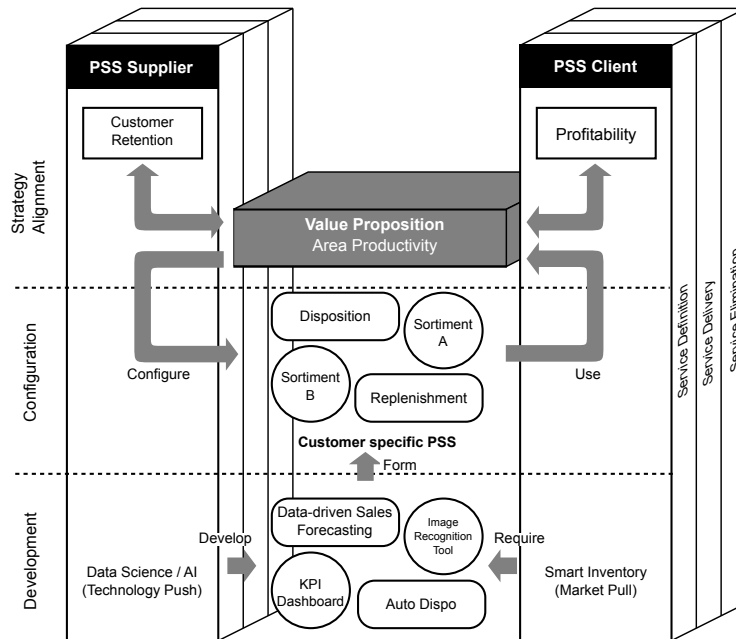
die Zusammenführung entscheidungsrelevanter Informationen (darunter insbesondere logistische Daten wie Bestände und Verkaufsdaten) über System- und Unternehmensgrenzen hinweg das Potenzial bietet, Kosten zu senken und die Servicequalität zu erhöhen. Insgesamt kann dadurch ein Beitrag für eine wertorientierte Ausgestaltung der Leistungsbeziehung erzielt werden.

Diese besondere Form der Zusammenarbeit zwischen Hersteller und dem Einzelhandel erfüllt die Kriterien eines Produkt-Service-Systems, welches durch den Lebensmittelanbieter bereitgestellt wird. Dienstleistungen können heute kaum noch standardisiert und vereinheitlicht über eine größere Kundenbasis erbracht werden (Becker und Klingner 2017). Die Ursache dafür liegt in der Erfordernis einer verstärkten Kundenorientierung, einerseits um Markterfordernisse zu erfüllen, andererseits um Abgrenzungsmerkmale gegenüber Wettbewerbern zu schaffen. Beitrag #3 adressiert diese Aufgabenstellung und untersucht die Individualisierbarkeit des PSS im Hinblick auf kundenspezifische Anforderungen. Zunächst wird ein allgemeiner Ordnungsrahmen für die Entwicklung und Konfiguration von PSS im Kontext von B2B-Geschäftsmodellen entwickelt (Abbildung 5.2). Der Ordnungsrahmen wird durch ausformulierte Gestaltungshypothesen gebildet und visuell veranschaulicht. Als zentrales Element dient die Definition eines Leistungsversprechens (Value Proposition), nach dessen Maßgabe alle Aktivitäten zur Konfiguration und Entwicklung des PSS ausgerichtet werden (Böhmman, Leimeister und Moeslein 2014). Es wird ein Life-Cycle-Management-Ansatz konstruiert, um die regelmäßige und zielgerichtete Aktualisierung der Services zu strukturieren.

Im nächsten Schritt wird der Ordnungsrahmen im Rahmen einer Fallstudie auf ein konkretes Anwendungsszenario im Lebensmitteleinzelhandel angewendet. In diesem Szenario bezieht der Handelskunde von seinem Lieferanten Flächenproduktivität (*as-*

5. Ergebnisse

Abbildung 5.2. Ordnungsrahmen für Vendor Managed Inventory



a-Service) für ein Produktsortiment. Die Flächenproduktivität ist gegeben durch eine Erfolgsgröße (i. d. R. Umsatz) im Verhältnis zur einer definierten Verkaufsfläche, für deren Bewirtschaftung der PSS-Anbieter die Verantwortung übernimmt. Die erforderlichen Services umfassen die Planung des Sortiments, die Distribution, Disposition und Verräumung der Waren (Tabelle 5.2). Im Rahmen der Servicekonfiguration werden Service Level und zugehörige Nebenbedingungen festgelegt. Entwicklungspotenziale sind vorrangig in Verbindung mit dem Einsatz von Informationstechnologie zur intelligenten Steuerung des Bestands sowie der Unterstützung zugehöriger Serviceprozesse identifiziert worden. In diesem Kontext wird der Begriff *Smart Inventory* für das bedarfsgerechte Steuern der Bestände auf Basis aktueller Daten bzw. Ereignisse sowie anhand konfigurierbarer Zielparameter verwendet.

Beitrag #3 untersucht Prozesse zur Entwicklung von Software, die dem PSS zugrunde liegt. Im Mittelpunkt der Betrachtung steht die Gestaltung von Informationssystemen zur Abbildung der vertrieblichen Rahmenbedingungen. Hierunter fällt insbesondere die Verwaltung von Konditionen wie Preise, Rabatte und sonstiger Abzüge, die im Zuge der Bereitstellung des PPS abzurechnen sind. Die Konditionen sind das Ergebnis von Verhandlungen und werden abhängig von der kundenindividuellen Konfiguration des PSS vereinbart. Zur besonderen Relevanz von Konditionsvereinbarungen zwischen der Lebensmittelindustrie und dem Lebensmitteleinzelhandel existieren zahlreiche Untersuchungen (Lademann 2015, Lademann 2012, Schlippenbach und Pavel 2011, Baumgartner und Schölling 2000). Lademann (2012) stellt für die Ernährungswirtschaft einen einzigartigen, branchentypischen Koordinationsbedarf fest. Ein wichtiger Aspekt für die

5. Ergebnisse

Gestaltung nachhaltiger Konditionssysteme ist das Prinzip der Leistungsbezogenheit, nach dem die Gewährung einer Kondition grundsätzlich immer an die Erbringung einer Gegenleistung durch den Kunden gekoppelt werden muss (Bendl 2000).

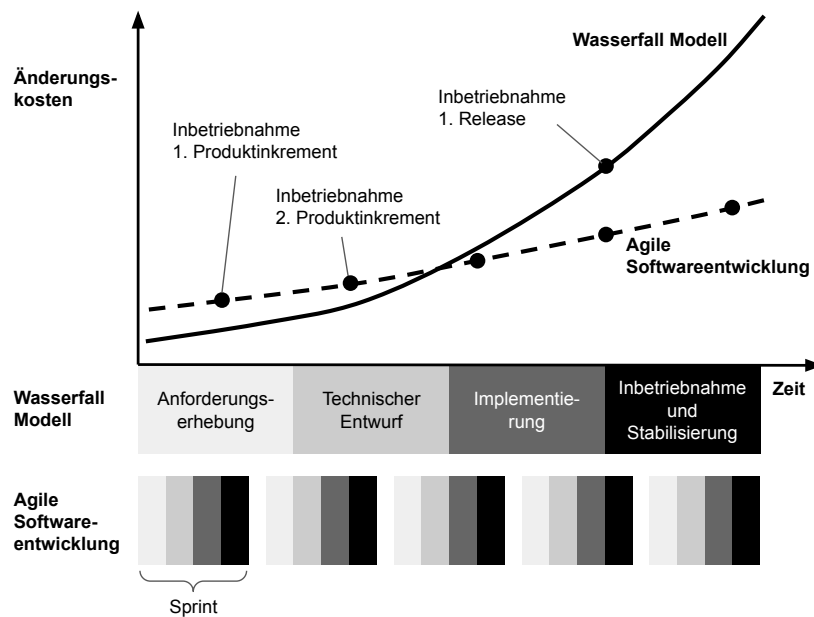
Die Abbildung leistungsbezogener Konditionen unter Berücksichtigung kundenindividueller Konfigurationen des PSS war mit den Mitteln der im Einsatz befindlichen Vertriebs- und Abrechnungssoftware nur schwierig zu bewerkstelligen, da passende Datenmodelle fehlten. Damit war der Ausgangspunkt für die Entscheidung zur Einführung eines neuen Systems zur Verwaltung der Konditionen gegeben. Davon ausgehend wurden zwei Softwareentwicklungsprojekte initiiert und nacheinander umgesetzt. Ersteres war auf Grundlage des Wasserfallmodells erfolgt, letzteres unter Anwendung agiler Softwareentwicklung, nachdem das im Rahmen der Wasserfallmethode entwickelte System die Anforderungen nicht vollständig erfüllen konnte. Im Rahmen von Beitrag #2 wurde ein Review mit den beteiligten Stakeholdern zum Hergang des ersten Projekts durchgeführt. Zum Kreis der Diskussionspartner gehörten Vertriebsleitung, Endanwender, IT-Leitung und Softwareentwickler. Als Untersuchungsmittel wurden Expertenbefragungen in Form semi-strukturierter Interviews, die Sichtung und Analysen der eingesetzten Technologien, Tools und des Quellcodes, sowie Shadowing der Softwareentwickler und Endanwender eingesetzt. Das Review ergab folgende Beobachtungen, die ausschlaggebend für den Übergang zu agilen Methoden waren:

- Der Start der Implementierungsphase wurde eingeleitet, ohne die Phase der Anforderungserhebung vollständig abzuschließen, um im Zeitplan zu bleiben
- Die Vervollständigung der Anforderungsbeschreibung während der Implementierungsphase konnte erreicht werden, nachdem erste Implementierungsergebnisse demonstriert worden waren und das Feedback von Fachexperten und Endanwendern vorlag
- Während der Implementierungsphase entstanden neue Anforderungen, nachdem erste Teile der Anwendung durch die Endanwender getestet worden waren
- Während der Implementierungsphase entstanden neue Anforderungen aufgrund neuer oder geänderter Kundenwünsche
- Die Anforderungen bzgl. Performanz und Benutzerinteraktion waren erst in der Testphase erkennbar, als die Endanwender die Gelegenheit hatten, das System zu bedienen
- Die Sicherstellung der Zuverlässigkeit der Anwendung über manuelle Tests wurde mit zunehmender Funktionalität immer aufwendiger

Während im Wasserfall-Projekt auf eine programmierbare Commercial-of-the-Shelf-Lösung zurückgegriffen wurde, kam im agilen Projekt Open Source Software zum Einsatz. Der Ablauf erfolgte nach Maßgabe der im Agilen Manifest (Beck u. a. 2001) beschriebenen Denkweisen und war durch Praktiken des eXtreme Programming (XP) (Beck

5. Ergebnisse

Abbildung 5.3. Agile Softwareentwicklung und Wasserfallmodell im Vergleich



1999) gekennzeichnet. Der wesentliche Unterschied in der Herangehensweise bestand darin, die für klassische Vorgehensmodelle typischen Phasen der Anforderungserhebung, des technischen Entwurfs, der Implementierung sowie Inbetriebnahme und Stabilisierung auf kurze Zeiträume zu komprimieren (Abbildung 5.3). Diese als *Sprints* bezeichneten Phasen dauerten jeweils sechs Wochen und wurden in iterativer Weise ausgeführt, bis alle Anforderungen abgebildet und produktivgesetzt waren. Mit dem Ziel, die Qualität der im Rahmen der beiden unterschiedlichen Vorgehensmodelle entwickelten Softwareprodukte zu beurteilen, wurde ein weiteres Review mit identischem Teilnehmerkreis und identischer Methodik durchgeführt. Im Ergebnis hat das unter Anwendung agiler Methoden entwickelte Softwareprodukt im Hinblick auf alle ausschlaggebenden Kriterien überzeugen können (Tabelle 5.3). Eine mit den Stakeholdern geführte Diskussion deutet auf sowohl organisatorische als auch technische Zusammenhänge hin, die ursächlich zu dem beobachteten Ergebnis geführt haben.

Eine Grundvoraussetzung für die Anwendung agiler Methoden war die Schaffung flacher Hierarchien mit kurzen Entscheidungswegen über die Akzeptanz eines weitgehend autonomen Teams, in welchem Softwareentwickler und Endanwender zu den wesentlichen Akteuren gehörten. Damit einhergehend war das Einverständnis unter den beteiligten Stakeholdern zu erreichen, mit der Softwareentwicklung zu beginnen, noch bevor alle Anforderungen bekannt bzw. dokumentiert sind. Darüber hinaus wurden erfolgskritische Voraussetzungen auch auf technischer Seite ausgemacht. Ein entscheidender Faktor war, das System jederzeit kostengünstig anpassen zu können. Um diese Eigenschaft herzustellen, mussten hohe Voraussetzungen im Bezug auf die Qualität des Software-Quellcodes erreicht werden. Außerdem war es wichtig, veränderte Softwarestände kurz-

5. Ergebnisse

Tabelle 5.3. Methodenvergleich anhand verschiedener Zielkriterien

<i>Zielkriterium</i>	<i>Ansatz</i>	<i>Tools</i>	<i>Beobachtung</i>
<i>Wasserfallmodell mit programmierbarer COTS-Lösung</i>			
<i>Zuverlässigkeit</i>	Manuelle Funktions- und Integrationstests	–	Hoher Testaufwand
<i>Erweiterbarkeit</i>	Programmierbare 3-Schichten-Architektur	MS Excel Add In, proprietärer BI Application Server, MS SQL + Analysis Services	Erweiterbarkeit grundsätzlich gegeben, aber erschwert durch unübersichtliche Codebasis
<i>Usability</i>	MS Excel-basierte Benutzeroberfläche	Visual Basic for Applications (VBA)	Lange Antwortzeiten beim Speichern und Lesen von Daten
<i>Produktivität</i>	Low-Code-Ansatz	Vorgefertigte und konfigurierbare Routinen in allen 3 Schichten	Anfangs hoch, dann abnehmend mit wachsendem Funktionsumfang der Codebasis. Client Rollouts erforderlich bei Änderungen der VBA-Codebasis.
<i>Agile Softwareentwicklung mit Open-Source-Komponenten</i>			
<i>Zuverlässigkeit</i>	Automatisierte Funktionstests, teil-automatisierte Integrationstests	RSpec ¹	Geringer bis mäßiger Testaufwand
<i>Erweiterbarkeit</i>	3-Schichten-Architektur mit Model-View-Controller Entwurfsmuster	Ruby on Rails (RoR) ² , Open-Source-Bibliotheken	Uneingeschränkte Erweiterbarkeit
<i>Usability</i>	Webanwendung	Bootstrap, JQuery	Keine Einschränkungen
<i>Produktivität</i>	Web Application Framework, Quellcode Management, Continuous Integration, Continuous Delivery	RoR, Git ³ , Jenkins, Selenium, Capistrano	Durchgehend hoch

¹ RSpec ist ein Werkzeug zur Unterstützung von test driven development (TDD) in der Programmiersprache Ruby, <https://rspec.info>

² Ruby on Rails ist ein Framework zur Entwicklung von Webanwendungen, welches das Model View Controller (MVC) Entwurfsmuster implementiert, <https://rubyonrails.org>

³ Git ist ein Werkzeug zur Unterstützung der verteilten Versionsverwaltung von Software-Quellcode, <https://git-scm.com>

5. Ergebnisse

fristig produktivsetzen zu können. Dazu wurde in die Automatisierung von Tests und Deploymentvorgängen investiert und entsprechende Werkzeuge genutzt.

