creativestyle
A SMILE GROUP COMPANY

VON DER SYSTEMWAHL BIS ZUM GO-LIVE: PRAXISWISSEN FÜR MODERNE E-COMMERCE-PROJEKTE

E-COMMERCE REPLATFORMING

Karol Schoensee

2026

INHALTSVERZEICHNIS

VORWORT 4
ENTSCHEIDUNGSPHASE 6
2.1 Signale für ein Replatforming 7
2.2 Ziele und Erwartungen definieren 9
Kostenoptimierung 9
Skalierbarkeit und Flexibilität 9
Umsatzsteigerung 10
SMARTe Ziele setzen 10
Wirtschaftliche KPIs (Messung von Umsatz, Rentabilität und Wachstum) 11
Technische KPIs (Messung der Technische Stabilität, Performance und Nutzererfahrung) 12
Nutzerzentrierte KPIs (Messung der Nutzererfahrung) 12
Prozessbezogene KPIs (Messung der Leistung von logistischen Prozessen) 13
Strategische KPIs (Messung der Qualität der E-commerce Software) 13
2.3 Alternativen zum Replatforming 14
2.4 Funktionale Anforderungen definieren 17
2.5 Die neue E-Commerce-Plattform wählen 21
Das richtige technische Fundament 21
Kriterien zur Wahl der richtigen E-Commerce Plattform 27
2.6 Externe Agentur vs. Inhouse Team 28
Option 1: Externe Agentur 29
Option 2: Inhouse-Entwicklung 29
Option 3: Hybrid-Modell 30
2.7 Die richtige Agentur wählen 31
2.8 Eine Entscheidung mittels Business Case treffen 34
2.9 Dos und Don'ts in der Entscheidungsphase 37
ANALYSE 38
3.1 Analyse Ist-Zustand 39
Technologische Aspekte 39
Funktionale Analyse Storefront und E-Mails 41
Funktionale Analyse Admin Bereich 69
Analyse der Systemlandschaft 102
PIM-Integration 104
ERP-Integration 106
CRM-Integration 108
PSP-Integration 111
Analyse der Business-Aspekte 112
3.2 Stakeholder Analyse 115
3.3 Wettbewerber Analyse 116

3.4 Zielgruppenanalyse 118
3.5 Business Model Canvas 120
3.6 Dos und Don'ts in der Analyse 123
KONZEPTION UND UMSETZUNG 125
4.1 Projektmanagement 126
SCRUM vs. Wasserfall 126
Was sind jeweils die Vor- und Nachteile von SCRUM? 127
Was sind jeweils die Vor- und Nachteile von Wasserfall? 128
SCRUM in der Praxis 129
Product Owner bestimmen 131
Product Backlog definieren 133
Projekt Timeline 135
Change Management 136
1. Beschreibung und Bewertung der geplanten Veränderung: 137
2. Einführung planen: 137
3. Stakeholder einbeziehen: 137
4. Überprüfung und Erfolgsmessung: 138
4.2 Storefront Konzeption 139
4.3 Integrationskonzept 142
4.4 Migrationskonzept 145
4.5 Infrastruktur aufsetzen 148
4.6 QA Prozess aufsetzen 151
4.7 Dos und Don'ts in der Konzeption und Umsetzung 154
TESTING UND GO-LIVE 156
5.1 SEO Basics 157
5.2 Analytics und Tag Manager Umzug 159
5.3 Schulung neues E-Commerce System 162
5.4 Projektdokumentation 164
5.5 Smoketesting 167
5.6 Integrationstests und End-to-End Tests 171
Produktimport aus dem PIM: 171
Kundenanlageprozess: 172
Check-out und Fulfillment: 172
5.7 Go-Live Agenda 175
5.8 Nach dem Go-Live 181
5.9 Dos und Don'ts im Testing und Go-Live 182
SCHLUSSWORT 184
ÜBER DEN AUTOR 185
FUSSNOTEN & QUELLEN 186

KAPITEL 1

VORWORT

Der E-Commerce hat in den letzten fünf Jahren eine Dynamik erlebt, wie kaum eine andere Branche. Unternehmen mussten sich ständig anpassen – sei es an neue Technologien, veränderte Marktbedingungen oder unerwartete globale Ereignisse. Von der Einführung rechtlicher Vorschriften wie der DSGVO, der Omnibus- und den Accessibility-Richtlinien bis hin zu tiefgreifenden gesellschaftlichen und wirtschaftlichen Umbrüchen: Der E-Commerce war und ist ein Spiegel unserer Zeit.

Die Pandemie hat als Akzelerator der Digitalisierung gewirkt, das Kaufverhalten der Konsumenten grundlegend verändert und den Druck auf Händler erhöht, in digitale Kanäle zu investieren. Gleichzeitig wurden Supply Chains massiv gestört, was die Herausforderungen für E-Commerce-Owner weiter verschärfte. Der Ukraine-Krieg, die Inflation und die sogenannte E-Commerce-Krise haben die Branche vor neue Herausforderungen gestellt: Händler müssen umstrukturieren, Prozesse rationalisieren und gleichzeitig flexibel auf Marktdynamiken reagieren.

Technologische Innovationen eröffnen dabei neue Chancen. Microservices, Headless-Commerce und Composable Commerce haben sich zu relevanten Architekturen entwickelt, während Künstliche Intelligenz (KI) ein weiterer Akzelerator technologischen Fortschritts ist – sei es im Bereich der Personalisierung, der Prozessautomatisierung oder der Datenanalyse.

Noch nie gab es so viele Möglichkeiten im E-Commerce, aber gleichzeitig zwingen die oben genannten Ereignisse Unternehmen dazu, ihre Strategien zu überdenken. Ein Replatforming, also der Wechsel auf eine neue E-Commerce-Plattform, ist dabei häufig unvermeidbar. Doch wann ist ein solcher Wechsel sinnvoll? Was gilt es zu beachten, und welche Alternativen gibt es?

Dieses Buch richtet sich an E-Commerce Owner, also den Personen mit Kontrolle über die E-Commerce-Plattform, sowie an Manager, Entwickler und Agenturen. Das Buch dient dabei als Leitfaden, Werkzeugkasten und Inspirationsquelle im Hinblick auf die folgenden Fragen:

- Wann ist ein Replatforming sinnvoll, und welche Alternativen gibt es?
- Wie wählt man die richtige E-Commerce-Plattform und den passenden Partner?
- Welche Vorarbeiten sind nötig, um ein Replatforming erfolgreich zu gestalten?
- Was sollte man bei der Konzeption und Umsetzung beachten?
- Wie bereitet man sich optimal auf den Go-Live vor?

Ziel des Buches ist nicht nur einen Leitfaden an die Hand zu geben, sondern auch die Hintergründe und Zusammenhänge der wichtigsten Entscheidungen verständlich zu machen. Der Erfolg des Projekts hängt nicht nur von der Wahl der richtigen Technologie ab, sondern auch von einer durchdachten Strategie und einer klaren Umsetzung.

Ich wünsche Ihnen viel Erfolg auf Ihrer Reise zu einem neuen und zukunftssicheren E-Commerce System!

Karol Schoensee

KAPITEL 2

ENTSCHEIDUNGSPHASE

Die Entscheidungsphase ist eine der wichtigsten Etappen im E-Commerce-Replatforming-Prozess. Sie legt den Grundstein für den Erfolg des Projekts, da in diesem Abschnitt strategische und operative Entscheidungen getroffen werden, die den gesamten weiteren Verlauf bestimmen. Zunächst gilt es, klare Signale für ein Replatforming zu erkennen und die Frage zu klären, ob Optimierungen am bestehenden Shop ausreichen oder ob ein vollständiger Wechsel sinnvoller ist.

Ein zentraler Bestandteil der Entscheidungsphase ist die Definition der Ziele und Erwartungen, die an die neue Plattform gestellt werden. Dies bildet die Basis für die Erstellung der funktionalen Anforderungen, die sicherstellen, dass das neue System optimal zu den Geschäftsprozessen passt. Ebenso wird in dieser Phase das technologische Fundament festgelegt, also ob die neue Plattform beispielsweise ein monolithisches System, Headless oder eine SaaS-Lösung sein soll.

Darüber hinaus muss sorgfältig geprüft werden, welche E-Commerce-Plattform die beste Wahl für die individuellen Bedürfnisse des Unternehmens darstellt. Kriterien wie Funktionalitäten, Flexibilität, Kosten und das Ökosystem spielen hier eine entscheidende Rolle. Ebenso stellt sich die Frage, ob die technische Implementierung durch eine externe Agentur oder mit einem inhouse Team erfolgen soll – eine Entscheidung, die nicht nur das Budget, sondern auch den Projektverlauf maßgeblich beeinflusst.

Die Entscheidungsphase wird durch die Erstellung eines Business Cases unterstützt, der die wirtschaftliche Sinnhaftigkeit des Projekts untermauert. Abschließend werden in einer Zusammenfassung die Do's und Don'ts beleuchtet, um Fehler zu vermeiden und das Fundament für einen erfolgreichen Projektstart zu legen.

2.1 Signale für ein Replatforming

Im Folgenden werden Signale beschrieben, die darauf hindeuten können, dass ein Replatforming für den Online-Shop notwendig ist.

Ein besonders klares Indiz für Optimierungsbedarf sind **Beschwerden von Kunden und Nutzern des Shops**. Diese betreffen oft Aspekte wie lange Ladezeiten, technische Fehler oder das Fehlen moderner Funktionen. Heutzutage setzen Nutzer von Online-Shop grundlegende Features als Standard voraus – von einer schnellen und intuitiven Produktsuche über Filter- und Sortiermöglichkeiten bis hin zu einem reibungslosen Check-out mit vielfältigen Zahlungs- und Versandoptionen. Funktionen wie die Speicherung von Kreditkartendaten (datenschutzkonform gemäß PCI-DSS Compliance) gelten mittlerweile als selbstverständlich. Werden diese Erwartungen nicht erfüllt, verlassen unzufriedene Nutzer den Shop ohne Conversion. Während manche Kunden ihre Frustration äußern, etwa per E-Mail oder im Support-Chat, entscheiden sich viele, stillschweigend zur Konkurrenz zu wechseln. Beschwerden sollten daher stets ernst genommen werden, denn sie sind oft nur die Spitze eines Eisbergs.

Auch auf der Kostenseite können **Beschwerden von Mitarbeitern** wichtige Signale für Handlungsbedarf liefern. Content-Manager klagen häufig über aufwändige Prozesse bei der Pflege oder Erstellung von Content, insbesondere wenn dazu Entwicklerkenntnisse oder sogar Deployments notwendig sind. Erwartet wird eine einfache Handhabung von wiederverwendbaren Content-Bausteinen und eine reibungslose Internationalisierung, die ohne manuelle Arbeit auskommt.