Zusammenfassend war die Erfahrung entstanden, dass agile Vorgehensmodelle in einer Umgebung mit unsicheren, d.h. schwierig erhebbaren oder veränderlichen Anforderungen an die Softwareentwicklung zu erkennbaren Vorteilen gegenüber klassischen Methoden führen können. Dieser Zusammenhang wird in Abbildung 5.3 illustriert. Die Schaffung geeigneter organisatorischer und technischer Rahmenbedingungen ist dabei ein kritischer Erfolgsfaktor.

Die kontinuierliche Auslieferung neuer Softwareversionen in kurzen Zeitabschnitten stellt den IT-Betrieb vor neue Aufgaben. Unter dem Begriff *DevOps* werden Methoden zusammengefasst, die eine Kultur integraler Zusammenarbeit der Bereiche Softwareentwicklung (Dev) und IT-Betrieb (Ops) fördern (Bass, Weber und Zhu 2015). Die Verbindung manifestiert sich vor allem durch Praktiken wie *Continuous Delivery* und die gemeinsame Nutzung entsprechender Werkzeuge (Ebert, Gallardo u. a. 2016). DevOps zielt darauf ab, die Softwarequalität sowie das Tempo der Auslieferung neuer Softwarestände deutlich zu steigern. Während des Betriebs auftretende Ereignisse wie Fehler, Performanceprobleme oder Supportanfragen der Endanwender sollen unmittelbar Berücksichtigung in der Softwareentwicklung finden. Dies hat zur Folge, dass Development- und Operationsprozesse ständig ineinandergreifen und entgegen klassischer Ansätze nicht mehr arbeitsteilig, sondern als einheitliche Zuständigkeit gebündelt werden (Thomas, Varwig u. a. 2017). DevOps kann insofern als die Fortführung agiler Prinzipien auf den IT-Betrieb zu einem ganzheitlichen Managementansatz für qualitativ hochwertige Software verstanden werden.

Beitrag #4 gewährt weitere Einblicke in Entwicklung und Betrieb der Informationssysteme, die dem PSS zugrundeliegen. Im Fokus der Betrachtung stehen fünf einzelne Anwendungen, die zur Ablösung von Legacysoftware bzw. zur Schaffung spezifischer Funktionalitäten entwickelt worden waren. Mit dem Ziel, eine dauerhaft hohe Qualität und Adaptionsfähigkeit der Informationssysteme zu erreichen, wurden monolithische Strukturen zu Gunsten einer *Microservices Architecture (MSA)* aufgegeben. Microservices sind kleine Applikationen, die unabhängig voneinander entwickelt, getestet, inbetriebgesetzt und skaliert werden können (Newman 2021, Thönes 2015). Unter dieser Maßgabe sind die folgenden Anwendungen entstanden:

- *Conditions* bietet Funktionen zur Pflege der mit dem PSS behafteten, kundenindividuellen Konditionen. Das System dient dem ERP-System als Datenquelle zur Erstellung von Fakturen.
- *Listings* erlaubt die Verwaltung kundenindividueller Listungen. Eine Listung bezeichnet die Aufnahme eines Produkts in das Sortiment eines Händlers.
- *Logistics* sammelt und integriert logistische Informationen wie Besuchsrhythmen, Tourenpläne, Lieferavisierungen sowie Trackinginformationen zu Sendungen.
- *PIM steht* für Product Information Management und bietet Funktionen zur Verwaltung sämtlicher Produktdaten im Hinblick auf die Regale sowie die darin be-

5. Ergebnisse

findlichen Verkaufsprodukte.

- *Specifications* ermöglicht die Verwaltung von Produktspezifikationen, um regulatorischen Anforderungen zu genügen

Nach dem Prinzip *you build it, you run it* (O’Hanlon 2006) wurde die Verantwortung für Entwicklung und Betrieb für jede einzelne Applikation in einem dedizierten Team gebündelt. Die Teams sind cross-funktional besetzt und bestehen aus ausgesuchten Endanwendern (Key User), Softwareentwicklern und Systemadministratoren. Entwickler stehen somit in Kontakt mit dem täglichen Betrieb ihrer Software und ihren Kunden. Die daraus resultierenden Feedbackschleifen gelten nach der DevOps-Bewegung als Mittel, um die Softwarequalität zu erhöhen. Tabelle 5.4 gibt Aufschluss über die Innovationsrate, mit welcher Entwicklung und Betrieb der in der Fallstudie betrachteten IT-Dienste über mehrjährige Zeiträume von drei bis fünf Jahren stattgefunden haben. Im Ergebnis wird deutlich, dass mit zwischen 6 bis zu 40 Deployments pro Monat eine kontinuierliche Entwicklung gegeben ist. Die Entwicklungsschritte umfassen sowohl technische Wartung und Upgrades als auch das Hinzufügen neuer Funktionalität und Verbesserungen der Codestruktur durch Refactoring. Die Erreichung einer hohen Testabdeckung und Testautomatisierung sind wesentliche Maßnahmen, um die Zuverlässigkeit der Anwendungslandschaft bei derart hohen Änderungsraten sicherzustellen.

Jeder IT-Dienst verfügt über eine eigene Datenhaltung und kann unabhängig von den anderen Services getestet und produktivgesetzt werden. Der Austausch von Informationen zwischen den Diensten wird ausschließlich über *Application Programming Interfaces (API)* (Zdun, Wittern und Leitner 2020) nach dem Prinzip der losen Kopplung umgesetzt (Hasselbring 2016, Shadija, Rezai und Hill 2017). Die technische Realisierung erfolgt leichtgewichtig unter Verwendung von Webservices nach REST (Representational State Transfer, Richards 2006) sowie über eine *Message Oriented Middleware (MOM)*, die das *Advanced Messaging Queuing Protocol (AMQP)* implementiert. Der Zugriff der Endanwender auf die Systeme erfolgt über ein einheitliche Weboberfläche auf Basis einer Portaltechnologie. Die Einzelheiten der modularen Architektur im Backend ist für den Endanwender somit nicht wahrnehmbar. Zur Modellierung der Verkaufsregale wird eine Standardsoftware (Shelf Modeller) eingesetzt, die mit einer eigenen Benutzeroberfläche ausgestattet ist. Abbildung 5.4 liefert einen Überblick über die Applikationslandschaft.

Im Ergebnis wird erkennbar, dass sich die nach MSA organisierten Dienste ergänzend zu Standardsoftware in die Systemlandschaft einfügen und zuverlässig betreiben lassen. Sie bieten fachspezifische Funktionalität, und integrieren Daten aus unterschiedlichen Quellen. Sie sind gekennzeichnet durch eine vergleichsweise hohe Änderungsrate und ermöglichen somit stetige Anpassung an fachliche oder technische Anforderungen. Zur Gewährleistung von Informationssicherheit werden bei MSA besondere Vorkehrungen benötigt (Pereira-Vale u. a. 2021, Nehme u. a. 2019 sowie Yarygina und Bagge 2018), die den Rahmen dieser Arbeit übersteigen und daher nicht weiter behandelt werden.

Die Entstehung und Anwendung von DevOps-Praktiken ist vorrangig im Umfeld von großen Technologie-Unternehmen bekannt. Zu den Vorreitern zählen u.a. Google, Amazon, Netflix und LinkedIn (Johnston 2020, Díaz u. a. 2019). Bemerkenswert ist, dass sich

5. Ergebnisse

Abbildung 5.4. Durch DevOps-Praktiken geprägte IT-Architektur

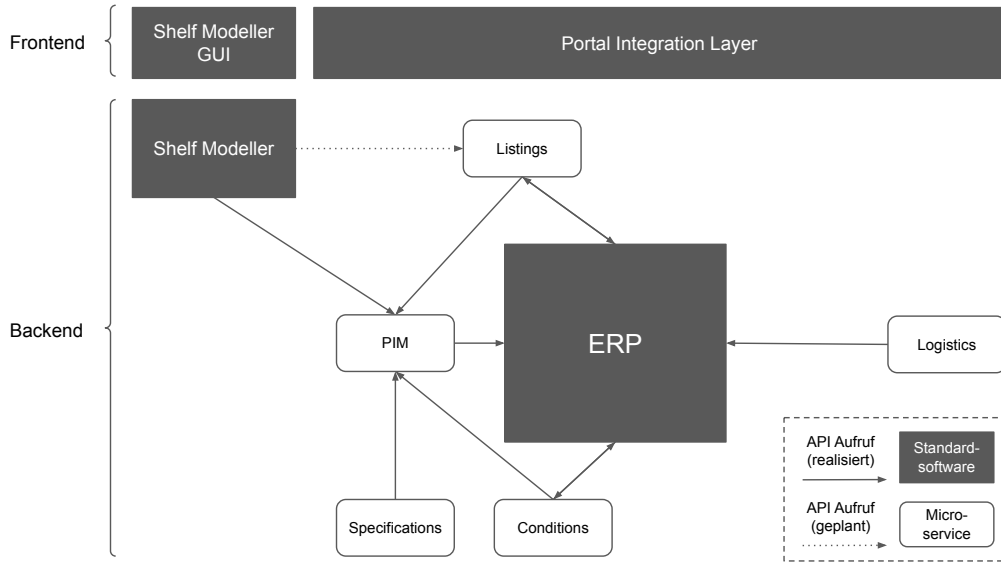
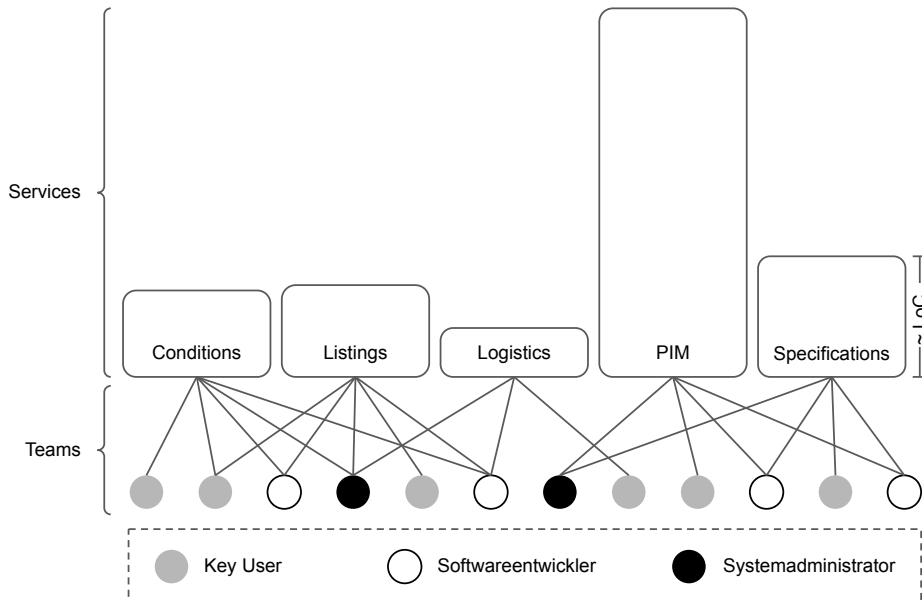


Abbildung 5.5. Zuordnung von DevOps-Teams zu Services



5. Ergebnisse

DevOps-Praktiken auch in Umgebungen mit deutlich begrenzteren Ressourcen erfolgreich umsetzen lassen. Die in der Fallstudie betrachteten Services wurden zum Stichtag der Datenerhebung von fünf Teams betreut, die aus insgesamt 6 Key Usern, 4 Softwareentwicklern und 2 Systemadministratoren zusammengesetzten waren. Die Mitgliedschaft einzelner Personen in mehreren Teams war übliche Praxis, sodass insgesamt eine übersichtliche Anzahl an Personen involviert war (Abbildung 5.5). Ein beachtenswerter Faktor ist, dass die verwendeten Werkzeuge und Technologien als Open Source Software quelloffen und überwiegend frei verfügbar ist (Tabelle 5.5), sodass Lizenzkosten kalkulierbar bleiben. Demgegenüber ist der Aufwand für die Sicherstellung des regelkonformen Einsatzes von Open Source Software im Sinne von Governance und Compliance zu sehen (Schoettle 2019, Helmreich und Riehle 2012, Gangadharan u. a. 2012).