Produktdatenmanager benötigen effektive Werkzeuge, um Kataloge effizient zu aktualisieren – vor allem bei Shops mit mehreren Tausend Artikeln (SKUs) und mehrsprachigen Beschreibungen. Massenänderungen von Attributen, wie das gleichzeitige Aktualisieren von Materialangaben für viele Produkte, sind hier ein Muss. Fehlende Automatisierungen in diesem Bereich verursachen nicht nur unnötige Kosten, sondern können auch zu fehlerhaften oder unvollständigen Produktinformationen führen, was wiederum potenziellen Umsatzverlust bedeutet.

Für Online-Marketer sind Flexibilität und Autonomie essenziell. Kampagnen und Rabatte, wie etwa Gutscheincodes, sollten im Admin-Bereich einfach einstellbar sein, ohne dass Entwickler eingreifen müssen. Zudem müssen Tracking-Tools wie Google

Tag Manager problemlos konfiguriert und Drittanbieter-Pixel (z. B. für Facebook oder Google Ads) schnell eingebunden werden können.

Auch Mitarbeiter im Bereich Buchhaltung und Fulfillment benötigen gut abgestimmte Prozesse. Der automatisierte Import von Bestelldaten ins ERP oder die Buchhaltungssoftware muss zuverlässig alle relevanten Informationen umfassen, darunter Rechnungs- und Lieferadressen, Artikelnummern, Versand- und Zahlungsmethoden sowie Rabatte und Steuern.

Neben den offensichtlichen Signalen gibt es auch schleichende Entwicklungen, die ein Replatforming erfordern könnten. Ein Beispiel hierfür sind kontinuierlich **steigende Total Cost of Ownership** ("TCO") bei gleichzeitig stagnierendem Umsatz. Die TCO umfassen unter anderem Hosting- und Lizenzgebühren sowie Kosten für interne Entwickler oder Agenturen. Sinkt der Deckungsbeitrag des Shops dadurch über längere Zeit, kann die Rentabilität gefährdet sein.

Weitere Hinweise auf Handlungsbedarf sind **Skalierungsprobleme** oder der Einsatz veralteter Technologien. Ein Shop, der bei starkem Traffic, etwa an Black Friday, ausfällt, verliert potenziell erheblichen Umsatz. Ähnlich problematisch wird es, wenn neue Länder oder Sprachen nur mit großem Aufwand eingeführt werden können.

Ein weiteres kritisches Signal ist das **End of Life (EOL) von Software** oder Bibliotheken, die im Shop verwendet werden. Dies kann nicht nur technische Probleme verursachen, sondern auch rechtliche Konsequenzen haben. Artikel 32 der DSGVO fordert, dass technisch-adäquate Schutzmaßnahmen („Stand der Technik") eingesetzt werden, um ein angemessenes Sicherheitsniveau zu gewährleisten. Werden Systeme nicht rechtzeitig aktualisiert, drohen Datenschutzrisiken und damit verbundene Sanktionen.

✓ CHECKLISTE SIGNALE FÜR EIN REPLATFORMING

- Es wurden Beschwerden von externen Stakeholdern (z.B. Kunden, Lieferanten, Agenturen, ...) wahrgenommen. Falls vereinzelt Beschwerden kommen, wurden die die externen Stakeholder z.B. per Umfrage nach eventuellen Beschwerden gefragt.
- Es wurden Beschwerden von internen Stakeholdern (Mitarbeiter) wahrgenommen. Falls vereinzelt Beschwerden kommen, wurden die die internen Stakeholder z.B. per Umfrage nach eventuellen Beschwerden gefragt.

- Der E-Commerce Owner prüft die aktuelle Plattform auf Wirtschaftlichkeit (z.B. Verhältnis TCO und Umsätze) über einen Zeitraum, sowie die Skalierbarkeit und eventuelle EOL Deadlines.

2.2 Ziele und Erwartungen definieren

Nachdem Signale für ein mögliches Replatforming erkannt wurden, sollte sich der E-Commerce Owner intensiv mit den Zielen und Erwartungen an die neue Plattform auseinandersetzen. Diese Ziele können aus unterschiedlichen Perspektiven definiert werden:

Kostenoptimierung

Ein zentrales Ziel ist häufig die Reduktion der Total Cost of Ownership (TCO) des Shops. Dies kann durch den Wechsel auf eine Software mit niedrigeren Lizenzgebühren oder sogar auf ein Open-Source-System erreicht werden. Ebenso können die Wahl einer kostengünstigeren Agentur oder die Nutzung von Software-as-a-Service (SaaS)-Systemen, die geringere Wartungs- und Hosting-Kosten mit sich bringen, Kosten einsparen. Allerdings sollten Wechselkosten, wie das Onboarding einer neuen Agentur, nicht außer Acht gelassen werden. Auch können Automatisierungen, beispielsweise durch effizientere PIM- oder ERP-Integrationen, die manuellen Aufwände und potenzielle Fehler reduzieren und so die Betriebskosten langfristig senken.

Skalierbarkeit und Flexibilität

Ein Shopsystem sollte Wachstum unterstützen und die Möglichkeit bieten, neue Märkte oder Funktionen mit überschaubarem Aufwand zu integrieren. Kann das System neue Produkte, Lieferländer oder Sprachen ohne große Entwicklungsarbeiten live schalten? Hält die Plattform auch unerwartet hohem Traffic, etwa durch Marketing-Kampagnen, stand? Verfügt sie über einen Marktplatz für Erweiterungen und bietet ein breites Angebot an Entwicklern und Agenturen? Diese Fragen sollten klar beantwortet werden. Ein neues System muss in der Lage sein, Kunden- und Mitarbeiterbeschwerden durch bessere Funktionen zu adressieren und gleichzeitig die Skalierbarkeit zu gewährleisten.

Umsatzsteigerung

Ein weiteres Ziel des Replatformings ist häufig eine Verbesserung des Umsatzes. Dies kann durch schnellere Ladezeiten, eine bessere Usability oder SEO-Optimierungen erreicht werden, etwa durch zusätzliche indexierte Elemente und optimierte Meta-Tags. Eine höhere Conversion-Rate – vor allem auf mobilen Endgeräten – trägt ebenfalls dazu bei. Technologien wie Progressive Web Apps (PWA) können dabei helfen, eine mobile App-ähnliche Erfahrung zu schaffen, mit Funktionen wie Offline-Modus, Push-Benachrichtigungen und schnellem Seitenaufbau.

SMARTe Ziele setzen

Es ist wichtig, die Erwartungen an die neue Plattform klar zu formulieren. Hierbei hilft das Konzept der SMARTen Ziele:

- **Spezifisch:** Das Ziel sollte eindeutig und klar definiert sein. Beispiel: Reduktion der monatlichen Lizenzgebühren.
- **Messbar:** Die Zielerreichung sollte anhand konkreter Kennzahlen (Key Performance Indicators, oder "KPI") nachvollziehbar sein, z. B. durch Google Analytics oder eine Auswertung im E-Commerce Admin Bereich.
- **Attraktiv:** Das Ziel sollte motivierend wirken, etwa indem es Kunden- oder Mitarbeiterprobleme löst.
- **Relevant:** Es muss im Einklang mit der Unternehmensstrategie stehen, z. B. eine geplante Marktexpansion unterstützen.
- **Terminiert:** Das Ziel sollte an einen klaren Zeitrahmen gekoppelt sein.

Im Folgenden sind einige SMARTe Ziele genannt, die oft im Zuge eines Replatformings genannt werden:

- Kostenoptimierung:
- Die monatlichen TCO inkl. Hosting, Softwarelizenzen und Agenturkosten sind innerhalb von sechs Monaten nach dem Replatforming durchschnittlich um 20 % niedriger
- Skalierbarkeit und Flexibilität
- Die Anzahl der E-Mail Kundensupport Anfragen ist nach dem Replatforming um 40 % niedriger
- Die Erstellung eines neuen Blogbeitrags dauert nach dem Replatforming 20 Minuten weniger

- Das Liveschalten eines neuen Produktes nimmt 10 Minuten weniger Zeit in Anspruch
- Umsatzsteigerung
- Die Conversion Rate in Google Analytics ist innerhalb von einem halben Jahr nach dem Replatforming um 20 % Prozentpunkte besser
- Der Sichtbarkeitsindex im SEO Crawling Tool ist innerhalb von einem halben Jahr nach dem Replatforming um 10 % Prozentpunkte besser

Bevor SMARTe Ziele definieren zu können, sollten zunächst die relevante KPIs zur Messung identifiziert werden. Im folgenden werden einige KPIs genannt, die oft in der Praxis zum Tragen kommen.

Wirtschaftliche KPIs (Messung von Umsatz, Rentabilität und Wachstum)

KPIS	BESCHREIBUNG	QUELLE
Conversion Rate	Anteil der Besucher, die kaufen	Analytics
Average Order Value (AOV)	Durchschnittlicher Warenkorbwert	Analytics
Customer Lifetime Value (CLV)	Langfristiger Wert eines Kunden	ERP, CRM, BI
Customer Acquisition Cost (CAC)	Kosten, um einen neuen Kunden zu gewinnen	Marketing Tools[o]
Gross Margin	Differenz zwischen Umsatz und Wareneinsatz	ERP oder BI
Revenue / Umsatz	Gesamtumsatz (nach Regionen/Kanälen segmentierbar)	Analytics, ERP, BI
Shopping Cart Abandonment Rate	Anteil abgebrochener Käufe	Analytics

Technische KPIs (Messung der Technische Stabilität, Performance und Nutzererfahrung)

KPIS	BESCHREIBUNG	QUELLE
Page Load Time	Zeit, bis eine Seite vollständig geladen ist	Website Performance Monitoring[1]
Time to First Byte (TTFB)	Serverantwortzeit	Website Uptime Monitoring[2]
Server Uptime	Erreichbarkeit des Shops	Website Uptime Monitoring
Fehlerrate / Error Rate	Anteil technischer Fehler (z.B. JavaScript Fehler)	Fehler Monitoring[3]
Mobile Performance Score	Mobile Ladezeit / UX	Website Performance Monitoring
API Response Time	API Response Time	Website Uptime Monitoring
SEO Page Speed Score (z. B. Lighthouse)	Technischer SEO-Score	SEO Crawling[4]

Nutzerzentrierte KPIs (Messung der Nutzererfahrung)

KPIS	BESCHREIBUNG	QUELLE
Net Promoter Score (NPS)	Weiterempfehlung durch Kunden	Umfrage Tool[5]
Bounce Rate	Besucher, die nach 1 Seite abspringen	Analytics
Pages per Session	Durchschnittlich besuchte Seiten	Analytics
Average Session Duration	Verweildauer auf der Seite	Analytics
Repeat Purchase Rate	Anteil wiederkehrender Käufer	CRM, BI

Support-Tickets / Beschwerden	Anfragen & Beschwerden	Umfrage Tool

Prozessbezogene KPIs (Messung der Leistung von logistischen Prozessen)

KPIS	BESCHREIBUNG	QUELLE
Order Fulfillment Time	Zeit von Bestellung bis Versand	ERP
Retourenquote	Anteil Rücksendungen	ERP

Strategische KPIs (Messung der Qualität der E-commerce Software)

KPIS	BESCHREIBUNG	QUELLE
Time to Market neuer Features	Entwicklungsdauer neuer Releases	Projektmanagement Tool[6]
Entwicklungsaufwand (Velocity, Story Points)	Notwendige Stunden bzw. Budget für das Liefern von Funtktionen	Projektmanagement Tool
Anzahl Bugs nach Releases	Stabilität der neuen Plattform	Projektmanagement Tool, Fehler Monitoring

✓ CHECKLISTE ZIELE DEFINIEREN

- Es wurden die Aspekte Umsatzsteigerung, Kostenoptimerung und Skalierung bzw. Flexibilität zum Definieren von Zielen beleuchtet.
- Es wurden relevante KPIs für die Definition von Zielen in den jeweiligen Quellsystemen identifiziert.
- Die Ziele wurden im Sinne der SMART Methode definiert.

2.3 Alternativen zum Replatforming

Je nachdem, welche Ziele erreicht werden sollen und welche Beschwerden es zu mindern gilt, muss ein Replatforming nicht immer die einzige Lösung sein. Die heutige E-Commerce-Landschaft bietet eine Vielzahl an Partnern und Softwarelösungen, die auf spezifische Herausforderungen zugeschnitten sind.

Ein **neues Theme** oder ein überarbeitetes Storefront kann beispielsweise dazu beitragen, die Ladezeiten zu verkürzen, die SEO-Performance zu verbessern und den Nutzern eine angenehmere User Experience (UX) zu bieten. Viele gängige Shop-Plattformen verfügen über App-Stores oder Marktplätze für Erweiterungen, in denen fertige Themes erhältlich sind. Diese können oft ohne großen Entwickleraufwand an die Corporate Identity angepasst werden. Für exotische Shopsysteme oder Eigenentwicklungen sind hingegen häufig individuelle Lösungen nötig, die von externen Dienstleistern oder internen Entwicklern umgesetzt werden müssen. Solche Maßnahmen stellen jedoch lediglich einen Frontend-Relaunch dar und sind kein vollständiges Replatforming, da das Backend unverändert bleibt.

Im Bereich der Performance-Optimierung gibt es ebenfalls zahlreiche Ansätze. Ein **Content Delivery Network (CDN)** kann dabei helfen, statische Inhalte wie HTML, CSS, JavaScript, Produktbilder oder Videos schneller auszuliefern, indem sie auf einem globalen Netzwerk verteilt werden. Nutzer erhalten die Inhalte dann von geografisch nahegelegenen Servern, was die Ladezeiten verkürzt und die SEO-Performance steigert. Ergänzend dazu können **Caching-Services** eingesetzt werden, die häufig genutzte Inhalte auf einem Server zwischenspeichern. Dadurch werden Serveranfragen reduziert, die Ladezeiten beschleunigt und Kosten durch eine effizientere Ressourcennutzung gesenkt.

Einige Anbieter wie Cloudflare[7] kombinieren CDN- und Caching-Lösungen mit zusätzlichen Funktionen. Dazu gehören die automatische Konvertierung von Bildern in moderne Formate wie WEBP oder AVIF sowie Schutzmaßnahmen gegen DDoS-Angriffe.

Zur Verbesserung der Conversion Rate (CRO) und der UX stehen zahlreiche Tools zur Verfügung. Ein **Chatbot** kann Nutzern schnell und unkompliziert bei Fragen helfen. Mithilfe künstlicher Intelligenz lassen sich viele Anliegen automatisiert bearbeiten. Bei komplexeren Themen oder rechtlichen Anforderungen, beispielsweise im Arzneimittelverkauf, ist jedoch der Einsatz menschlicher Mitarbeiter sinnvoll. Ein

weiteres nützliches Tool ist eine **externe Such-Engine**, die bessere Ergebnisse als die Standardsuche eines Shops liefert. Diese kann beispielsweise Synonyme und Tippfehler berücksichtigen oder relevante Suchvorschläge anbieten.

Zusätzlich können **Recommendation Engines** eingesetzt werden, um personalisierte Produktempfehlungen zu generieren. Diese analysieren das Nutzerverhalten und schlagen passende Produkte vor – beispielsweise „andere Kunden kauften auch“ im Warenkorb. Eine **Personalization Engine** geht noch einen Schritt weiter, indem sie Inhalte und Angebote basierend auf Nutzerdaten anpasst. So könnten etwa in einem Tierfutter-Shop auf der Startseite personalisierte Inhalte für Hunde- oder Katzenbesitzer angezeigt werden. Chatbots, Recommendation, Such- und Personalization Engines lassen sich oft unkompliziert per JavaScript in das Storefront einbetten.

Auch im Check-out gibt es Optimierungsmöglichkeiten. **Express-Checkout-Lösungen** wie Amazon Pay oder PayPal Express verkürzen den Bestellvorgang, indem sie Zahlungs- und Adressdaten automatisch ausfüllen. Wallet-Lösungen wie Apple Pay oder Google Pay bieten ähnliche Vorteile.

Im Bereich der Kundenbindung können Tools wie **Marketing Automation Tools** genutzt werden, um personalisierte E-Mails, SMS oder Push-Benachrichtigungen zu verschicken. So lassen sich beispielsweise Newsletter mit relevanten Aktionen automatisiert erstellen. Auch Bonusprogramme und Treuepunktesysteme können eingesetzt werden, um Nutzer zu wiederholten Käufen zu motivieren und langfristig an den Shop zu binden.

Ein externes **Content-Management-System (CMS)** kann ebenfalls eine wirkungsvolle Ergänzung sein – insbesondere, wenn die Anforderungen im Bereich Content-Organisation und redaktionelle Prozesse steigen. Gerade bei international ausgerichteten Shops lässt sich mit einem dedizierten CMS Content gezielt nach Sprache, Region oder Zielgruppe organisieren und ausspielen. Für große Teams mit mehreren Redakteuren bietet ein CMS zudem klare Vorteile: Inhalte lassen sich rollenbasiert verwalten, etwa mit getrennten Zuständigkeiten für Redaktion, Review und Freigabe. Durch definierte Workflows kann ein Redakteur Inhalte vorbereiten, ein anderer prüfen und schließlich veröffentlichen – unterstützt durch Vorschaufunktionen und zeitgesteuerte Veröffentlichung, etwa im Rahmen geplanter Marketingkampagnen. Moderne Headless-CMS lassen sich dabei per API flexibel in

bestehende Shop-Infrastrukturen integrieren, ohne tief in das Backend einzugreifen. Ein gutes CMS ist zudem intuitiv bedienbar und erlaubt Content-Erstellung ohne Programmierkenntnisse, teilweise sogar mit vordefinierten Vorlagen.

Schließlich bietet eine **PIM-Software (Product Information Management)** umfangreiche Unterstützung bei der Verwaltung und Pflege von Produktdaten. Sie ermöglicht effiziente Massenänderungen, unterstützt bei der Internationalisierung und optimiert die Datenbereitstellung für verschiedene Verkaufskanäle. Dadurch können Kosten gesenkt und die Time-to-Market verkürzt werden.

Diese Alternativen zum Replatforming zeigen, dass es oft möglich ist, spezifische Probleme gezielt anzugehen, ohne den gesamten Shop zu erneuern. Welche Lösung letztlich die richtige ist, hängt von den individuellen Anforderungen und Zielen des E-Commerce Owners ab.

✓ CHECKLISTE ALTERNATIVE ZUM REPLATFORMING

- Es wurde geprüft, ob die SEO Performance, Nutzererfahrung oder Ladezeiten durch ein neues Storefront (oder Theme) verbessert werden können.
- Es wurde geprüft, ob die Ladezeiten durch die Einführung eines CDN oder Caching Servers verbessert werden können.
- Es wurde geprüft, ob die Conversion Rate und die Nutzererfahrung durch einen Chatbot oder eine Such-, Recommendation- oder Personalization-Engine verbessert werden können.
- Es wurde geprüft, ob die Conversion Rate im Checkout durch einen Express Checkout Anbieter verbessert werden kann.
- Es wurde geprüft, ob die Kundenbindung durch ein Marketing Automation Tool verbessert werden kann.
- Es wurde geprüft, ob ein externes CMS ein schnelleres Veröffentlichen von Content sowie die Internationalisierung fördern kann,
- Es wurde geprüft, ob die Produktdatenqualität durch ein PIM verbessert werden kann.

2.4 Funktionale Anforderungen definieren

Die Definition funktionaler Anforderungen ist einer der strategisch wichtigsten Schritte im Replatforming-Projekt. Sie bilden den zentralen Ausgangspunkt für die Planung und Durchführung des Projekts und entscheiden maßgeblich über dessen Erfolg. Werden die funktionalen Anforderungen falsch oder unvollständig formuliert, kann das Projekt scheitern, noch bevor es richtig begonnen hat. Ressourcen wie Zeit, Budget und Arbeitskraft werden dadurch ineffizient genutzt. Es ist daher essenziell, sich gezielt auf die Verfassung präziser und gut durchdachter Anforderungen zu konzentrieren.

Falls im Unternehmen keine ausreichende Erfahrung oder Ressourcen für diesen Prozess vorhanden sind, kann eine externe Beratung hinzugezogen werden. Experten mit Erfahrung im E-Commerce können nicht nur den Prozess strukturieren, sondern auch wertvolle Perspektiven aus anderen Projekten einbringen.