Beitrag #5 behandelt die in Beitrag #3 identifizierte Aufgabenstellung zum Steuern des Bestands am POS anhand konfigurierbarer Zielparameter und unter Berücksichtigung relevanter Datenquellen. In diesem Zusammenhang bezeichnet der Begriff Smart Inventory einen Service, der kontextabhängig erbracht wird, d.h. unter Berücksichtigung von Ereignissen, die in einem definierten Bezugsrahmen stattfinden. Der praktisch gegebene Bezugsrahmen ist durch die verfügbaren Datenquellen begrenzt und soll bestmöglich ausgenutzt werden.

Im Rahmen einer explorativen Studie werden in Beitrag #5 die verfügbaren Datenquellen analysiert und auf ihre Eignung zur Prognose der Nachfrage untersucht. Neben Bestell- und Lieferdaten kommen vor allem Scannerdaten in Betracht. Scannerdaten sind Transaktionsdaten über den Absatz der verkauften Produkte, die an den Kassen von Einzelhandelsgeschäften erfasst werden (Koch und Erdemsiz 2020). Sie repräsentieren Kaufereignisse, die als Näherungswert für die Nachfrage am POS herangezogen werden können. Eine wesentliche Problemstellung zeigt sich in der Diversität der produkt- und marktbezogenen Kassendaten. Die einzelnen Zeitreihen zeigen ein breites Spektrum an Verlaufsformen, darunter stabile und instabile Formen, mit und ohne Trend, schwache und starke Saisonalität, Ausreißer und strukturelle Brüche sowie Lebenszykluseffekte, um nur einige zu nennen. Zur Lösung dieser Problemstellung wurde ein Verfahren entwickelt, welches die Eignung unterschiedlicher Prognosemodelle überprüft und für jede Zeitreihe das am besten geeignete Verfahren identifiziert.

Unter den Modellen befinden sich klassische Verfahren der Zeitreihenanalyse, heuristische Methoden sowie Verfahren des Maschinellen Lernens (ML). Für eine Stichprobe aus 100 repräsentativ ausgewählten Zeitreihen wurde gezeigt, dass in 80 Prozent der Fälle ein hinreichend genaues Prognoseverfahren automatisch selektiert werden kann. Auf dieser Basis kann unter Berücksichtigung der aktuellen Bestandswerte die Disposition der Waren automatisiert ausgeführt werden. Aktuelle Bestandswerte können entweder per Inventur erhoben und direkt übertragen oder als Ergebnis von Zu- und Abgängen für hinreichend kurze Zeiträume angenähert werden. Der gesamte Prozess lässt sich über eine Simulation veranschaulichen (Abbildung 5.6). Die Unsicherheiten in der Vorhersage für die übrigen 20 Prozent können durch die Konfiguration von Sicherheitsbeständen eingedämmt werden. Die Gegenüberstellung von Lagerkosten und Kosten für entgangene Umsätze aufgrund von Bestandslücken wird über ein Dashboard für die Entschei-

5. Ergebnisse

Tabelle 5.4. Kennzahlen zur Dynamik der Anwendungsentwicklung

IT-Dienst	Entwicklungsbeginn	Codezeilen (LoC) ¹	Testabdeckung (%) ¹	Anzahl Deployments ²	Ø Anzahl Deployments pro Monat
Conditions	11/2011	1293	95,20	447	7,2
Listings	11/2013	1371	89,27	633	16,6
Logistics	11/2011	733	94,48	397	6,4
PIM	04/2012	5507	96,97	2339	40,3
Specifications	09/2014	1801	85,53	605	21,6

¹ Zum Stichtag einschließlich 01/2017

² Im Zeitraum seit jeweiligem Entwicklungsbeginn bis einschließlich 01.2017

Tabelle 5.5. Werkzeuge zur Anwendung von DevOps-Praktiken

Werkzeug	DevOps-Phase	Werkzeugtyp	Konfiguration	Sprache	Lizenz
Ruby	Development	Sprache		C	2-clause BSD
Ruby on Rails	Development	Framework	YAML	Ruby	MIT
Git	Development	Continuous Integration, Versioning	Kommandozeilen Parameter, Konfigurationsdateien	U.a. C/C++, Unix Shell	GNU GPL v2
RSpec	Development	Test Automation	Ruby-basierte DSL	Ruby	MIT
Rake	Build	Build	Ruby	Ruby	MIT
Drone CI / Woodpecker CI	Build	Continuous Integration / Testing	UI	Java	Proprietär / Apache 2.0
Chef	Deployment	Configuration Management	Ruby-basierte DSL	Ruby, Erlang	Apache 2.0
Puppet	Deployment	Configuration Management	YAML	Ruby, C/C++, Closure	Apache 2.0
Capistrano	Deployment	Deployment / Configuration Management	Chef	Ruby	MIT
Docker Engine / Docker Business	Operations	Sandboxing, Container Virtualisierung		Go	Apache 2.0 / Proprietär
Errbit	Operations	Error Tracking			MIT
Icinga	Operations	Monitoring	UI	C	GNU GPL v2

dungsträger visualisiert (Abbildung 5.7). Damit werden die Stellhebel sichtbar, die das Geschehen am POS beeinflussen. Auf Basis der verfügbaren Datenquellen wird somit ein Beitrag zur Entscheidungsunterstützung geleistet. Die Disposition der Waren wird ereignisgesteuert optimiert und die Reaktionsfähigkeit insgesamt erhöht.

In Anlehnung an das MSA-Paradigma wurde die Applikation als Microservice realisiert und in die Anwendungslandschaft eingefügt. Zur Bedienung der Anwendung wurden gerätespezifische GUI entwickelt (insb. für PC und mobile Endgeräte), die auf eine einheitliche API zum Anruf von Daten sowie zum Aktivieren von Funktionen zugreifen. Die Implementierung der Datenverarbeitungsroutinen, insbesondere ETL-Prozesse sowie die Generierung modellgestützter Forecasts, erfolgte auf Basis von Python-Bibliotheken.

5.2. Implikationen

Die vorliegenden Ergebnisse erweitern die Wissensbasis für die Entwicklung und Gestaltung von PSS in der Lebensmittelbranche. Die im Rahmen der Behandlung von FF1 erbrachten Ergebnisse beschreiben Ausgangsbedingungen, Zielsetzungen, Nebenbedingungen, Akteure und Informationssysteme, die im Kontext des untersuchten Anwendungsszenarios eine Rolle spielen. Der zu FF1b entwickelte Ordnungsrahmen bildet eine Orientierungshilfe für die Gestaltung individualisierbarer Leistungsbeziehungen für PSS in B2B-Märkten. Wesentliche Merkmale des Ordnungsrahmens liegen in der Verknüpfung konfigurierbarer Produkt-Service-Bündel mit übergeordneten strategischen Zielen. Darüber hinaus werden technologische Entwicklungen sowie marktseitige Entwicklungen einbezogen. Beitrag #3 belegt die praktische Nützlichkeit dieses Konzepts mithilfe einer Fallstudie, in welcher der Ordnungsrahmen für den Lebensmitteleinzelhandel ausgeprägt und angewandt wird. In diesem Kontext wird mit *Smart Inventory* eine Prozessinnovation identifiziert. Sie umfasst die kundenindividuelle, ereignisgesteuerte Ausführung von Dispositionsvorgängen unter Berücksichtigung aktueller Daten und unter Anwendung einschlägiger Verfahren des Maschinellen Lernens. Im Rahmen der Behandlung von FF3 demonstriert Beitrag #5 über eine Softwareimplementierung, welche Potenziale und Grenzen mit diesem Konzept im konkreten Anwendungsfall verbunden sind.

Diese Arbeit veranschaulicht Ansätze zur Digitalisierung von Serviceleistungen aus der Perspektive eines Lebensmittelproduzenten. Dabei werden Einblicke in die Anwendung methodischer Konzepte sowie in die Gestaltung inhaltlich-funktionaler Lösungen gewährt. Es wird gezeigt, wie durch die Implementierung von intelligenten Diensten ein Beitrag zum Ausbau einer servicebasierten Differenzierungsstrategie geleistet werden kann. Damit einhergehend folgt die Erkenntnis, dass produktbezogene Dienstleistungen im konkreten Anwendungsfall als strategisches Differenzierungsmerkmal eingeordnet werden können. In diesem Kontext werden Smart Services spezifische Eigenschaften zugeschrieben, u.a. automatisiert, ereignisgesteuert, bedarfsgerecht und individualisierbar, optimierbar mit konfigurierbaren Zielparametern, messbar und visualisiert, flexibel anpassbar und interoperabel. Es wird unmittelbar klar, dass der Service ohne individuelle Software in dieser Qualität nicht geleistet werden kann. Software wird somit zu einem wesentlichen Teil der Dienstleistungserbringung.

5. Ergebnisse

Abbildung 5.6. Was-wäre-wenn-Simulation für definierbare Parameter

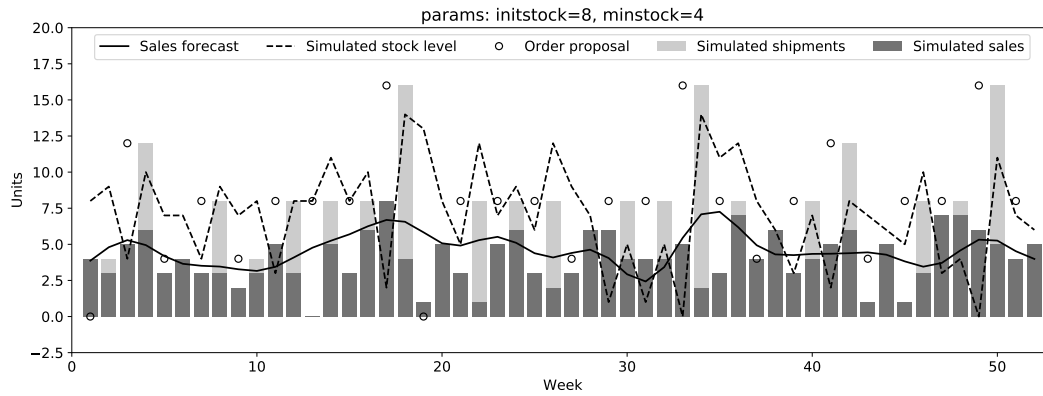
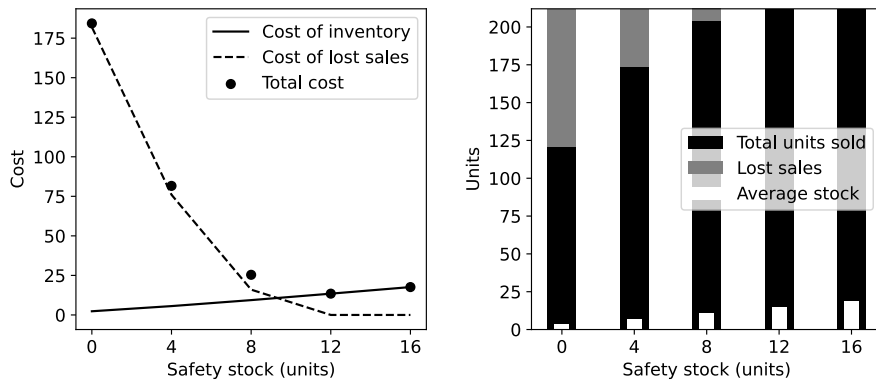


Abbildung 5.7. KPI-Simulation für verschiedene Konfigurationen des Sicherheitsbestands



5. Ergebnisse

Im Zuge der technischen Umsetzung wird die erfolgskritische Rolle der Softwareentwicklung deutlich. Es entsteht ein Bedarf für die zielgerichtete Anpassung von Informationssystemen bzw. für Individualentwicklungen. Agile Softwareentwicklung, DevOps und MSA bieten die Chance, die Anpassungsfähigkeit der IT-Landschaft zu steigern und somit einen wirksamen Beitrag zur Innovationsfähigkeit des Unternehmens zu leisten. Bemerkenswert ist, dass die Anwendung derartiger Paradigmen auch jenseits von Großkonzernen im mittelständisch geprägten Umfeld mit begrenzten Ressourcen möglich ist. Um dafür die Voraussetzungen zu schaffen, muss sich auch die interne Unternehmens-IT weiterentwickeln und vom Enabler zum Treiber von Innovationen werden. Wesentliche Bausteine dafür liegen in der Förderung einer offenen Einstellung zur Adaption neuer Technologien und Methoden, im verstärkten Einsatz cross-funktionaler Teams sowie in der Steigerung der Attraktivität als Arbeitgeber für talentierte IT-Fachkräfte im Allgemeinen.

Die im Rahmen dieser Arbeit aufgezeigte Verknüpfung servicebasierter Differenzierungsstrategien über Servicemodelle und Softwareentwicklung bis hin zum IT-Betrieb zeigt die besondere, strategische Rolle der Wirtschaftsinformatik in der unternehmerischen Praxis sehr deutlich. Die daraus gewonnenen Erkenntnisse sprechen für eine unmittelbare Berücksichtigung von einschlägigem Know How aus den Gebieten Service Engineering und Software Engineering bei strategischen Entscheidungen und deren Operationalisierung. Diese Erkenntnis steht im Einklang zur weiterhin steigenden Bedeutung von Informationstechnologien. IT wandelt sich branchenübergreifend von einem unterstützenden Faktor immer mehr zu einem steuernden Faktor (Brassel und Gadatsch 2018). Der Einfluss von IT auf den geschäftlichen Erfolg verstärkt sich fortwährend. In diesem Zuge werden verschiedene Berufsbilder diskutiert, die sich wandeln bzw. neu entstehen. Zu den prominentesten zählen Chief Information Officer (CIO), Chief Process Officer (CPO) und Chief Digital Officer (CDO) (Hess und Sciuk 2022; Walchshofer und Riedl 2017). Vor diesem Hintergrund zeigen die Ergebnisse dieser Arbeit ein weiteres Beispiel für die Wichtigkeit, digitales Know How konsequent bei der Entwicklung neuartiger Leistungen bzw. Geschäftsmodelle einfließen zu lassen.