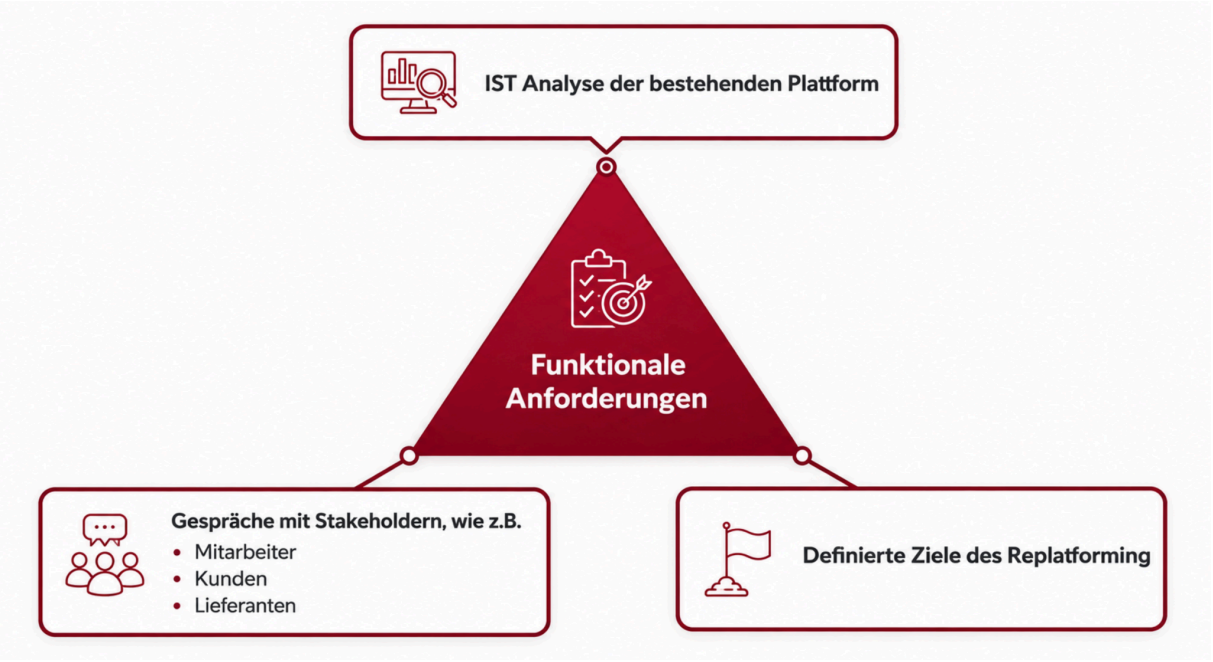
Der erste Schritt zur Definition funktionaler Anforderungen ist die enge Zusammenarbeit mit den Stakeholdern und Partnern des Projekts. Diese haben sich schon in dem vorherigen Kapitel bemerkbar gemacht und Signale für das Replatforming gegeben. Diese Gespräche sollten alle relevanten Stakeholder einbeziehen, darunter:

- **Externe Partner**: Aktuelle Agenturen, Marketing-Dienstleister, ERP-Partner oder Lieferanten.
- **Interne Teams**: Shop-Administratoren, Produktdatenmanager, Content-Manager und andere Mitarbeiter, die regelmäßig mit der Plattform arbeiten.
- **Kunden**: Durch Feedback, Umfragen oder Reviews lassen sich wertvolle Einblicke in die Schwächen des aktuellen Systems gewinnen.

Extrem hilfreich an dieser Stelle ist der Abgleich mit der IST Analyse des bestehenden Systems. Anhand Analyse der aktuellen Plattform und den Gesprächen mit relevanten Stakeholdern kann man erfahren,

- welche Funktionen bereits oft genutzt sind und daher wichtig für die neue E-Commerce Plattform
- welche Funktionen weniger hilfreich sind, und daher kein Bestandteil der funktionalen Anforderungen sein sollen und schließlich
- welche Funktionen noch fehlen.

In jedem Fall sollten die definierten Ziele aus dem vorherigen Kapitel in die Anforderungen einbezogen werden. Im Wesentlichen werden also die funktionalen Anforderungen aus Gesprächen mit Stakeholdern, den definierten Zielen, sowie der IST Analyse des bestehenden Systems gebildet.



Des Weiteren sollte der Verfasser der funktionalen Anforderungen hier auf die Zielgruppenanalysen, das Business-Canvas-Modell sowie die Wettbewerbsanalysen aus den Folgekapiteln zurückgreifen. Inspiration kann auch durch den Einsatz von KI-Tools gewonnen werden, die als Sparringspartner für neue Ideen oder zur Überprüfung von Anforderungen dienen.

Neben der Definition der richtigen Anforderungen ist es ebenso wichtig, diese richtig zu formulieren. Diese sollten in einer **standardisierten Struktur** geschrieben werden, d.h. Anforderungen sollten einheitlich und systematisch erfasst werden, etwa in Tabellen- oder gegliederter Textform. Eine Gruppierung nach Funktionsbereichen, wie Produktdatenmanagement, CMS, Preise und Rabatte, Adminverwaltung oder Integrationen, sorgt für zusätzliche Übersichtlichkeit.

Es kann zudem hilfreich sein, an dieser Stelle mit **User Stories** zu arbeiten. Dies ist eine praxisnahe Methode zur Formulierung von Anforderungen: Beispiel: „Als Kunde möchte ich Produkte nach Preis, Farbe und Marke filtern können, um die für mich relevanten Artikel schneller zu finden.“ Mehr dazu in den späteren Kapiteln bzgl. dem Product Backlog .

Wichtig ist hier bereits die Einordnung von Anforderungen in **Must-haves und Nice-to-haves**: Must-haves sind essenziell für den Betrieb des Projekts, während Nice-to-haves wünschenswerte, aber nicht zwingend notwendige Funktionen beschreiben. Must-haves müssen also in jedem Teil fester Bestandteil der des ersten Releases sein. Dabei kann gerne auf das Konzept des **"Minumum Viable Product" (MVP)** zurückgegriffen werden: Ein MVP für ein E-Commerce-Replatforming umfasst die Kernfunktionen, die benötigt werden, um die Plattform live zu schalten und den Grundbetrieb des Shops sicherzustellen. Es handelt sich um eine schlanke Version der Plattform, die die wichtigsten „Must-have"-Anforderungen abdeckt, aber auf weniger dringliche oder komplexe Funktionen verzichtet und diese in weiteren Releases geliefert werden. Die Konzepte MVP sowie Must-haves und Nice-to-haves werden im Abschnitt zum Product Backlog weiter beschrieben.

Schließlich sollten die Anforderungen **klar, eindeutig und präzise** geschrieben sein, um Missverständnisse zu vermeiden.

Ein schlechtes Beispiel: Die Produktdetailseite sollte SEO-friendly sein

Ein gutes Beispiel: Die Produktdetailseite enthält selbst referenzierende Canonical Tags, Hreflang Tags, strukturierte Daten im schema.org Markup (inkl. aller Standardfelder wie SKU, Name, Preis, Verfügbarkeit, ...), Alt Tags für Bilder, open-graph Tags, Meta-Beschreibung und -Titel

Ein weiteres schlechtes Beispiel: Es gibt eine intuitive Suche

Ein weiteres gutes Beispiel: Im Header Bereich gibt es eine Suche, die ein Dropdown mit Produkten und Kategorien als Ergebnisse anzeigt. Zum Suchbegriff werden Produkte (inkl. Preis, Name und Bild), Kategorien, verwandte Suchbegriffe und Content-Seiten gezeigt. Durch die Enter Taste gelangt der Nutzer auf eine Suchergebnisseite mit einer Sortierung und Filtern.

Dem Autor der funktionalen Anforderungen sollte klar sein, dass dies keine einmalige Aufgabe, sondern ein iterativer Prozess ist. Sie sollten im Laufe des Projekts regelmäßig überprüft und erweitert werden, um sicherzustellen, dass sie den aktuellen Erkenntnissen und Bedürfnissen entsprechen. Dieser iterative Ansatz erlaubt es, strategische Anpassungen vorzunehmen, ohne den Projektfortschritt zu gefährden.

✓ CHECKLISTE FUNKTIONALE ANFORDERUNGEN DEFINIEREN

- Es wurden Anforderungen von allen relevanten Stakeholdern (intern, extern, Kunden) eingeholt.
- Es hat ein Abgleich mit der aktuellen E-Commerce Plattform stattgefunden und es wurde kritisch hinterfragt, welche Funktionen fehlen, welche Funktionen übernommen werden und welche Funktionen verabschiedet werden können.
- Die funktionalen Anforderungen fördern das Erreichen der definierten Ziele und berücksichtigen Erkenntnisse aus der Stakeholder-, Wettbewerbs- sowie Zielgruppenanalyse.
- Die Anforderungen sind präzise und eindeutig formuliert, und bestenfalls in Must-haves und Nice-to-haves kategorisiert.

2.5 Die neue E-Commerce-Plattform wählen

Nachdem sich der E-Commerce Owner für ein Replatforming entschieden und Anforderungen definiert hat, kann er sich Gedanken zur Wahl der zukünftigen E-Commerce-Plattform machen. Bevor eine Wahl der Technologie, oder dem E-Commerce System, getroffen wird, sollte das Fundament in Form der Infrastruktur und der Architektur definiert werden. Im Folgenden werden die gängigen architektonischen Ansätze sowie deren Vor- und Nachteile beschrieben.

Das richtige technische Fundament

Die Frontend-Architektur: Coupled oder Headless?

Bei einem **Headless-Ansatz** ist das Storefront vom Backend (der E-Commerce Geschäftslogik) entkoppelt. Die Kommunikation zwischen Beiden erfolgt über APIs. Bei jeglicher Interaktion der Nutzer im Frontend, wie z.B. ein Produkt in den Warenkorb legen, in das Kundenkonto einloggen, in den Check-out gehen, usw. erfolgt eine Kommunikation mit dem Backend per API. Ein E-Commerce System, das bspw. Magento 2 als Backend nutzt, könnte ein völlig anderes Storefront (z. B. auf Basis von React oder Vue.js) verwenden, das die Daten per Magento-API abruft.

Dieser Ansatz hat einige Vorteile. Das Frontend kann unabhängig vom Backend entwickelt werden, d.h. das Deployment für eine Anpassung des Frontends kann separat vom Backend Deployment getätigt werden. Die Innovation ist damit schneller, da Änderungen am Design oder neue Funktionen im Frontend ohne Backend Deployments erfolgen können. Zum anderen kann das E-Commerce Backend Inhalte für verschiedene Kanäle per API bereitstellen (z. B. Website, mobile App, IoT), womit der E-Commerce Owner flexibler auf Kunden- und Marktbedürfnisse reagieren kann.

Ein Headless Storefront birgt allerdings auch einige Risiken und trägt einige Nachteile. So ist der Entwicklungsaufwand in der Regel höher, da zwei separate Systeme verwaltet und instandgehalten werden müssen. Die notwendigen zusätzlichen Ressourcen und das Know-how für API-Entwicklung und Frontend-Technologien führen zu mehr Kosten. Zudem steigt aufgrund der engen API Kommunikation zwischen Frontend und Backend die technische Komplexität. Eine ineffiziente API-Integration kann die Ladezeiten verlängern sowie die SEO Performance verschlechtern.

Bei einem **Coupled Ansatz** ist das Frontend mit dem Backend eng miteinander verbunden, meistens per PHP Funktionen, JavaScript und AJAX Calls. Änderungen im Frontend erfordern häufig auch Anpassungen im Backend und es gibt in der Regel ein gemeinsames Deployment. Dies ist auch der klassische Magento 2 Ansatz, bei dem das Standard-Frontend direkt mit dem Magento-Backend verbunden ist (z.B. für die Produktdarstellung).

In dem klassischen Coupled Ansatz existieren für gängige E-Commerce Systeme oft Out-of-the-box-Lösungen (etwa Extensions), bei denen Frontend und Backend nahtlos zusammenarbeiten. Im Vergleich zu Headless sind die Implementierung schneller und technische Anforderungen geringer. Als Folge sind die Projektkosten im Allgemeinen geringer.

Jedoch genießt das Coupled Frontend weniger Flexibilität, da Anpassungen am Frontend sind stark vom Backend abhängig sind. Zudem lassen sich moderne Frontend-Technologien, etwa moderne JavaScript Bibliotheken, schwieriger in ein Coupled Frontend integrieren und aufgrund der Backend-Abhängigkeit kann die Skalierung bei wachsendem Geschäftsumfang an seine Grenzen stoßen.

Zusammenfassen lässt sich sagen, dass sowohl Coupled als auch Headless Frontend-Architekturen ihre Daseinsberechtigung haben und es vollkommen vom Geschäftsmodell des E-Commerce Owners abhängt, welcher Ansatz die bessere Wahl ist. Größere und komplexere (Omnichannel) Business-Modelle mit mehreren Kanälen (z.B. E-Commerce Storefront als Website, iOS App, Android App, ...), mit größeren Projektbudgets werden eher zu einem Headless Ansatz tendieren. Hingegen werden kleinere Händler mit einem weniger komplexen Geschäftsmodell in der Regel mit einem Coupled Ansatz sehr gut und kosteneffizient bedient.

Die Backend-Architektur: Ein Monolith, Microservices, oder lieber etwas dazwischen (Composable Commerce)?

Beim **Microservices-Ansatz** wird die E-Commerce-Anwendung in kleinere, unabhängige Dienste unterteilt, die jeweils eine spezifische Aufgabe übernehmen, wie beispielsweise Produktkatalog, Bestandsmanagement oder Zahlung. Die Kommunikation zwischen diesen Microservices und den übrigen E-Commerce-Anwendungen erfolgt in der Regel über APIs. Ein Beispiel im Kontext von Magento 2 wäre, eine eigenständige Microservice-Anwendung für den Checkout zu entwickeln, die per API mit dem Magento-System verbunden ist.

Der größte Vorteil dieses Ansatzes liegt in der hohen Skalierbarkeit: Jeder Service kann unabhängig weiterentwickelt oder angepasst werden. Damit verbunden ist auch eine enorme Flexibilität, da neue Dienste problemlos hinzugefügt werden können, ohne dass bestehende Komponenten gestört werden. Darüber hinaus sorgen die getrennten Dienste für eine höhere Ausfallsicherheit, da ein Fehler in einem Service nicht das gesamte System beeinträchtigt.

Allerdings bringt der Microservices-Ansatz auch Nachteile mit sich. Er ist oft mit höheren Kosten verbunden, da zusätzliche Aufwände für Entwicklung, Infrastruktur, Monitoring und Debugging anfallen. Die Komplexität steigt ebenfalls, da die Koordination der API-Kommunikation zwischen den Services anspruchsvoll ist. Schließlich können die Hosting- und Infrastrukturkosten aufgrund der Vielzahl an Diensten deutlich höher ausfallen.

Ein **monolithisches System** hingegen integriert alle Funktionen – vom Frontend und Backend bis zur Datenbank – in einer eng verknüpften Codebasis. Bei einer Standardinstallation von Magento 2 sind beispielsweise Kernfunktionen wie Suche, Filterung, Produktmanagement und Check-out in einem einzigen System vereint.

Der Monolith zeichnet sich durch einfache Entwicklung und Verwaltung aus, da nur eine einzige Codebasis gepflegt werden muss. Extensions und Anpassungen können in vielen Fällen mit relativ geringem Aufwand integriert werden, was eine schnellere Time-to-Market ermöglicht. Monolithische Systeme sind weniger ressourcenintensiv und verursachen geringere Hosting- und Infrastrukturkosten als Microservices.

Jedoch hat auch der Monolith Schwächen: Die Flexibilität ist begrenzt, was größere Anpassungen und Erweiterungen aufwendig und teuer macht. Bei steigendem Traffic oder erweiterten Anforderungen kann die Skalierbarkeit schnell an ihre Grenzen stoßen. Zusätzlich führt die Ansammlung von technischen Schulden – insbesondere bei umfangreichen Individualisierungen – zu erhöhten Wartungs- und Updatekosten.

Ein **Composable-Commerce-Ansatz** stellt einen Mittelweg zwischen Microservices und Monolithen dar. Hier werden spezialisierte Systeme modular miteinander kombiniert, um eine maßgeschneiderte Lösung zu schaffen. Beispielsweise könnte in Magento 2 die Produktverwaltung durch ein PIM-System wie Akeneo [8]ergänzt, die Content-Pflege durch ein CMS wie Storyblok[9] übernommen und die native Suche durch ein Such-Tool wie Algolia[10] ersetzt werden.

Der Vorteil dieses Ansatzes liegt im Best-of-Breed-Prinzip, bei dem aus jeder Funktion das Beste herausgeholt wird, indem spezialisierte Systeme eingebunden werden. Dies bietet eine höhere Anpassbarkeit und Zukunftssicherheit, da einzelne Komponenten bei Bedarf ausgetauscht werden können, ohne das gesamte System zu beeinträchtigen.

Jedoch bringt Composable Commerce eine erhöhte Komplexität mit sich, da die Integration und Koordination der verschiedenen Systeme anspruchsvoll ist. Hinzu kommen höhere Kosten, sowohl durch initiale Investitionen für die Entwicklung und Lizenzen als auch durch laufende Wartung und Instandhaltung der Integrationen. Zusätzlich entstehen Verwaltungsaufwände durch die Zusammenarbeit mit verschiedenen Anbietern. Technisch gesehen müssen Abhängigkeiten und Kompatibilitäten zwischen den Systemen sorgfältig berücksichtigt werden.

Die Wahl der Backend-Architektur hängt somit von den Anforderungen des Projekts ab. Während ein Monolith eine überschaubare Komplexität und geringere Kosten bietet, sind die Flexibilität und Skalierbarkeit eingeschränkt. Microservices erlauben eine hohe Anpassungsfähigkeit und Skalierung, gehen aber mit mehr Kosten und einem höheren Koordinationsaufwand einher. Composable Commerce bietet einen ausbalancierten Ansatz, indem es die Vorteile eines Monolithen mit spezialisierten Systemen kombiniert, ist jedoch komplexer und oft kostenintensiver. Die richtige Wahl der Architektur ist letztlich eine Abwägung zwischen Flexibilität, Kosten und technischer Machbarkeit.

Das Deployment-Modell: SAAS oder on-premise, oder lieber etwas dazwischen (PAAS)?

Beim Modell **Software as a Service (SaaS)** wird die E-Commerce-Software von einem externen Anbieter bereitgestellt und vollständig gehostet. Der Anbieter stellt die gesamte Infrastruktur zur Verfügung, inklusive regelmäßiger Software-Updates. Die Nutzung erfolgt über eine monatliche oder jährliche Gebühr. Ein prominentes Beispiel für eine SaaS-E-Commerce-Lösung ist Shopify. Hier hat der E-Commerce Owner die Möglichkeit, die gelieferten Features individuell zu konfigurieren, allerdings ohne direkten Zugriff auf den Quellcode. Anpassungen an der Geschäftslogik erfordern entweder den Einsatz von Extensions oder eine externe Anwendung, die als Microservice per API angebunden wird.

Der Vorteil von SaaS liegt vor allem in der geringen technischen Komplexität. Der E-Commerce Owner muss sich nicht um Hosting, Servermanagement oder Updates kümmern, da diese Aufgaben vollständig vom Anbieter übernommen werden. Dadurch ist die Implementierung schnell und die Kosten für die Nutzung sind transparent kalkulierbar, was eine klare Kostenkontrolle ermöglicht.

Allerdings bietet SaaS nur eine eingeschränkte Flexibilität, da der E-Commerce Owner keine direkten Änderungen am Code vornehmen kann. Zudem besteht eine Abhängigkeit vom Anbieter, da neue Features und Updates nur durch diesen bereitgestellt werden. Fraglich bleibt auch die Datenhoheit, da sämtliche Daten auf den Servern des Anbieters gespeichert sind.

Bei der **On-Premise-Lösung** wird die E-Commerce Software auf eigenen Servern oder bei einem Cloud-Anbieter wie Amazon Web Services, Google Cloud oder Microsoft Azure gehostet. Hier trägt der E-Commerce Owner die volle Verantwortung für die Konfiguration, Verwaltung und Wartung der Infrastruktur, inklusive Updates, Backups und Datensicherheit. Bekannte Beispiele für On-Premise-Lösungen sind die Open-Source-Versionen von Shopware 6 und Magento 2.

Der Hauptvorteil von On-Premise liegt in der vollständigen Kontrolle über die Infrastruktur. Anpassungen können individuell vorgenommen werden, und es besteht keine Abhängigkeit von einem externen Anbieter. Dies ermöglicht maximale Flexibilität, insbesondere für E-Commerce Systeme mit spezifischen Anforderungen.

Nachteilig ist jedoch der hohe Initialaufwand für die Anschaffung und Einrichtung von Servern sowie die laufenden Kosten für Betrieb und Wartung. On-Premise erfordert zudem spezialisiertes IT-Personal, um die Infrastruktur zu verwalten. Bei Cloud-Anbietern mit Pay-as-you-go-Modellen können die Hostingkosten schwer kalkulierbar sein. Auch das Management von Traffic-Spitzen, etwa an saisonalen Peaks wie dem Black Friday, kann herausfordernd sein, wenn keine automatische Skalierung vorgesehen ist.