Aus wissenschaftlicher Perspektive ergibt sich die weiterführende Forschungsfrage, inwieweit der Ordnungsrahmen für andere Branchen ausgeprägt und angewandt werden kann und ggfs. welche weiteren Eigenschaften intelligente Dienste erfüllen müssen. Aus praktischer Perspektive ist auffällig, dass im Rahmen der Fallstudie nur wenige Händler dazu bereit waren, Scannerdaten zur Verfügung zu stellen bzw. diese über eine maschinenlesbare Schnittstelle anzubieten. Folglich müssen Kollaborationsmodelle gefunden werden, die die legitimen Interessen des Lebensmitteleinzelhandels für den Schutz eigener Daten ausbalancieren mit den Potenzialen, intelligente Services in die Wertschöpfungskette zu integrieren, die von Partnern erbracht werden.

5.3. Limitationen

Die in dieser Arbeit behandelten Themenfelder sind verschiedenen Forschungsgebieten zuzuordnen und haben daher einen fachübergreifenden Charakter. Diese Einsicht steht

im Einklang mit dem Verständnis der Wirtschaftsinformatik als multidisziplinäre Forschung. Die Anwendung von Methoden aus den Bereichen Service und Software Engineering wird im Rahmen einer soziotechnischen Würdigung in der Art und Weise vorgenommen, dass keine isolierte Betrachtung methodischer oder technischer Aspekte erfolgt, sondern deren Nützlichkeit im Bezug auf die Erreichung von Unternehmenszielen im Vordergrund steht. Insoweit wird durchgehend eine ganzheitliche und multidisziplinäre Betrachtungsweise zu Grunde gelegt. Darauf aufbauend stützt sich die Relevanz der vorliegenden Arbeit sowohl auf theoretische Erkenntnisziele als auch auf praktische Gestaltungsziele. Erstere liegen in der kritischen Reflexion darüber, wie gut bzw. unter welchen Bedingungen Methoden aus den Gebieten Service und Software Engineering ihren Zweck erfüllen. Letztere liegen in der Anwendung ebendieser Methoden mit dem Ziel, konkretes Gestaltungswissen für die Praxis zu erarbeiten.

Der Autor dieser Arbeit war in einem führenden Unternehmen der Nahrungsmittelbranche als Leiter IT und später als CIO für die strategische Ausrichtung und den operativen Betrieb der IT zur Unterstützung spezifischer Geschäftsmodelle verantwortlich. Diese Arbeit repräsentiert damit auch die Erfahrungen, die während der 10-jährigen Tätigkeit des Autors in dieser Rolle entstanden sind. Die Validierung der entwickelten Konzepte und Lösungen ist ausschließlich im Zuge von Fallstudien aus der Perspektive eines einzelnen Unternehmens erfolgt. Durch den unmittelbaren Anwendungsbezug lässt sich die praktische Relevanz und Nützlichkeit der Ergebnisse untermauern. In gleichem Zuge wird deren Generalisierbarkeit auf andere Unternehmen oder Branchen eingeschränkt.

6. Zusammenfassung

Die Schwerpunkte dieser Arbeit liegen in der Erlangung von Gestaltungswissen für PSS in der Lebensmittelindustrie sowie in der Gewinnung methodischer Erkenntnisse für Entwicklung und Betrieb zugrunde liegender Informationssysteme. Der Forschungsprozess folgt dem Design-Science-Paradigma mit dem Ziel, Erkenntnisse durch die Beobachtung von Problemstellungen im Kontext realer Anwendungsfälle sowie durch die Konstruktion und Evaluation geeigneter Lösungsansätze zu erlangen. Zentraler Untersuchungsgegenstand ist die Leistungsbeziehung eines Anbieters von Lebensmittelprodukten gegenüber dem Einzelhandel. Im Mittelpunkt steht die Bereitstellung von Verkaufsflächen seitens des Handels und der damit verbundene Auftrag an den Hersteller, die Flächen möglichst erfolgreich zu bewirtschaften.

Zur Modellierung der Leistungsbeziehungen zwischen Hersteller und Handel wird ein allgemeiner Ordnungsrahmen konstruiert und individuell ausgeprägt. Im Ergebnis entsteht ein Modell, welches die Entwicklung und Konfiguration von Services zur Erreichung übergeordneter, strategischer Ziele beschreibt. Darüber hinaus werden Technologien identifiziert, auf deren Basis Serviceinnovationen umgesetzt werden können. In methodischer Hinsicht wird demonstriert, inwieweit agile Methoden der Softwareentwicklung in Kombination mit DevOps-Praktiken für die Gestaltung und den Betrieb

6. Zusammenfassung

der zugrunde liegenden Anwendungslandschaft geeignet sind. Dabei gilt die Maßgabe, die mit dem Betrieb des PSS einhergehenden Anforderungen in Software abzubilden und dauerhaft Prozessinnovationen zu ermöglichen. Um dies zu erreichen, ist eine hohe Anpassungsfähigkeit der Systeme herzustellen und nachhaltig aufrechtzuerhalten. Unter Berücksichtigung der gewonnenen Erkenntnisse wird eine Softwareanwendung entwickelt, die sich in die bestehende Anwendungslandschaft einfügt und eine Prozessinnovation realisiert.

Literatur

- Bass, L., I. Weber und L. Zhu (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
- Baumgartner, T. und M. Schölling (2000). „Die Konditionenspreizung der Hersteller gegenüber dem Handel – gesehen als Principal-Agent-Problem“. In: *der markt – international Journal of Marketing* 39.1, S. 3–17.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Publishing Company.
- Beck, K. u. a. (2001). *Manifesto for Agile Software Development*. URL: <http://agilemanifesto.org> (besucht am 05.07.2022).
- Becker, J., R. Holten u. a. (2003). *Forschungsmethodische Positionierung in der Wirtschaftsinformatik: Epistemologische, ontologische und linguistische Leitfragen*. Arbeitsberichte des Instituts für Wirtschaftsinformatik 93. Münster.
- (2004). „Epistemologische Positionierungen in der Wirtschaftsinformatik am Beispiel einer konsensorientierten Informationsmodellierung“. In: *U. Frank (Hrsg.): Wissenschaftstheorie in Ökonomie und Wirtschaftsinformatik – Theoriebildung und -bewertung, Ontologien, Wissensmanagement* 1, S. 335–366.
- Becker, J. und D. Pfeiffer (2006). „Beziehungen zwischen behavioristischer und konstruktionsorientierter Forschung in der Wirtschaftsinformatik“. In: *Fortschritt in den Wirtschaftswissenschaften – Wissenschaftstheoretische Grundlagen und exemplarische Anwendungen*. Zelewski, S. und Akca, N., S. 1–17.
- Becker, J., B. Niehaves u. a. (2009). „Forschungsmethodik einer Integrationsdisziplin – Eine Fortführung und Ergänzung zu Lutz Heinrichs „Beitrag zur Geschichte der Wirtschaftsinformatik“ aus gestaltungsorientierter Perspektive“. In: *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik*. Hrsg. von J. Becker, H. Krcmar und B. Niehaves. Heidelberg: Physica-Verlag HD, S. 1–22.
- Becker, J., M. Rosemann und R. Schütte (1995). „Grundsätze ordnungsmäßiger Modellierung“. In: *Wirtschaftsinformatik* 37, S. 435–445.
- Becker, M. und S. Klingner (2017). „Konzepte zur kundenspezifischen Anpassung von Dienstleistungen“. In: *Smart Service Engineering: Konzepte und Anwendungsszenarien für die digitale Transformation*. Hrsg. von O. Thomas, M. Nüttgens und M. Fellmann. Wiesbaden: Springer Fachmedien, S. 2–28.
- Bendl, H. (2000). „Planung, Steuerung und Kontrolle leistungsbezogener Konditionen: eine empirische Analyse zwischen Industrie und Lebensmittelhandel“.
- Blinn, N. u. a. (2010). „Lebenszyklusmodelle hybrider Wertschöpfung: Modellimplikationen und Fallstudie“. In: *Hybride Wertschöpfung: Mobile Anwendungssysteme für effiziente Dienstleistungsprozesse im technischen Kundendienst*. Hrsg. von O. Thomas, P. Loos und M. Nüttgens. Berlin, Heidelberg: Springer, S. 130–143.

Literatur

- Böhm, T., J. M. Leimeister und K. Moeslein (2014). „Service Systems Engineering“. In: *Business & Information Systems Engineering* 6, S. 73–79.
- Bosch, J. (2016). „Speed, Data, and Ecosystems: The Future of Software Engineering“. In: *IEEE Software* 33, S. 82–88.
- Brassel, S. und A. Gadatsch (2018). „IT-Management in turbulenten Zeiten: vom IT-Leiter über den CIO zum CDO“. In: *Nachhaltiges Wirtschaften im digitalen Zeitalter: Innovation – Steuerung – Compliance*. Hrsg. von A. Gadatsch u. a. Wiesbaden: Springer Fachmedien, S. 121–134.
- Broy, M. u. a. (2006). „Manifest*: Strategische Bedeutung des Software Engineering in Deutschland“. In: *Informatik-Spektrum* 29, S. 210–221.
- Bullinger, H.-J., W. Ganz und J. Neuhüttler (2017). „Smart Services – Chancen und Herausforderungen digitalisierter Dienstleistungssysteme für Unternehmen“. In: *Dienstleistungen 4.0: Konzepte – Methoden – Instrumente. Band 1. Forum Dienstleistungsmanagement*. Hrsg. von M. Bruhn und K. Hadwich. Wiesbaden: Springer Fachmedien, S. 97–120.
- Díaz, J. u. a. (2019). „DevOps in Practice – A Preliminary Analysis of Two Multinational Companies“. In: *Product-Focused Software Process Improvement*. Hrsg. von X. Franch, T. Männistö und S. Martínez-Fernández. Springer International Publishing, S. 323–330.
- Eberhard, K. (1999). *Einführung in die Erkenntnis- und Wissenschaftstheorie – Geschichte und Praxis der konkurrierenden Erkenntniswege*. 3. Aufl. Kohlhammer.
- Ebert, C. und J. Favaro (2017). „Automotive Software“. In: *IEEE Software* 34.3, S. 33–39.
- Ebert, C., G. Gallardo u. a. (2016). „DevOps“. In: *IEEE Software* 33.3, S. 94–100.
- Gangadharan, G. u. a. (2012). „Managing license compliance in free and open source software development“. In: *Information Systems Frontiers* 14.2, S. 143–154.
- Goedkoop, M. u. a. (1999). *Product Service systems, Ecological and Economic Basics*. Techn. Ber. Amersfoort, The Netherlands.
- Gregor, S. (2002). „Design Theory in Information Systems“. In: *Australasian Journal of Information Systems* 10.1.
- Gregor, S. und A. Hevner (2013). „Positioning and Presenting Design Science Research for Maximum Impact“. In: *MIS Quarterly* 37, S. 337–356.
- Hasselbring, W. (2016). „Microservices for scalability: Keynote talk abstract“. In: *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering*, S. 133–134.
- Helmreich, M. und D. Riehle (2012). „Geschäftsrisiken und Governance von Open Source in Softwareprodukten“. In: *HMD – Praxis der Wirtschaftsinformatik* 49.1, S. 17–25.
- Hess, T. und C. Sciuk (2022). „Das Zusammenspiel von Chief Executive Officer und Chief Digital Officer bei der digitalen Transformation“. In: *HMD – Praxis der Wirtschaftsinformatik* 59.3, S. 840–853.
- Hevner, A. (2007). „A Three Cycle View of Design Science Research“. In: *Scandinavian Journal of Information Systems* 19.

Literatur

- Hevner, A. u. a. (2004). „Design Science in Information Systems Research“. In: *Management Information Systems Quarterly* 28, S. 75–106.
- Johnston, C. (2020). „DevOps Infrastructure“. In: *Advanced Platform Development with Kubernetes*. Springer, S. 33–69.
- Jones und Gregor (2007). „The Anatomy of a Design Theory“. In: *Journal of the Association for Information Systems* 8, S. 19.
- Koch, J. und B. Erdemsiz (2020). „Einsatz von Scannerdaten während der COVID-19-Pandemie“. In: *WISTA – Wirtschaft und Statistik* 72.4, S. 96–106.
- Lademann, R. (2012). „Die ökonomischen Aspekte der Preisempfehlung im Lichte von Besonderheiten der Ernährungswirtschaft“. In: *Vertikale Preis- und Markenpflege im Kreuzfeuer des Kartellrechts*. Springer, S. 233–242.
- (2015). „Zur Nachfragemacht von Unternehmen des Lebensmitteleinzelhandels Über den Einfluss von Unternehmensgröße im Handel auf die Einkaufskonditionen“. In: *WUW: Wirtschaft und Wettbewerb* 65.7, S. 716–732.
- Myers, M. D. (2009). *Qualitative Research in Business & Management*. London: Sage Publications Ltd.
- Nehme, A. u. a. (2019). „Securing Microservices“. In: *IT Professional* 21.1, S. 42–49.
- Newman, S. (2021). *Building Microservices – Designing Fine-Grained Systems*. California: O’Reilly Media, Inc.
- O’Hanlon, C. (2006). „A Conversation with Werner Vogels: Learning from the Amazon Technology Platform“. In: *ACM Queue* 4.4, S. 14–22.
- Oates, B. J. (2005). *Researching Information Systems and Computing*. Sage Publications Inc.
- Österle, H. u. a. (2010). „Memorandum zur gestaltungsorientierten Wirtschaftsinformatik“. In: *Zeitschrift für betriebswirtschaftliche Forschung* 6.62, S. 664–672.
- Oswald, G. und H. Krcmar, Hrsg. (2018). *Digitale Transformation: Fallbeispiele und Branchenanalysen*. Springer Nature.
- Peppers, K., T. Tuunanen, M. A. Rothenberger u. a. (2008). „A Design Science Research Methodology for Information Systems Research“. In: *Journal of Management Information Systems* 24, S. 45–77.
- Peppers, K., T. Tuunanen und B. Niehaves (2018). „Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research“. In: *European Journal of Information Systems* 27.2, S. 129–139.
- Pereira-Vale, A. u. a. (2021). „Security in microservice-based systems: A Multivocal literature review“. In: *Computers & Security* 103, S. 102200.
- Richards, R. (2006). „Representational State Transfer (REST)“. In: *Pro PHP XML and Web Services*. Berkeley, CA: Apress, S. 633–672.
- Riedl, R. (2006). „Erkenntnisfortschritt durch Forschungsfallstudien“. In: *Fortschritt in den Wirtschaftswissenschaften: Wissenschaftstheoretische Grundlagen und exemplarische Anwendungen*. Hrsg. von S. Zelewski und N. Akca. Wiesbaden: DUV, S. 113–145.
- Robra-Bissantz, S. und S. Strahringer (2020). „Wirtschaftsinformatik-Forschung für die Praxis“. In: *HMD – Praxis der Wirtschaftsinformatik* 57, S. 162–188.