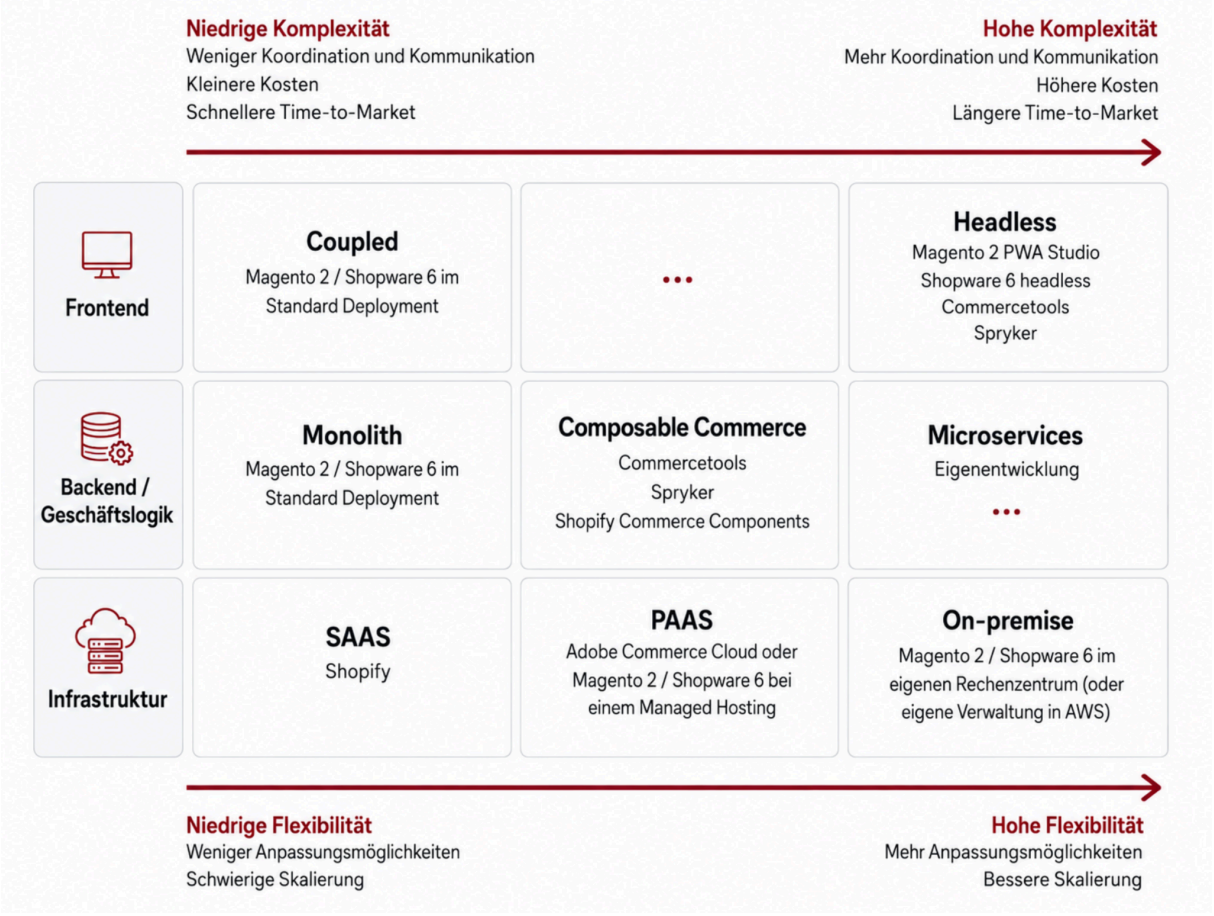
Das Modell **Platform as a Service (PaaS)** bietet einen Mittelweg zwischen SaaS und On-Premise. Hier stellt der Anbieter eine Plattform bereit, auf der der E-Commerce Owner eigene Software installieren und verwalten kann. Die zugrunde liegende technische Infrastruktur – einschließlich Server, Datenbanken und Caching – wird vom Anbieter bereitgestellt. Der E-Commerce Owner hat jedoch Zugriff auf den

Quellcode und kann die Geschäftslogik nach Bedarf anpassen. Ein bekanntes Beispiel für PaaS ist die Adobe Commerce Cloud, die lizenzpflichtige Variante von Magento 2.

PaaS bietet mehr Flexibilität als SaaS, da der E-Commerce Owner die Geschäftslogik individuell gestalten kann. Gleichzeitig entfällt der Aufwand für die Einrichtung und Verwaltung der Infrastruktur, was den technischen Verwaltungsaufwand im Vergleich zu On-Premise reduziert.

Allerdings fallen die Kosten für PaaS in der Regel höher aus als bei SaaS, insbesondere durch die Lizenzkosten und den potenziellen Bedarf an Entwicklerkapazitäten für individuelle Anpassungen. Zudem kann auch hier eine gewisse Abhängigkeit vom Anbieter entstehen, vorwiegend bei der Skalierung der Infrastruktur.

Zusammenfassend kann man festhalten, dass **SaaS** geringe Verwaltungsaufwände und eine schnelle Time-to-Market bietet, aber weniger flexibel ist. **On-Premise** hingegen bietet maximale Kontrolle und Flexibilität, erfordert jedoch erhebliche Investitionen und spezialisiertes IT-Know-how. **PaaS** kombiniert Elemente beider Modelle, mit einer guten Balance zwischen Flexibilität und reduziertem Infrastrukturaufwand, jedoch zu höheren Kosten.



Eine SaaS-Lösung, sowie ein coupled Frontend und eine monolithische Architektur, eignen sich vor allem für kleinere Projekte. Diese zeichnen sich oft durch ein geringeres Budget, niedrigere Umsätze sowie überschaubare Anforderungen und ein einfaches Geschäftsmodell aus – ein typisches Szenario für Start-ups. Im Gegensatz dazu tendieren größere Händler, die bereits wirtschaftlich erfolgreich sind und ihr Geschäft skalieren oder erweitern möchten, eher zu einer Microservices- oder Headless-Architektur. Solche Ansätze bieten die notwendige Flexibilität und Skalierbarkeit, um komplexe Anforderungen und ein anspruchsvolles Geschäftsmodell zu bewältigen.

Die zentrale Frage bei der Wahl der Architektur und des Deployment-Modells lautet: **Passe ich mein Geschäft an die Software an, oder passe ich die Software an mein Geschäft an?**

Um die richtige Entscheidung zu treffen, ist es essenziell, die spezifischen Anforderungen und Ziele des Projekts sorgfältig zu analysieren und mit den jeweiligen Stärken und Schwächen der verschiedenen Ansätze abzugleichen. Nur so kann das technologische Fundament gewählt werden, das sowohl den aktuellen Bedürfnissen als auch den langfristigen Wachstumsplänen des E-Commerce Owners gerecht wird. Genau dies wird im nächsten Abschnitt beschrieben.

Kriterien zur Wahl der richtigen E-Commerce Plattform

Die Auswahl des richtigen E-Commerce Systems hängt entscheidend davon ab, welches **technologische Fundament** am besten zu den Zielen und Anforderungen des E-Commerce Owners passt. Hierbei gilt es zu klären, ob eine SaaS-, On-Premise-, Microservices-, Composable Commerce-, Monolith-, Coupled- oder Headless-Architektur die strategischen und operativen Anforderungen am besten unterstützt. Die gewählte Architektur sollte optimal zur geplanten IT-Strategie des Unternehmens passen.

Ein ebenso wichtiger Faktor ist die Fähigkeit des Systems, die eigenen **funktionalen Anforderungen** abzudecken. Plattformen mit umfangreichen Out-of-the-Box-Funktionen eignen sich häufig für Geschäftsmodelle mit standardisierten Anforderungen. Für komplexere oder spezifische Anforderungen hingegen sind individuelle Anpassungen oder der Einsatz von Drittanbieter-Extensions erforderlich, um das gewünschte Ergebnis zu erzielen.

Neben technischen Überlegungen spielen auch andere Kriterien eine wichtige Rolle. Die **Verfügbarkeit von Agenturen und Entwicklern** mit Erfahrung in der jeweiligen Plattform ist essenziell, um eine reibungslose Implementierung und Wartung zu gewährleisten. Ein breites Netzwerk an Partnern sorgt dafür, dass der E-Commerce Owner bei Bedarf unkompliziert zwischen Agenturen wechseln oder zusätzliche Entwickler einbinden kann. Zudem gibt die **Reife des Ökosystems** – etwa die Anzahl an Extensions, Diskussionsbeiträge auf Stack Overflow oder GitHub-Aktivitäten – einen Hinweis darauf, wie gut die Plattform unterstützt wird. Plattformen mit einer etablierten Community und langjähriger Marktpräsenz bieten in der Regel schnellere Lösungen für Bugs oder die Einführung neuer Funktionen.

Ein weiterer relevanter Aspekt ist die **Zukunftssicherheit der Plattform**. E-Commerce Owner sollten prüfen, ob der Anbieter regelmäßig Updates bereitstellt und eine klare Roadmap für die Weiterentwicklung kommuniziert. Auch die **Transparenz und Kalkulierbarkeit der Kosten** – etwa Lizenzgebühren, Gebühren für Erweiterungen und laufende Betriebskosten – ist ein zentrales Entscheidungskriterium.

Schließlich spielt die **hauseigene Expertise** des Unternehmens eine wichtige Rolle. Verfügt der E-Commerce Owner über ein internes IT-Team mit Erfahrung im technologischen Stack der neuen E-Commerce-Plattform, kann dies die Implementierung beschleunigen und die Anpassungsfähigkeit des Systems erhöhen. Diese interne Kompetenz stärkt zudem die langfristige Unabhängigkeit von externen Dienstleistern und sorgt für mehr Kontrolle über das Projekt.

✓ CHECKLISTE NEUE E-COMMERCE PLATTFORM WÄHLEN

- Der E-Commerce Owner und die Entscheider sind mit den Architekturparadigmen für Frontend (Coupled vs. Headless), Backend (Monolith, Composable, Microservices) und Deployment (On-premise, PAAS und SAAS) vertraut.
- Die Vor- und Nachteile der jeweiligen Architekturparadigmen wurden hinsichtlich dem Erreichen der definierten Ziele gegenüber gestellt.
- Es wurde geprüft, ob die neue E-Commerce Plattform die funktionalen Anforderungen abgedeckt.
- Es wurden “softe” Faktoren geprüft, wie z.B. Zukunftssicherheit der Plattform (gibt es große Investoren oder Unternehmen hinter der Software?), die Größe und Reife des Ökosystems (Anzahl Extensions und verfügbare Agenturen) sowie die hauseigene Expertise.

2.6 Externe Agentur vs. Inhouse Team

Die Entscheidung, ob die technische Umsetzung eines E-Commerce-Replatformings durch eine externe Agentur oder ein internes Entwicklerteam erfolgen soll, ist ein strategisch wichtiger Schritt. Jede Option hat Vor- und Nachteile, die sich auf die Projektlaufzeit, Kosten und langfristige Flexibilität auswirken. In einigen Fällen kann auch ein hybrider Ansatz sinnvoll sein, um das Beste aus beiden Welten zu kombinieren.

Option 1: Externe Agentur

Der Einsatz einer spezialisierten Agentur bringt viele Vorteile mit sich. Agenturen verfügen oft über tiefgreifende Erfahrung mit spezifischen Softwarelösungen oder modernen Architekturen wie Microservices, Headless oder Composable Commerce. Sie bieten in der Regel ein breiteres Leistungsspektrum, das über die reine Entwicklung hinausgeht und Bereiche wie Testing, Design und Online-Marketing abdeckt, und sich damit Synergien ergeben. Dank ihrer Ressourcen können Agenturen schnell skalieren und die Umsetzung zügig vorantreiben.

Allerdings bringt die Zusammenarbeit mit einer Agentur auch Herausforderungen mit sich. Die Kommunikation mit externen Entwicklern kann schwieriger sein, und die Transparenz im Projektverlauf ist oft geringer, sofern die Kontrolle beim Projektmanagement der Agentur liegt. Zudem können die Kosten langfristig schwer planbar sein: Tagessätze können nach einer Zeit angepasst werden, und Nachbudgetierungen sind nicht ungewöhnlich. Ein weiteres Risiko ist die Abhängigkeit von der Agentur. Sollte die Zusammenarbeit enden, kann wertvolles Know-how verloren gehen, das im eigenen Team nicht aufgebaut wurde.

Option 2: Inhouse-Entwicklung

Ein internes Entwicklerteam bietet klare Vorteile, insbesondere bei der Kontrolle und Management des Projekts. Die eigenen Mitarbeiter sind eher mit den Prozessen und den Zielen des Unternehmens vertraut und können diese nahtlos in die Umsetzung integrieren. Die Kommunikation ist direkter, was gerade bei dringenden Anpassungen oder Problemlösungen von Vorteil ist. Ein weiterer Pluspunkt ist die langfristige Kostenkontrolle, da ausschließlich die Personalkosten anfallen und Tagessätze keine Rolle spielen. Gleichzeitig wird durch das Projekt wertvolles technisches Wissen intern aufgebaut, wodurch zukünftige Abhängigkeiten vermieden werden.

Jedoch gibt es auch Herausforderungen bei der Inhouse-Entwicklung. Das Team ist möglicherweise nicht kurzfristig skalierbar, da neue Mitarbeiter eingestellt und eingearbeitet werden müssen. Auf der anderen Seite können bestehende Ressourcen durch das Projekt gebunden werden, was andere Vorhaben im Unternehmen beeinträchtigt. In einigen Fällen kann dies die Entwicklungszeit verlängern, insbesondere wenn das Team wenig Erfahrung mit der neuen Technologie hat. Da im Unternehmen Erfahrung mit der neuen E-Commerce-Technologie fehlt, können die Fähigkeiten der hauseigenen Entwickler kaum sinnvoll eingeschätzt werden.

Option 3: Hybrid-Modell

Das Hybrid-Modell kombiniert die Vorteile beider Ansätze. Dabei übernimmt eine Agentur die initiale Implementierung bis zum Go-Live und das interne Team die Wartung und Weiterentwicklung. Dieses Modell ist besonders effizient, wenn Verantwortlichkeiten klar definiert werden. Workflows wie Deployment-Prozesse, Code Reviews und Testing müssen sorgfältig abgestimmt werden, um reibungslos zu funktionieren. Der Wissenstransfer spielt hierbei eine zentrale Rolle: Die Agentur sollte Dokumentationen bereitstellen und das interne Team entsprechend schulen.

Durch dieses Modell kann das Unternehmen die Expertise der Agentur nutzen, um einen schnellen Livegang des MVPs zu gewährleisten, während gleichzeitig langfristig Unabhängigkeit durch den Aufbau internen Wissens sichergestellt wird.

✓ CHECKLISTE EXTERNE AGENTUR ODER INHOUSE TEAM

- Im Unternehmen wurde geprüft, ob ausreichend Kompetenz und Ressourcen für das Projekt verfügbar sind (Entwickler, Designer, Tester, Systemadmins, Projektleitung, ...).
- Falls eine externe Agentur zur Realisierung des Projekts in Frage kommt, werden die Vor- und Nachteile einer gemeinsamen Zusammenarbeit abgewägt.
- Es wurde geklärt, ob die externe Agentur das Projekt im Wesentlichen alleine realisiert, oder ob es eine Kollaboration z.B. auf Entwickler-Ebene, geben wird. Die Verantwortlichkeiten wurden genau definiert.

2.7 Die richtige Agentur wählen

Die Auswahl der passenden Agentur ist ein entscheidender Schritt, weil diese nicht nur die technische Umsetzung leisten sollte, sondern als ein langfristiger Partner mit Know-how und Flexibilität die Ziele des Projekts unterstützt. Hier sind die zentralen Aspekte, die bei der Wahl der richtigen Agentur beachtet werden sollten:

Die Kommunikation ist ein Schlüsselfaktor in jedem Projekt. Eine **gemeinsame Sprache und ggf. eine geografische Nähe** erleichtern Meetings, die Abstimmung und die Zusammenarbeit erheblich. Besonders in kritischen Phasen (z.B. beim Projekt Go-Live) oder bei komplexen Anforderungen kann ein persönlicher Vor-Ort-Termin hilfreich sein.

Man sollte vorab prüfen, ob die Agentur **fundierte Erfahrung mit der gewünschten Plattform** (z. B. Magento 2, Shopware 6, Shopify, ...) und dem gewünschten Architekturansatz (z.B. Headless Commerce, Microservices oder On-Premise) Lösungen hat. Zusätzlich sollte die Agentur im entsprechenden Geschäftsmodell (B2B, B2C, D2C, Dropshipping) spezialisiert sein. Zertifizierungen oder Partnerschaften mit der Plattform können als Nachweis für ihre Expertise dienen.

Referenzen und Bewertungen von früheren Kunden können besonders hilfreich sein. Diese geben Einblick in die Arbeitsweise der Agentur, ihre Zuverlässigkeit und ihre Fähigkeit, Projekte erfolgreich umzusetzen. Falls die Agentur gute Arbeit leistet, wird sie einen oder mehrere Kundenkontakte herausgeben, um mehr Informationen über die Zusammenarbeit zu erfahren. Diese Gelegenheit sollte man unbedingt wahrnehmen. Eine Live-Demo oder die Vorstellung eines ähnlichen Projekts durch den potenziellen Auftragnehmer kann dabei helfen, Vertrauen aufzubauen und den "Nasenfaktor" zu ermitteln.

Des Weiteren sollte die **Teamgröße** und die Verfügbarkeit von spezialisierten Entwicklern geprüft werden, ggf. auf der Internetpräsenz der Agentur. Wichtig ist das Verhältnis der Entwickler zu den verschiedenen Technologien, damit sichergestellt ist, dass genügend Experten für Ihre Plattform und Anforderungen bereitstehen.

Viele Agenturen bieten ein **breites Dienstleistungsspektrum**, das über die reine Entwicklung hinausgeht. Dazu gehören Design, SEO/Marketing, UX-Beratung, Testing sowie A/B-Tests. Je nach Projektanforderung ist es sinnvoll, eine Agentur mit einem umfassenden Angebot zu wählen, um Synergien zu nutzen.

Die **Arbeitsweise und das Projektmanagementmodell** der Agentur sollten zum Unternehmen passen. Ob SCRUM, Wasserfall oder hybride Ansätze – die Agentur sollte über klare Prozesse verfügen. Außerdem ist die Wahl der Kommunikationstools entscheidend: Werden Slack[11], E-Mails oder gemeinsame Sprint-Reviews genutzt? Erhält der Auftraggeber einen eigenen Zugang zum Projektmanagement Tool (z.B. JIRA), um den Status verfolgen zu können? Transparenz und regelmäßige Updates sind essenziell für den Projekterfolg.

Das **Kostenmodell** der Agentur sollte zum Budget und der Ausgabenpolitik des Auftraggebers passen. Gängig sind Time-and-Material (T&M), Fixpreis oder agile Fixpreise. Bei Fixpreisangeboten sollte man auf die Granularität achten: Sind alle Anforderungen (siehe vorheriges Kapitel zu den eigenen Anforderungen) detailliert abgedeckt, oder gibt es unklare Bereiche, die zu Nachbudgetierungen führen können? Vergleiche mit anderen Angeboten und gezielte Rückfragen helfen, falls einzelne Posten unverhältnismäßig erscheinen.

Der **Vertrag** sollte klar Verantwortlichkeiten und Fristen regeln. Falls der Vertrag unverhältnismäßig auftragnehmerfreundlich gestaltet ist, sollte man diesen kritisch hinterfragen. Außerdem geprüft werden, wie die Agentur die **Betreuung und Wartung nach dem Go-Live** gestaltet. Gibt es mehrere Servicepakete und Entwicklerkontingente? Bietet sie Support mit Service-Level-Agreements (SLAs) an? Gibt es eine Roadmap-Beratung für künftige Weiterentwicklungen? Wichtig ist auch, ob das ursprüngliche Team weiterhin betreut oder ein neues Team übernimmt.

Persönliche Chemie spielt eine nicht zu unterschätzende Rolle und es ist extrem hilfreich, das Team und insbesondere den Projektleiter, in der Anbahnungsphase kennenzulernen. Die Vorstellung einer Live-Demo oder eines anderen Kundenprojekts kann helfen, die Arbeitsweise und die Qualität der Agentur zu bewerten. Sympathie und Vertrauen fördern eine produktive Zusammenarbeit.

✓ CHECKLISTE AGENTUR WÄHLEN

- Die Agentur hat nachgewiesene Kompetenzen und Erfahrungen (z.B. in Form von Kundenreferenzen) im Rahmen der gewünschten E-Commerce Plattform.
- Die Agentur hat eine ausreichende Teamgröße und bietet weitere Leistungen an, die Synergieeffekte für das Projekt haben werden (z.B. Design, Testing, Online-Marketing, Hosting, ...).

- Das Kostenmodell sowie der Projektmanagement-Ansatz sind fair und transparent. Das Angebot und der Vertrag bilden die Annahmen aus den Vorgesprächen ab. Insbesondere das Angebot wurde bzgl. der definierten funktionalen Anforderungen abgeglichen.

2.8 Eine Entscheidung mittels Business Case treffen

In den vorherigen Abschnitten wurde beschrieben, wie die Entscheidung zu richtigen E-Commerce Plattform getroffen werden soll. Zudem hat der E-Commerce Owner eine Orientierung, ob das Projekt inhouse vom eigenen Team, oder von einem externen Dienstleister implementiert werden soll. In diesem Abschnitt wird besprochen, ob das Projekt durchgeführt werden soll und ob sich das Projekt rentiert. Um diese Frage zu beantworten, kann die Erstellung eines Business Cases helfen. Dieser enthält im Regelfall eine Gegenüberstellung des erwarteten Nutzens und der durch das Projekt entstehenden Kosten. Hilfreich dabei ist die Erstellung mehrerer Szenarien, um konservative, realistische und optimistische Fälle abzudecken. Im Folgenden werden zwei Beispiele für Business Cases skizziert.

Business Case A: Der E-Commerce Owner möchte mit einer externen Agentur ein Projekt angehen, um die Ladezeiten und die UX des Storefronts zu verbessern. Der potenzielle Partner und der E-Commerce Owner rechnen also mit einer Verbesserung der Conversion Rate. Neben dem Fixpreis für die Implementierung des Projekts entstehen jährliche Lizenzgebühren für die E-Commerce Software.

KPI	KONSERVATIV	REALISTISCH	OPTIMISTISCH
Aktueller Umsatz	€10.000.000,00	€10.000.000,00	€10.000.000,00
Anzahl Bestellungen	35000	35000	35000
Anzahl der Nutzer	1300000	1300000	1300000
Aktueller Warenkorbwert	€307,69	€307,69	€307,69
Aktuelle Conversion Rate	2,50 %	2,50 %	2,50 %
Verbesserung der Conversion Rate (Prozentpunkte)	**5,00 %**	**10,00 %**	**15,00 %**

Verbesserte Conversion Rate	2,63 %	2,75 %	2,88 %
Verbesserter Umsatz in EUR	€10.500.000,00	€11.000.000,00	€11.500.000,00
Umsatzsteigerung in EUR	€500.000,00	€1.000.000,00	€1.500.000,00
Projektkosten (Fixpreis)	€190.000,00	€190.000,00	€190.000,00
Jährliche Lizenzgebühren	€10.000,00	€10.000,00	€10.000,00
Return on Invest	**1,5**	**4**	**6,5**

Der E-Commerce Owner ermittelt in seinem Analytics Tool die oben genannten KPIs. In den drei Szenarien werden jeweils konservative (5%), realistische (10%) und optimistische (15%) Annahmen für die Verbesserung der Conversion Rate getroffen. Der E-Commerce Owner errechnet, dass der Umsatz c.p. um 500.000,00 €, 1.000.000,00 € und 1.500.000,00 € steigt. In jedem Szenario ergibt sich ein positiver Return on Invest, womit sich das E-Commerce Replatforming rentieren sollte.