Literatur

- Schlippenbach, V. von und F. Pavel (2011). „Konzentration im Lebensmitteleinzelhandel: Hersteller sitzen am kürzeren Hebel“. In: *DIW Wochenbericht* 78.13, S. 2–9.
- Schoettle, H. (2019). „Open Source License Compliance – Why and How?“. In: *Computer* 52.08, S. 63–67.
- Shadija, D., M. Rezai und R. Hill (2017). „Towards an Understanding of Microservices“. In: *23rd International Conference on Automation and Computing (ICAC)*. IEEE, S. 1–6.
- Streibich, K.-H. (2018). „Die Herausforderung der digitalen Transformation meistern“. In: *Digitalisierung im Spannungsfeld von Politik, Wirtschaft, Wissenschaft und Recht: 1. Band: Politik und Wirtschaft*. Hrsg. von C. Bär, T. Grädler und R. Mayr. Berlin, Heidelberg: Springer, S. 521–528.
- Thomas, O. (2006). *Management von Referenzmodellen – Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen*. Berlin: Logos Verlag.
- Thomas, O., P. Loos und M. Nüttgens (2010). *Hybride Wertschöpfung – Mobile Anwendungssysteme für effiziente Dienstleistungsprozesse im technischen Kundendienst*. Berlin, Heidelberg: Springer.
- Thomas, O., A. Varwig u. a. (2017). „DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter“. In: *HMD – Praxis der Wirtschaftsinformatik* 54.2, S. 178–188.
- Thomas, O., M. Nüttgens und M. Fellmann (2017). *Smart Service Engineering – Konzepte und Anwendungsszenarien für die digitale Transformation*. Wiesbaden: Springer Gabler.
- Thomas, O. und A.-W. Scheer (2003). „Customizing von Dienstleistungsinformationssystemen“. In: *Service Engineering – Entwicklung und Gestaltung innovativer Dienstleistungen*. Hrsg. von H.-J. Bullinger und A.-W. Scheer. Berlin, Heidelberg: Springer, S. 677–718.
- Thönes, J. (2015). „Microservices“. In: *IEEE Software* 32.1, S. 116–116.
- Tukker, A. und U. Tischner (2006). „Product-services as a research field: past, present and future. Reflections from a decade of research“. In: *Journal of cleaner production* 14.17, S. 1552–1556.
- Vaishnavi, V. und W. Kuechler (2015). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. CRC Press, Taylor & Francis Group.
- Vyatkin, V. (2013). „Software Engineering in Industrial Automation: State-of-the-Art Review“. In: *IEEE Transactions on Industrial Informatics* 9.3, S. 1234–1249.
- Walchshofer, M. und R. Riedl (2017). „Der Chief Digital Officer (CDO): Eine empirische Untersuchung“. In: *HMD – Praxis der Wirtschaftsinformatik* 54.3, S. 324–337.
- Wilde, T. und T. Hess (2006). *Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung*. Arbeitsbericht des Instituts für Wirtschaftsinformatik und Neue Medien. München.
- Yarygina, T. und A. H. Bagge (2018). „Overcoming Security Challenges in Microservice Architectures“. In: *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, S. 11–20.

Literatur

- Yin, R. (2009). *Case Study Research: Design and Methods*. 4. Aufl. California: Sage Publication Inc.
- Zdun, U., E. Wittern und P. Leitner (2020). „Emerging Trends, Challenges, and Experiences in DevOps and Microservice APIs“. In: *IEEE Software* 37.1, S. 87–91.

Teil II.

Einzelbeiträge

1. Wertorientiertes Vendor Managed Inventory durch IT-gestützte Steuerung

<i>Titel</i>	Wertorientiertes Vendor Managed Inventory durch IT-gestützte Steuerung
<i>Autoren</i>	Alexander Fuchs , Carl Stolze, Rüdiger Breitschwerdt, Sebastian Hucke, Oliver Thomas
<i>Publikationsorgan</i>	HMD – Praxis der Wirtschaftsinformatik
<i>Ranking</i>	WKWI: B / VHB: D
<i>Status</i>	Veröffentlicht
<i>Bibliographische Information</i>	Fuchs, A. , Stolze, C., Breitschwerdt, R., Hucke, S. und O. Thomas (2012), „Wertorientiertes Vendor Managed Inventory durch IT-gestützte Steuerung“. In: <i>HMD – Praxis der Wirtschaftsinformatik</i> 49.3, S. 95–103.
<i>Zusammenfassung</i>	Im Wettbewerb um die begrenzten Verkaufsflächen im Lebensmitteleinzelhandel bieten Lieferanten Vendor Managed Inventory (VMI) als komplementäre Dienstleistung zu ihren Produkten an. VMI ist ein Logistikkonzept, bei dem der Lieferant die Verantwortung für den Bestand seiner Produkte beim Kunden übernimmt. Der Lebensmitteleinzelhandel stellt mit über 30.000 Filialen deutschlandweit besondere Herausforderungen an Zulieferer, die im Rahmen von VMI die Warendisposition in den Verkaufsregalen steuern. Eine effiziente Bewältigung dieser logistischen Aufgabe kann nur durch eine eng aufeinander abgestimmte Zusammenarbeit zwischen den beteiligten Akteuren, zu denen auch Transportdienstleister und Rackjobbing-Anbieter zählen, erfolgen. Die gezielte Zusammenführung und Nutzung von Informationen zur wertorientierten Steuerung eröffnet dabei Potenziale zur Einsparung von Kosten und Steigerung der Servicequalität, wie eine Fallstudie in der NuG-Branche aufzeigt.
<i>Identifikation</i>	https://doi.org/10.1007/BF03340703
<i>URL</i>	https://link.springer.com/article/10.1007/BF03340703
<i>Copyright</i>	Springer Fachmedien Wiesbaden, 2012

2. Von der klassischen zur agilen Softwareentwicklung – Evolution der Methoden am Beispiel eines Anwendungssystems

<i>Titel</i>	Von der klassischen zur agilen Softwareentwicklung – Evolution der Methoden am Beispiel eines Anwendungssystems
<i>Autoren</i>	Alexander Fuchs , Carl Stolze, Oliver Thomas
<i>Publikationsorgan</i>	HMD – Praxis der Wirtschaftsinformatik
<i>Ranking</i>	WKWI: B / VHB: D
<i>Status</i>	Veröffentlicht
<i>Bibliographische Information</i>	Fuchs, A. , Stolze, C., Thomas, O. (2013), „Von der klassischen zur agilen Softwareentwicklung – Evolution der Methoden am Beispiel eines Anwendungssystems“. In: <i>HMD – Praxis der Wirtschaftsinformatik</i> 50.2, S. 17–26
<i>Zusammenfassung</i>	Agile Methoden in der Softwareentwicklung sind keine zufällige Erscheinung. Sie haben sich aus der Unzulänglichkeit und Untauglichkeit altbewährter Methoden heraus entwickelt, die – zumindest in bestimmten Problemstellungen – agilen Ansätzen unterlegen sind. Doch was sind die entscheidenden Faktoren, die eine Abkehr von traditionellen Ansätzen rechtfertigen? Dieser Beitrag zeigt anhand einer Fallstudie auf, wie agile Herangehensweisen in Szenarien mit unsicheren Anforderungen an die Funktionalität und Bedienbarkeit von Software konventionelle Methoden verdrängen und welche Rolle organisatorische und technische Aspekte dabei spielen.
<i>Identifikation</i>	https://doi.org/10.1007/BF03340792
<i>URL</i>	https://link.springer.com/article/10.1007/BF03340792
<i>Copyright</i>	Springer Fachmedien Wiesbaden, 2013

3. Designing Customer-Specific Product-Service Systems in B2B Markets

<i>Titel</i>	Designing Customer-Specific Product-Service Systems in B2B Markets
<i>Autoren</i>	Alexander Fuchs , Sebastian Bittmann, Deniz Özcan
<i>Publikationsorgan</i>	Enterprise Modelling and Information Systems Architectures, German Informatics Society (GI)
<i>Ranking</i>	WKWI: – / VHB: C
<i>Status</i>	Veröffentlicht
<i>Bibliographische Information</i>	Fuchs, A. , Bittmann S. und D. Özcan (2015), „ <i>Designing Customer-Specific Product-Service Systems in B2B Markets – A Consecutive Framework for Development and Configuration Management</i> “. In: <i>Enterprise Modelling and Information Systems Architectures (EMISAJ) – International Journal of Conceptual Modeling</i> 10.1, S. 109–124
<i>Zusammenfassung</i>	Suppliers of Product-Service Systems (PSS) pursue the objective of creating long-term customer relationships. In order to achieve that, PSS in Business to Business (B2B) markets need to be designed as a tool for PSS suppliers and clients, supporting them in achieving their individual business goals. One approach can be applied by identifying common goals and aligning all PSS-related management activities. This paper shows how this approach can be designed in a systematic way. Therefore, a consecutive framework for the development and configuration management of customer-specific PSS in B2B markets is derived. Applying it to a case study of food industry, the framework's relevance and its transferability into practice is demonstrated.
<i>Identifikation</i>	https://doi.org/10.18417/emisa.10.1.6
<i>URL</i>	https://emisa-journal.org/emisa/article/view/125
<i>Copyright</i>	Copyright is retained by the authors.

4. DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter

<i>Titel</i>	DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter
<i>Autoren</i>	O. Thomas, A. Varwig, F. Kammler, B. Zobel, A. Fuchs
<i>Publikationsorgan</i>	HMD – Praxis der Wirtschaftsinformatik
<i>Ranking</i>	WKWI: B / VHB: D
<i>Status</i>	Veröffentlicht
<i>Bibliographische Information</i>	Thomas, O., Varwig, A., Kammler, F., Zobel, B. und A. Fuchs (2017), „DevOps: IT-Entwicklung im Industrie 4.0-Zeitalter“. In: <i>HMD – Praxis der Wirtschaftsinformatik</i> 54.2, S. 178–188.
<i>Zusammenfassung</i>	Geringe Reaktionszeiten und hohe Flexibilität in der Informationsverarbeitung werden für Unternehmen aller Größen und Branchen immer wichtiger. Gleichzeitig steigen die zu verarbeitenden Datenmengen durch immer vielseitigere Sensoren und den Einsatz neuer Technologien. Nicht zuletzt nehmen auch Unternehmenskooperationen und die Zahl der interorganisationellen IT-Plattformen stetig zu. Dadurch haben sich die Anforderungen an moderne IT-Lösungen und -Systeme in den letzten Jahren drastisch verändert. Branchen nahezu aller Art stehen unter dem Druck, ihre Geschäftsmodelle und die unterliegenden IT-Lösungen in immer kürzeren Zyklen an die neuen Möglichkeiten anzupassen. Dies macht die Abkehr von klassischen, rigiden Softwareentwicklungsprozessen und die Umsetzung von agilen Softwareentwicklungsstrategien bereits heute zu einem festen Bestandteil vieler IT-Organisationen. DevOps-Praktiken führen das agile Prinzip auf nachgelagerte Prozesse wie Bereitstellung und Betrieb von Software fort und gelten als letzter Baustein für eine reaktionsfähige IT im digitalen Zeitalter.
<i>Identifikation</i>	https://doi.org/10.1365/s40702-017-0291-8
<i>URL</i>	https://link.springer.com/article/10.1365/s40702-017-0291-8
<i>Copyright</i>	Springer Fachmedien Wiesbaden, 2017

5. Machine Learning for Demand Forecasting – A Case Study in the German Food Retail Sector

<i>Titel</i>	Machine Learning for Demand Forecasting – A Case Study in the German Food Retail Sector
<i>Autoren</i>	A. Fuchs , Dr. A. Varwig, H. Kortum, O. Thomas
<i>Publikationsorgan</i>	Universitätsbibliothek Osnabrück
<i>Status</i>	Veröffentlicht
<i>Bibliographische Information</i>	Fuchs, A. , Varwig, A., Kortum, H. (2023): Machine Learning for Demand Forecasting – A Case Study in the German Food Retail Sector
<i>Zusammenfassung</i>	Avoiding stockouts is a constant challenge in retail. Over the last decades, various forms of cooperation and data interchange between retail and industry have been developed in order to improve stockout rates. Even so, mostly due to inaccurate demand forecasts stockouts do still occur. Forecasts drive operational decisions and have become vital in order to meet increasing customer expectations, decreasing lead times and the need to manage limited, or even perishable resources. This is even more relevant in the competitive field of food retail. In this study we analyze an implementation of Vendor Managed Inventory in the German food retail sector. We demonstrate the potential of statistical modelling and machine learning in predicting sales out the available data sources. We further show how to build a decision support tool to balance the risk of stockouts against the cost of inventory.
<i>Identifikation</i>	https://doi.org/10.48693/537

Machine Learning for Demand Forecasting

A Case Study in the German Food Retail Sector

Alexander Fuchs, Andreas Varwig, Henrik Kortum, Oliver Thomas

Abstract. Avoiding stockouts is a constant challenge in retail. Over the last decades, various forms of cooperation and data interchange between retail and industry have been developed in order to improve stockout rates. Even so, mostly due to inaccurate demand forecasts stockouts do still occur. Forecasts drive operational decisions and have become vital in order to meet increasing customer expectations, decreasing lead times and the need to manage limited, or even perishable resources. This is even more relevant in the competitive field of food retail. In this study we analyze an implementation of Vendor Managed Inventory in the German food retail sector. We demonstrate the potential of statistical modelling and machine learning in predicting sales out the available data sources. We further show how to build a decision support tool to balance the risk of stockouts against the cost of inventory.

Keywords: Demand Forecasting, Machine Learning, Data Science, Decision Support, Food Retail, Case Study

1 Introduction

In the era of digital transformation, retailers are facing constant challenges of meeting the emerging demands and expectations of customers. From online shops and food delivery concepts to mobile payment and cashierless stores, retail chains are exploring new ways of attracting customers and enriching the overall shopping experience. However, one essential issue, despite being addressed in a multitude of strategies and approaches, remains unsolved.

By measuring out-of-stock rates at an average of 8%, studies indicate that the availability of goods to customers is still one of the most relevant problems in the retail industry [9]. According to studies among German retailers, the main reasons for stockouts are imprecise sales forecasts and belated orders [7]. The term “out-of-stock” in this context is seen as an event where a product for distribution is not available for the customer. For products affected by advertising or sales promotions, even higher rates of over 10% can be observed [9]. Earlier research has already identified average out-of-stock rates about 8% [8, 23, 27].

For information interchange, EDI (Electronic Data Interchange) has been established as the defacto standard in retail. Our research is focused on the question how such data can be utilized to automate and improve demand forecasts and respective replenishment routines. The result of our work is a decision support application, helping to balance the risk of stockouts against inventory cost including spoilage. We evaluate the applicability and limitations of our approach and point out the potential of further research.

2 Related Work

Precise and seasonable sales forecasting is crucial for retail business as it directly affects stock management, sales and profitability [18]. In the food domain, large assortments of goods need to be offered to the customer, some of them are highly perishable. Promotions, price changes, weather and seasonal habits are causing variances in the consumer demand in the long run other influences like changing consumer preferences are likely to happen, adding

even more complexity. Operations research and operations management methods have played an important role in solving inventory management problems [2, 24].

One widely used approach is the newsvendor problem (NVP), for which a variety of applications have been published (e.g. [11, 17]). Numerous applications of statistical methods and machine learning techniques have been applied in the food industry to make predictions independent of specific demand curves, e.g. by quantile regression, and to connect production and supply quantity planning [12]. Repeatedly, it has been demonstrated that an effective solution to a data-driven newsvendor problem is achievable if only the database is sufficiently large. Other studies, on the other hand, emphasize that taking into account the historical demand distribution, especially in a heterogeneous data set, is crucial to identify an optimal solution to the NVP [15].

Demand forecasting techniques for retail have been extensively discussed in literature and practice. Most of the research has applied means of statistical time series analysis (ARMA, ARIMA) (e. g. [3, 20, 22, 27]) and exponential smoothing (ES) approaches (e. g. [18, 20, 26]) as well as multivariate regression models [16]. Moreover, numerous heuristic approaches based on machine learning techniques have been used like e.g. artificial neural networks (ANN) (e. g. [1, 13, 14, 19]).

3 Experimental Description

3.1 Research Design

We conduct an exploratory study following the CRISP-DM approach (figure 1). The study took as its starting point the underlying business model and respective data sources. We focus our research on practical application with the aim to leverage real data from the available sources. We compare the results of different forecasting approaches on sales data and show how to identify the best region- and product-specific forecasting procedure to be used in the replenishment process. As a result, we develop a dashboard-like tool for decision makers. A main feature lies in the simulation of the effect of input parameters on performance indicators such as stockouts and inventory cost. All implementation work for data preparation, forecasting and simulation has been done with Python and respective libraries. The application code serves as a prototype to be extended to production-ready software in further iterations. In conclusion of the study, we evaluate the results, discuss practical implications, and identify potentials for further research.

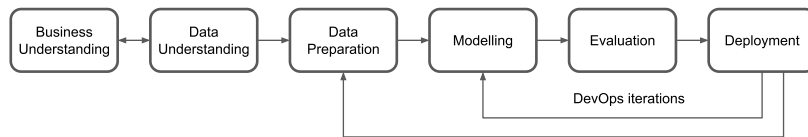


Figure 1. CRISP DM trajectory used in this research

3.2 Business and Data Understanding

Our study is part of a collaborative project in the food retail industry. We take the perspective of a food supplier that provides a assortment of several hundred articles to over 10.000 outlets in German food retail. The supplier is in charge of inventory and replenishment at the point of sale, which is referred to as vendor managed inventory (VMI). We aim to find most accurate demand forecasting procedures for each product in order to optimize

replenishment routines. In this context we understand *demand* as the desire to buy a certain quantity of a product in a specific retail store.

The vendor has access to retail customer orders and shipments from underlying ERP systems as well as checkout data from retail stores. We have decided to use actual sales figures that we draw from the point of sale as a proxy for demand. Such data is obtained when product barcodes are scanned during the checkout process. Scanner-based checkout systems are widely used, well-developed and reliable. However, there are limitations which can cause data quality issues. For example, in the case of accidental double scanning of a single product going unnoticed, or when a barcode is not readable and scanning is replaced by manual inputs. Additionally, demand modeling may become skewed if products that are actually desired cannot be purchased in a store at all. This can occur, for example, in the event of supply shortages, theft or simple misplacement. As a result, it is to be expected that scanner data may be incomplete or inaccurate.

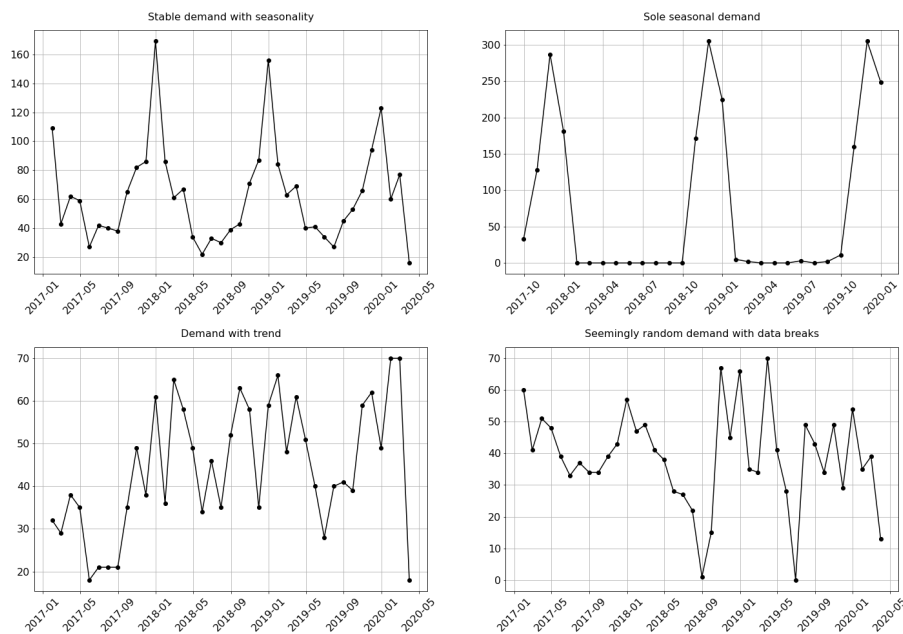


Figure 2. Sample of dataset after preparation routine

3.3 Data Preparation

The data we had access to comprises a set of daily sales figures of 10 products in 10 german retail outlets over 6 years of time (100 demand curves in total). The dataset was selected in close consultation with various domain experts from the industry partner in order to investigate high-turnover products while covering the widest possible variety of demand patterns that may occur. We examine products with stable demand over several years, products whose demand is highly seasonal, products with a demand with seasonality and a trend and products with apparently random demand.

The data provider has submitted the POS data in tabular form in csv files. In general, EDI based sales reports would also serve, but are not used in this case. The files contain product numbers, customer numbers, a time stamp and the quantity sold. A record represents the cumulative sales quantity of a particular product at a particular point of sale on a particular day. In order to meet data confidentiality requirements, the sales records have been

pseudonymized prior to further use in this study. To reduce noise, we aggregate the data to weekly sales. To avoid overfitting to historical, irrelevant data, we use a parameterizable time filter that allows us to truncate old data at any point during the forecast. (Figure 2).

3.4 Modelling Demand

The variety of characteristics associated with the available data is challenging. The time series differ significantly and show a broad spectrum of patterns including stable, non-stable, trend-afflicted, seasonality, outliers and structural breaks, in- and outphasing, to name a few. We therefore need to find techniques that can cope with any form of demand. We would also need to find a way to select an appropriate method in each case and to deal with uncertainties when necessary.

We follow an experimental approach and choose a number of candidate models to be applied on the scanner data. The candidates include machine learning (ML) approaches as well as time series analysis. For implementation purposes we rely on respective software libraries¹ based on Python. Figure 3 provides a visual impression of the fitting process of different models. The models that have been used are explained in the following together with their specific configuration and implementation details.

We use an **ARIMA** model for making short-term forecasts based on time series data. ARIMA consists of an autoregressive part (AR model) and a moving average component (MA model) [6]. The implementation that we took from the library *pmdarima* wraps the library *statsmodels* and extends it with functionality to fit a range of ARIMA models with different parameters (p, d, q). The parameters are selected to fit ARIMA on a training set. This functionality is to a large extent equivalent to the common R-library *auto.arima*.

Holt-Winters method (ETS) is an extension of exponential smoothing, which introduces three smoothing parameters for creating trend and seasonal adjusted forecasts [10, 28]. We refer to this method as ETS in this work and use the library *statsmodels* for implementation. The Seasonality is parameterized to 52 periods (weeks). The seasonal and trend components are used in an additive form.

Prophet is a forecasting approach that has been released by Facebook Research in 2018 [25]. The Prophet approach was motivated by requirements that have been identified in the context of forecasting user created events on the Facebook platform. It is intended to help business domain experts to cope with the challenges of producing high quality forecasts at scale when data science analysts are rare and automatic forecasting techniques are hard to tune. These circumstances are somewhat similar to our case study. Prophet uses an additive time series model covering non-periodic changes (trend), periodic changes (e.g. weekly or yearly seasonality) and the effects of holidays that may occur on potentially irregular schedules. Like in ARIMA, an error term is used to represent stochastic effects that cannot be explained by the model. Periodic patterns are modeled with Fourier series. As an additive component, Prophet's seasonality model is similar to the approach of exponential smoothing. The trend model combines nonlinear growth that saturates at a carrying capacity with piecewise linear growth with change points. In order to capture the effects of holidays and events, lists with specific dates can be used as an input. Prophet then tries to find a parameter for each holiday that represents the corresponding change in the forecast. We use the class *Prophet* from the python library *fbprophet*. The seasonality parameter is set to weekly mode. We provide unitary german holidays as an input for the holiday component. For the trend model we provide no input, implying that change points are to be determined automatically.

¹ The implementation has been done in a Python 3.8 environment using the libraries pandas, statsmodels, sklearn, hmmlearn, pmdarima, keras and tensorflow.

Artificial Neural Networks (ANN) Various studies show superior results for ANN compared to classical methods, especially when trends and seasonal patterns occur in the data [1, 14, 19]. The frequency of the data respectively the level of aggregation also influences the performance of forecasting models. For weekly and monthly data frequencies, a better performance of ANNs, relative to ARIMA models, was observed compared to quarterly or annual data frequencies [1].

The ANN method has been implemented using the *LSTM*-packages of the Keras API for Python along with the TensorFlow framework. To reduce the risk of overfitting we have added a 0.3 drop out layer after the ANN layer. The widely accepted *adam* optimizer is applied for training the network and the MSE error metric is used as loss function. The training was based on 80 percent training data and a validation data set representing 10 percent of the total data set. Each model was trained over a maximum of 300 epochs.

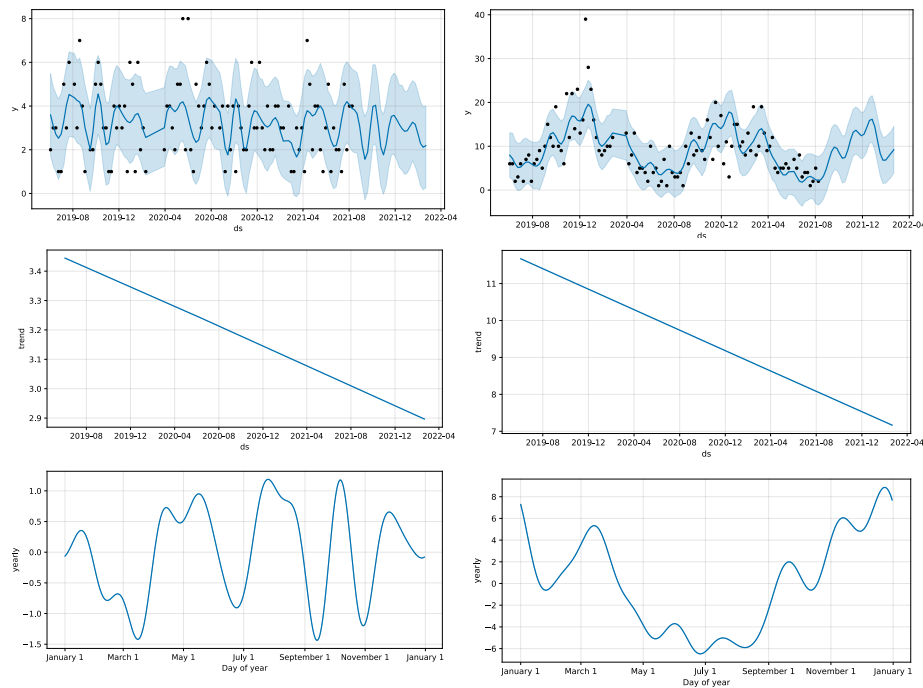


Figure 3. Model fitting and factor decomposition

3.5 Modelling Replenishment

The aim of our optimization is to determine a cost-optimal minimum inventory and the resulting replenishment quantity, taking into account capital commitment, storage space restrictions and fixed batch sizes of delivery quantities. There is a rich body of literature for inventory management in the field of operations research (see e.g. [4, 21, 24]) that addresses such problems. Among others, the newsvendor problem or economic order quantity (EOQ) are relevant concepts that have been developed in various forms. We intend to let demand forecasts drive the replenishment processes to a large extent. In the practical environment of our case, we learned that it is critical for our users to build trust in the system. This process largely depends on the degree to which the output of a system can be understood by users. In our case, users would accept limitations in terms of accuracy for the sake of explainability. It is essential to provide an easy way to understand, operate and control the system in order

to put users in charge of the process. We therefore needed to formulate a replenishment strategy, that is comprehensible, manageable and scalable. In summary, our discussions with stakeholders led to the following requirements:

- Comprehensible formulation of the replenishment strategy, preferably additive
- Data-driven approach, allowing current stock and sales forecast as inputs
- Only few, compelling tuning parameters to be controlled by users including cost of lost sales, inventory cost, shelf space and shipping batch size

These findings led us to refrain from extensions of NVP, EOQ and the like. NVP would require a probability distribution function of demand as input. In view of the highly heterogeneous forms of demand in the available real data, we argue that modeling demand as a random variable is subject to errors, especially in cases with high variance. Moreover, the trend component is completely absent in this model. In the context of our case, the practical idea of replenishment requires to optimize shelf inventory under the given constraints through triggering orders in the right quantity at the right time and leveraging the available data. To specify implementation, we need to introduce some formalism upfront.

Given an initial inventory we can estimate the inventory at a single retail outlet by contrasting outgoing sales and incoming deliveries. Defining $t_{n+1} = t_n + d$, we divide the timeline into discrete sections of a constant length d . While scaling d allows to adjust the granularity of time in our model, we assign d as one week, corresponding to the resolution of the data sample. Using that formalism, we can model actual sales of a certain product in a specific retail outlet within a given time span $[t_n, t_{n+1}]$ as a function $s(t_n, t_{n+1})$. Likewise, we model deliveries as a function $d(t_n, t_{n+1})$. Let i_{t_0} denote the inventory at time t_0 . We define $\hat{i}(t_0) = i_{t_0}$ and estimate the inventory $\hat{i}(t_n)$ at any time t_n with $n > 1$ as

$$\hat{i}(t_n) = \hat{i}(t_{n-1}) + d(t_{n-1}, t_n) - s(t_{n-1}, t_n). \quad (3.4.1)$$

Given an initial inventory at the POS at a specific point of time, we retrieve the inventory at the end of the regarded period by adding incoming goods while subtracting outgoing goods. An appropriate order quantity at that very point of time is given by the sales forecast minus the current inventory plus a definable safety amount i_{safe} :

$$o(t_n) = p(t_n, t_{n+1}) - \hat{i}(t_n) + i_{safe}. \quad (3.4.2)$$

This way we provide a comprehensible, additive form. The safety amount is an intuitive parameter that will be used to cope with uncertainties. It needs to be defined carefully, high enough to reduce stock-outs but low enough to avoid overstocks or even spoilage. In order to include products that can only be ordered in certain batch sizes b , we extend the order function as follows:

$$o_{batch}(t_n) = \max \left[\text{round} \left(\frac{p(t_n, t_{n+1}) - \hat{i}(t_n) + i_{safe}}{b} \right) \cdot b, 0 \right]. \quad (3.4.3)$$

Given a forecast model $m(t_n, t_{n+1})$ that returns the sales prediction for any horizon $h = [t_n, t_{n+1}]$, the function $o(t_n)$, respectively $o_{batch}(t_n)$, would return exactly enough items to match the forecast plus the safety stock i_{safe} .

3.6 Evaluation of Demand Models

We identify the most accurate model by computing standard error metrics MAE, MASE, RMSE and MAPE for all combinations of product and outlet. Furthermore, the cumulative errors as well as the cumulative over- and underestimations are taken into account as quality figures. Especially from the point of view of the industry partner, these metrics represent

important indicators. As benchmarks serve a simple moving average of 1 (the last observed value), which we refer to as Naive as well as the actual value from the prior season, which we refer to as Seasonal Naive.

Table 2. Key forecast quality metrics

KQI		ARIMA	ETS	Prophet	ANN	Naive	Seas. Naive
MAE	Min	0,79	0,97	1,29	0,92	0,00	1,00
	Median	3,57	4,00	4,14	3,46	4,50	4,83
	Max	71,91	128,30	8611,56	74,42	123,83	86,00
	Std. Dev.	10,49	16,03	364,99	10,11	13,30	14,89
MASE	Min	0,16	0,15	0,17	0,17	0,00	0,18
	Median	0,37	0,41	0,39	0,35	0,44	0,48
	Max	48,43	8,33	5741,04	32,00	7,73	35,50
	Std. Dev.	2,66	0,48	244,17	2,25	0,44	2,11
MAPE	Min	0,13	0,16	0,16	0,15	0,00	0,20
	Median	0,43	0,52	0,52	0,45	0,52	0,63
	Max	48,43	61,60	6898,95	41,33	60,68	49,42
	Std. Dev.	3,48	3,56	301,14	3,04	3,38	3,77
RMSE	Min	1,28	1,00	1,79	1,00	0,00	1,35
	Median	4,55	5,00	5,14	4,47	5,56	6,16
	Max	81,56	136,14	8789,80	85,20	131,25	97,71
	Std. Dev.	12,65	18,75	410,51	12,27	15,63	17,70
Err Total	Min	-741,89	-1282,03	-11569,66	-838,00	-979,00	-854,00
	Median	-9,98	3,56	5,39	-5,00	-7,00	5,00
	Max	862,94	1539,54	9147,40	495,00	1486,00	706,00
	Std. Dev.	108,29	190,24	590,45	93,40	164,40	130,61
Err Over	Min	0,00	0,00	0,00	0,00	0,00	0,00
	Median	0,00	3,56	5,39	0,00	0,00	5,00
	Max	862,94	1539,54	9147,40	495,00	1486,00	706,00
	Std. Dev.	55,21	84,27	312,79	47,40	97,60	50,38
Err Under	Min	-741,89	-1282,03	-11569,66	-838,00	-979,00	-854,00
	Median	-9,98	0,00	0,00	-5,00	-7,00	0,00
	Max	0,00	0,00	0,00	0,00	0,00	0,00
	Std. Dev.	88,00	165,21	497,68	75,62	120,92	115,20
Number of best forecasts		46	12	11	23	3	5

We apply a rolling-window cross-validation (CV) in order to identify the best forecast algorithm for each product in each region. For this purpose, 10 forecasts are generated for each product in each region for the upcoming quarter (the next 12 weeks), with each of the 10 CV runs using one additional observation more for training the forecast algorithm. So, in total, $10 \times 10 \times 10 = 1000$ forecasts (number of products * number of regions * number of CV runs) are computed for each algorithm for 12 weeks. For each run, the defined list of key quality indicators KQIs is calculated. Subsequently a ranking is computed for each CV run and each combination of product and region across all algorithms. The method with the smallest absolute value (per KQI) is given rank 1, the one with second smallest value is given rank 2, and so on. For each combination of product and region, the algorithm that has achieved the lowest overall rank on average across all CV runs and all quality measures is now determined.

From Table 2, it becomes immediately evident that none of the forecasting methods is consistently superior to the others. For the 100 combinations of products and regions examined, it can rather be stated that the differences in the historical demand distributions are so significant that the greatest possible estimation accuracy of the forecasting models can only be achieved by using multiple algorithms. Although the ARIMA method, for example, is identified as optimal in 46 out of 100 cases, in other cases the forecasts deviated considerably from the realized values even with this method. Similar results can be observed for any other statistical or ML approach. In 8 out of 100 cases, the naive extrapolation even beats the algorithmic forecast. An overall indication of the models performance is given in figure 4.

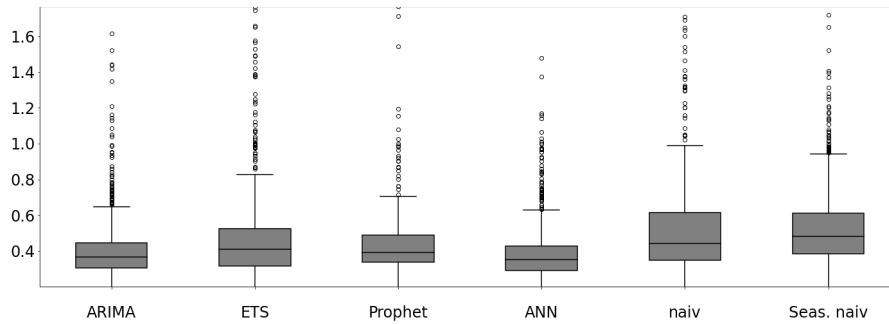


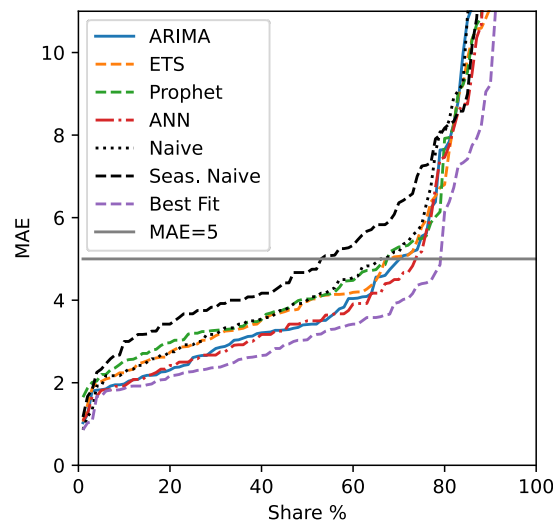
Figure 4. Distribution of MASE.

All statistical and ML methods outperform the naive benchmarks with respect to the variance of the estimation errors and also with respect to the estimation error on average. Nevertheless, there are significant outliers for each method. This is caused by the strong variability of the time series in the dataset. While some algorithms cope better with apparently random demand curves, others have advantages with highly seasonal time series.

Table 3. Quantiles of MAE distribution

%	Arima	ETS	Prophet	ANN	Naive	Seas. Naive	Best Fit
10	1,97	2,24	2,53	1,92	2,25	3,00	1,86
20	2,31	2,74	2,97	2,42	2,75	3,42	2,17
30	2,83	3,14	3,27	2,67	3,17	3,83	2,37
40	3,21	3,52	3,53	3,17	3,55	4,17	2,66
50	3,41	3,97	4,03	3,50	4,00	4,67	3,04
60	4,04	4,19	4,48	3,91	4,55	5,33	3,42
70	4,98	5,06	5,29	4,50	5,17	6,36	3,95
80	7,65	6,80	7,92	7,46	8,08	8,17	6,14
90	15,70	11,10	12,99	12,58	17,42	16,75	9,22
100	57,87	79,64	1.983,39	52,83	61,82	76,00	27,46

In the practical context of the case study, we regard forecasts to be sufficiently accurate when the MAE is less or equal to 5 as this is the standard batch size for shipping. Table 3 shows the quantiles of MAE, a visual indication is given in figure 5. Under this assumption the benchmark models Naïve (Seasonal Naïve) generate valid forecasts for 66% (50%) of the cases. The models Arima, ETS and Prophet serve for 70%, 68% and 67%, barely beating the naïve baseline. ANN generates valid forecasts for 73%. If we select the most accurate model for each single time series (pair of product and region) in regard to median MAE over all 10 CV runs, sufficient forecasts can be produced in 79% (Best Fit).

**Figure 5.** Distribution of MAE

3.7 Evaluation of Replenishment Model

To evaluate the performance of the replenishment process, we propose a what-if analysis, simulating sales performance as an outcome of the replenishment orders we generate from the data. The replenishment function is to be customized with the parameters initial stock, safety stock and a demand model. We use a backward simulation to generate the expected sales as a function of estimated demand and current stocks. Rolling forecasts are generated

with a horizon of two weeks each. The following example illustrates the process for a concrete setting, having a two-week service interval a one-week delivery time. In this particular scenario, replenishment orders and deliveries are to be conducted alternating weekwise in order to utilize latest stock information when ordering. Following the notation of 3.4.3 we define

$$o_{2wk}(t_n) = \begin{cases} 0, & \text{if } n \in (1,3,5, \dots) \\ o_{batch}(t_n), & \text{else} \end{cases} \quad (3.6.4)$$

Deliveries are simulated by defining

$$\hat{d}(t_n, t_{n+1}) = o_{2wk}(t_n) \quad (3.6.5)$$

That strategy creates replenishment orders at odd week numbers, so that deliveries arrive at even week numbers. As a result, we receive a what-if-scenario, demonstrating the effects on shipments and stocks. By tuning the safety stock parameter i_{safe} we can express our preference regarding the trade-off between product availability and inventory cost as shown in figure 6.

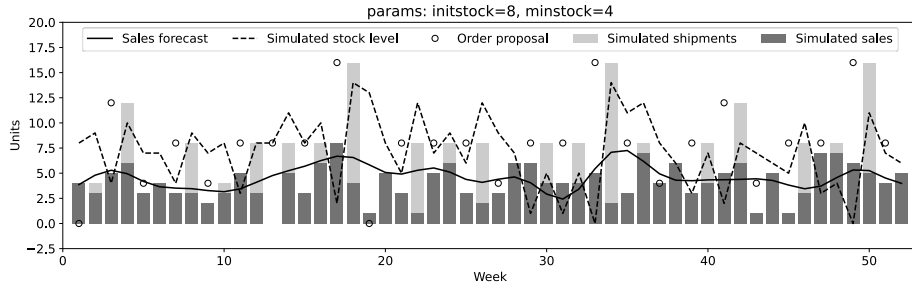


Figure 6. What-if simulation

In order to evaluate the performance of the replenishment strategy, we suggest a set of key performance indicators as shown in table 3. Applying the KPIs on the test scenario yields to the following insights. When minimizing stockouts (respectively maximizing sales) is priority, respectively high safety stocks ($i_{safe}=8$) will serve. However, this configuration leads to relatively high average stocks of 10,98 units. The resulting inventory cost can be derived multiplying it with the product- and outlet-specific inventory cost. When perishable products are involved, the spoilage rate can be approximated by comparing the product-specific durability with the turnover rate.

Table 3. KPI simulation for different what-if scenarios

KPI	Formula	Safety stock (minstock)				
		0	4	8	12	16
Average stock level (i_{avg})	$\frac{1}{n} \sum_{i=1}^n i(t_i)$	3,4	7,06	10,98	14,98	18,98
Cost of inventory	$c_0 \sum_{i=1}^{n-1} \frac{i(t_i) + i(t_{i+1})}{2}$	2,33	5,6	9,83	13,46	17,62
Total units delivered	$\sum_{i=1}^{n-1} \hat{d}(t_i, t_{i+1})$	188	208	216	220	224

Total units sold	$\sum_{i=1}^{n-1} \hat{s}(t_i, t_{i+1})$	121	174	204	212	212
	$\hat{s}(t_i, t_{i+1}) = \begin{cases} s(t_i, t_{i+1}), & \text{if } \hat{i}(t_i, t_{i+1}) \geq 0 \\ s(t_i, t_{i+1}) + \hat{i}(t_i, t_{i+1}), & \text{else} \end{cases}$					
Lost sales (\bar{s})	$\sum_{i=1}^{n-1} \max\{0, s(t_i, t_{i+1}) - \hat{i}_i\}$	91	38	8	0	0
Cost of lost sales	$c_1 \bar{s}$	182	76	16	0	0

A moderate or even low safety stock level of 4 (or respectively 0) units would reduce the average stock level significantly to 6,98 (or even 2,98) units. In case of perishable products, this measure allows to reduce the risk of spoilage as the stock coverage goes down from 18,85 over 11,99 to 5,12 days when tuning down the safety stock parameter (figure 7). However, that way, stockouts become more probable. The simulation estimates a sales decline by approximately 1.9% to even 12% when operating with less safety stock.

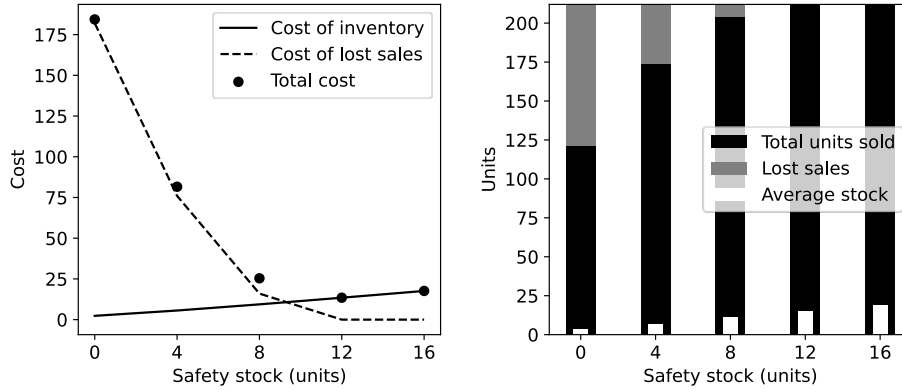


Figure 7. KPI simulation for different safety stock settings

4 Discussion

The paper investigates demand forecasting procedures in the context of VMI processes in food retail. In the course of a case study we developed a mechanism for automatic replenishment, leveraging the existing data sources to generate model-based sales forecasts. We determined optimal forecasts using a mix of different statistical and ML algorithms and designed a conceivable simulation of KPIs, which serves as a decision support tool. It can be observed that:

(a) the automation of sales forecasts seems largely feasible. This can significantly reduce the required manual effort during the planning process. However, despite the consideration of a wide range of algorithms, sales forecasts (in model testing) still deviate strongly from realized values in individual cases.

b) the outlined disposition mechanism, due to the multitude of possible influencing variables and parameters, does not go beyond an approximate solution. It is not possible to decide conclusively which criterion should be the decisive factor in determining whether product- and region-specific replenishment can be automated.

The findings from both artifacts of our study still represent considerable added value for use in practice. In fact, for the remaining relatively imprecise forecasts, decision-makers still face the challenge of manually weighing the risk of shortages and overstocks or spoilage. However, since the planning effort can be greatly reduced, planners have greater capacity to focus on "difficult" time series.

The limited number of forecasting algorithms considered is a limitation of our research. All time series-based models attempt to find patterns in historical data, which are then extrapolated to predict the future. There may still be some potential for optimization by hyperparameter tuning or adding more regressors to the models. However, the integration of exogenous regressors could increase the bias in the presence of data quality issues. For example, in case of stockouts, no information on demand is registered. Since this is equivalent to zero demand, the predictions tend to underestimate the demand for a product and could assign a dampening influence to any exogenous regressor. On the other hand, product substitutions, which are a direct response to stockouts, lead to higher sales quantities of substitute products. Here, systematic overestimates of demand are inevitably produced. While a number of methods for solving these problems have been proposed in related works, most of them are too complex for practical use [5]. Promising approaches are based on modeling demand as a function of product availability, variety, and time. However, such approaches impose much higher requirements on data collection and quality. The added value of the approach outlined in our study lies in its immediate applicability as it is solely built on easily obtainable transactional sales data.

5 Conclusion and Outlook

With this work, we present a set of tools for VMI in retail. Based on a data-driven approach, we generate forecasts from historical sales data using time series analysis and machine learning. This helps decision makers weigh the risk of stockouts against the cost of holding inventory. In practice, while for long-life products the avoidance of stockouts at lowest cost is crucial, for perishable products there is likely an optimal trade off, depending on the cost structure. To address the challenge of forecasting the diverse patterns of demand, we have implemented a set of forecasting algorithms and a routine to determine the most accurate model. With the goal of automating sales forecasting, we propose a set of practical forecasting quality metrics. Moreover, we implemented a what-if simulation to model the replenishment process. The forecasting procedures and the simulation have been implemented as prototypic software to help decision makers understand and adjust the logistical parameters that drive their business.

The findings and results of this work can be transferred to similar processes where inventories are managed by a third party supplier. It should be emphasized that accurate forecasting is the key to delinking the processes of order entry and (physical) in-store replenishment. We believe that it would be worth investigating whether complete delinking of the sub-processes can be achieved. Further research should hence address the necessary preconditions, including how to achieve sufficient forecast quality across a broad portfolio of products and how to make accurate inventory information available in real time.

In this context, it becomes apparent that while communication standards such as EDI are widely used, this does not necessarily mean that detailed data is exchanged at the POS level between partners along the supply chain. In fact, while conducting our case study, we found that few retailers in the grocery sector offer services for data exchange at transaction level. As a result, we conclude that appropriate collaboration models need to be found that balance

retailers' legitimate data security interests with the value of data-driven approaches to retail logistics.

Acknowledgements

The authors would like to thank Prof. Dr. Marco Barenkamp, LMIS AG Osnabrück, for insightful comments and suggestions for implementation work and improving the manuscript.

References

1. Alon, I. et al.: Forecasting aggregate retail sales: *J. Retail. Consum. Serv.* 8, 3, 147–156 (2001). [https://doi.org/10.1016/s0969-6989\(00\)00011-4](https://doi.org/10.1016/s0969-6989(00)00011-4).
2. Andersson, H. et al.: Industrial aspects and literature survey: Combined inventory management and routing. *Comput. & Oper. Res.* 37, 9, 1515–1536 (2010).
3. Arunraj, N.S., Ahrens, D.: A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *Int. J. Prod. Econ.* 170, 321–335 (2015). <https://doi.org/10.1016/j.ijpe.2015.09.039>.
4. Beamon, B.M., Kotleba, S.A.: Inventory management support systems for emergency humanitarian relief operations in South Sudan. *Int. J. Logist. Manag.* (2006).
5. Boone, T. et al.: Forecasting sales in the supply chain: Consumer analytics in the big data era. *Int. J. Forecast.* 35, 1, 170–180 (2019). <https://doi.org/10.1016/j.ijforecast.2018.09.003>.
6. Box, G.E.P. et al.: *Time series analysis: forecasting and control*. John Wiley & Sons (2015).
7. EHI Retail Institute: Gründe für Out-of-Stock-Quote im Einzelhandel in Deutschland 2013, <https://de.statista.com/statistik/daten/studie/372043/umfrage/gruende-fuer-out-of-stock-quote-im-einzelhandel-in-deutschland/>, last accessed 2020/08/24.
8. Fitzsimons, G.J.: Consumer Response to Stockouts. *J. Consum. Res.* 27, 2, 249–266 (2000). <https://doi.org/10.1086/314323>.
9. FMI/GMA: Solving the Out-of-Stock Problem: A FMI/GMA Trading Partner Alliance Report. (2015).
10. Gardner, E.S.: Exponential smoothing: The state of the art-Part II. *Int. J. Forecast.* 22, 4, 637–666 (2006). <https://doi.org/10.1016/j.ijforecast.2006.03.005>.
11. Harrison, J.M., Van Mieghem, J.A.: Multi-resource investment strategies: Operational hedging under demand uncertainty. *Eur. J. Oper. Res.* 113, 1, 17–29 (1999).
12. Huber, J. et al.: A data-driven newsvendor problem: From data to decision. *Eur. J. Oper. Res.* 278, 3, 904–915 (2019).
13. Khashei, M., Bijari, M.: An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Syst. Appl.* 37, 1, 479–489 (2010). <https://doi.org/10.1016/j.eswa.2009.05.044>.
14. Kihoro, J. et al.: Seasonal time series forecasting: a comparative study of arima and ann models. *African J. Sci. Technol.* 5, 2, (2006). <https://doi.org/10.4314/ajst.v5i2.15330>.
15. Levi, R. et al.: The data-driven newsvendor problem: new bounds and insights. *Oper. Res.* 63, 6, 1294–1306 (2015).
16. Ma, S. et al.: Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. *Eur. J. Oper. Res.* 249, 1, 245–257 (2016). <https://doi.org/10.1016/j.ejor.2015.08.029>.
17. Mieghem, J.A. Van, Rudi, N.: Newsvendor networks: Inventory management and capacity investment with discretionary activities. *Manuf. & Serv. Oper. Manag.* 4, 4, 313–335 (2002).
18. Narayanan, A. et al.: Demand and order-fulfillment planning: The impact of point-of-sale data,

- retailer orders and distribution center orders on forecast accuracy. *J. Oper. Manag.* 65, 5, 468–486 (2019). <https://doi.org/10.1002/joom.1026>.
19. Palkar, A. et al.: Demand Forecasting in Retail Industry for Liquor Consumption using LSTM. Presented at the August 4 (2020). <https://doi.org/10.1109/icesc48915.2020.9155712>.
 20. Pongdatu, G.A.N., Putra, Y.H.: Seasonal Time Series Forecasting using SARIMA and Holt Winter's Exponential Smoothing. In: *IOP Conference Series: Materials Science and Engineering*. p. 012153 Institute of Physics Publishing (2018). <https://doi.org/10.1088/1757-899X/407/1/012153>.
 21. Ravinder, H., Misra, R.B.: ABC analysis for inventory management: Bridging the gap between research and classroom. *Am. J. Bus. Educ.* (2014).
 22. Ren, S. et al.: A Comparative Study on Fashion Demand Forecasting Models with Multiple Sources of Uncertainty. *Ann. Oper. Res.* 257, 1–2, 335–355 (2017). <https://doi.org/10.1007/s10479-016-2204-6>.
 23. Schottmiller, P.: Why Do You Still Have Out-of-Stocks? A New Approach to Solving an Old Problem. , San Jose (2010).
 24. Silver, E.A.: Operations research in inventory management: A review and critique. *Oper. Res.* 29, 4, 628–645 (1981).
 25. Taylor, S.J., Letham, B.: Forecasting at Scale. *Am. Stat.* 72, 1, 37–45 (2018). <https://doi.org/10.1080/00031305.2017.1380080>.
 26. Tratar, L.F.: Joint optimisation of demand forecasting and stock control parameters. *Int. J. Prod. Econ.* 127, 1, 173–179 (2010). <https://doi.org/10.1016/j.ijpe.2010.05.009>.
 27. Da Veiga, C.P. et al.: Demand forecasting based on natural computing approaches applied to the foodstuff retail segment. *J. Retail. Consum. Serv.* 31, 174–181 (2016). <https://doi.org/10.1016/j.jretconser.2016.03.008>.
 28. Da Veiga, C.P. et al.: Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *WSEAS Trans. Bus. Econ.* 11, 1, 608–614 (2014).