Business Case B: Der E-Commerce Owner eines B2B Shops möchte mit seinen eigenen Entwicklern eine neue E-Commerce-Plattform implementieren und diese mit einem PIM, CRM und CMS integrieren, um Prozesse zu automatisieren und manuelle Aufwände zu reduzieren. Der E-Commerce Owner geht damit von einer Reduzierung der Personalkosten aus. Die entsprechenden Mitarbeiter, darunter Content-Manager, Marketer, Produktdatenmanager, Kundensupport, und Verkäufer im Außendienst, können schneller und weniger fehleranfällig arbeiten, womit sie Ihre Zeit für alternative Tätigkeiten nutzen können.

KPI	KONSERVATIV	REALISTISCH	OPTIMISTISCH
Aktuelle Personalkosten f. E-Commerce	€1.500.000,00	€1.500.000,00	€1.500.000,00

Reduzierung der Personalkosten in %	**10,00 %**	**20,00 %**	**30,00 %**
Reduzierte Personalkosten in €	150.000,00 €	300.000,00 €	450.000,00 €
Projektkosten (Personalkosten hauseigene Entwickler)	€250.000,00	€250.000,00	€250.000,00
Return on Invest	**-0,4**	**0,2**	**0,8**

Im konservativen Szenario wird ein negativer Return on Invest erreicht, während im realistischen und optimistischen Szenario jeweils ein positiver Return on Invest erreicht wird. Es besteht also ein Risiko, dass sich die Durchführung des Projekts nicht lohnen wird.

Bei den beiden obigen Beispielen sollte man beachten, dass nur die einmaligen und isolierten Projektkosten betrachtet werden. Der E-Commerce Owner sollte jedoch vielmehr die gesamten TCO (inkl. Hosting sowie Wartung und Instandhaltung) für mehrere Jahre betrachten.

✓ CHECKLISTE BUSINESS CASE RECHNEN

- Es existieren ausreichend viele KPIs und Quellen für die Errechnungen eines Business Cases.
- Der Business Case berücksichtigt mindestens drei Szenarien: konservativ (Worst-Case), realistisch sowie optimistisch (Best-case).
- Der Business Case basiert auf realistischen Annahmen. Die Annahmen wurden von Externen oder von dritten Quellen validiert. Der Business Case wurde nicht “schöngerechnet”.

2.9 Dos und Don'ts in der Entscheidungsphase

In der Entscheidungsphase eines Replatforming-Projekts gibt es einige zentrale Dos und Don'ts, die bereits vor dem eigentlich Projektstart über den Erfolg entscheiden können. Der E-Commerce Owner sollte **sich nicht von Trends wie Headless oder Microservices blenden lassen**, wenn diese nicht zu den tatsächlichen Anforderungen und Zielen passen – Technologie sollte immer Mittel zum Zweck sein. Viele kleinere Händler mit einem simplen Geschäftsmodell brauchen keine komplexe Architektur, die erhebliche Kosten verursacht. Stattdessen sollte man sorgfältig die Plattform evaluieren und die **Reife und Größe der Plattform in Betracht ziehen**. Eine starke Community, zahlreiche Erweiterungen und reife Agenturlandschaft, wie sie bei Magento 2 oder Shopware vorhanden sind, können entscheidende Vorteile bieten.

Zudem sollte der E-Commerce Owner vermeiden, die **Wahl der Agentur ausschließlich auf Basis des günstigsten Angebots oder Stundensatzes** zu treffen. Qualität hat ihren Preis, und eine falsch ausgewählte Agentur kann langfristig teurer werden oder es kommt ggf. zu Nachbudgetierungen. Man sollte sich eher darauf konzentrieren, die passende Expertise und ein solides Team mit klar definierten Rollen für Ihr Projekt zu finden. Tipp: falls die Agentur eigenständig einen Business Case errechnet, zeugt das von Selbstvertrauen in die eigenen Fähigkeiten.

Auch sollten die verschiedenen Partner im Projekt mit Bedacht gewählt werden. Das **Involvieren von zu vielen Agenturen oder Teams führt zu mehr Koordination** und Kommunikation, was wiederum schnell in Abstimmungsproblemen und verzögerten Entscheidungen resultiert. Ein breites Leistungsportfolio von Agenturen wird derartige Herausforderungen vermeiden, indem es zu Synergien kommt und unnötige Aufwände für Meetings reduziert werden.

Schließlich sollte der E-Commerce Owner den **Business Case nicht „schön" rechnen**, um das Projekt zu rechtfertigen – unrealistische Annahmen können später die Wirtschaftlichkeit gefährden. Man sollte besser auf Transparenz und fundierte Zahlen aus einem Analytics Tools setzen, um eine langfristig tragfähige Entscheidung zu treffen.

KAPITEL 3

ANALYSE

Die Analysephase bindet oftmals die größten Ressourcen in einem E-Commerce-Replatforming. In dieser Phase wird der Ist-Zustand der aktuellen Plattform gründlich untersucht, um Potenziale und Anforderungen zu identifizieren. Dabei liegt der Fokus nicht nur auf den technologischen Grundlagen, sondern auch auf den funktionalen Aspekten der Storefront, des Admin-Bereichs sowie der gesamten Systemlandschaft. Nur mit einer klaren und detaillierten Analyse können fundierte und faktenbasierte Entscheidungen für die kommende Plattform getroffen werden.

Ein Schwerpunkt dieser Phase liegt auf der Analyse der Business-Aspekte. Umsatz, Conversion Rates und Kundenfeedback liefern wertvolle Einblicke in die wirtschaftliche Performance des E-Commerce Systems und helfen, Optimierungsansätze zu identifizieren. Ergänzend dazu wird in einer Stakeholder-Analyse untersucht, welche internen und externen Akteure von der Plattform betroffen sind und welche Interessen und Erwartungen diese mitbringen.

Auch die Betrachtung der Wettbewerber und Zielgruppen ist entscheidend, um das Replatforming richtig zu positionieren. Eine detaillierte Zielgruppenanalyse zeigt auf, welche Bedürfnisse und Kaufmotive die Kunden haben, während die Wettbewerberanalyse dabei hilft, Differenzierungsmöglichkeiten und Best Practices zu erkennen.

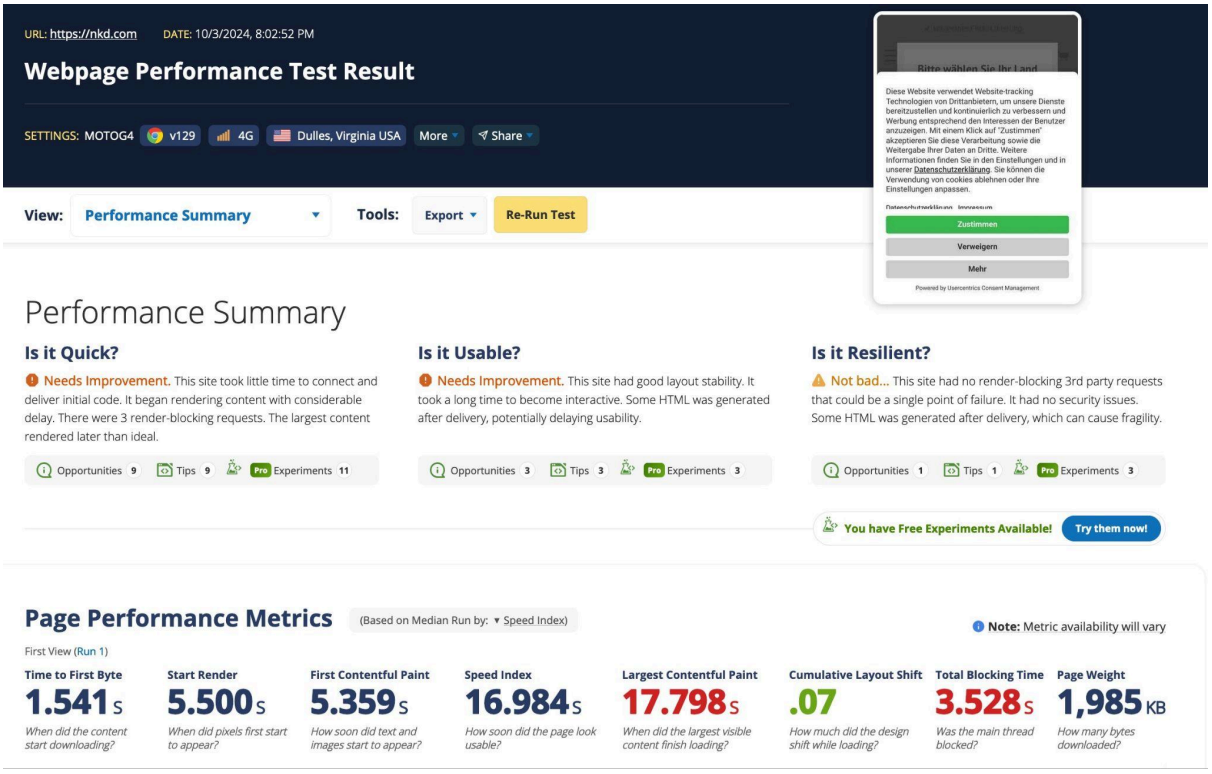
Zusätzlich wird das Business Model Canvas genutzt, um das E-Commerce Geschäftsmodell umfassend zu verstehen und strategische Ansätze zu konkretisieren. Abgeschlossen wird die Analysephase mit einer Übersicht über die wichtigsten Dos und Don'ts, um häufige Fehler zu vermeiden und sicherzustellen, dass die Analyse fundierte Ergebnisse liefert, die als solide Basis für die weiteren Projektphasen dienen können.

3.1 Analyse Ist-Zustand

Die Analyse vom Ist-Zustand des aktuellen E-Commerce Systems ist eine der wichtigsten und vermeintlichen schwierigsten Aufgaben in einem Replatforming Projekt. Hierbei sollten technologische und Business Aspekte berücksichtigt sein, sowie eine funktionale Analyse vom Storefront und dem Admin-Bereich erfolgen. Zudem sollte die Systemlandschaft beschrieben werden. Da die Ist-Analyse sowohl eine sehr gute ganzheitliche Kenntnis des aktuellen E-Commerce Systems als auch ein solides Grundverständnis E-Commerce IT im Allgemeinen verlangt, bindet dieser Arbeitsschritt erhebliche Zeit und Mitarbeiterressourcen. Zeitgleich bildet die Analyse vom Ist-Zustand den Grundstein zur Definition der Anforderungen an das neue E-Commerce-System.

Technologische Aspekte

Bei technologischen Aspekten werden etwa die KPIs zur technischen Performance festgehalten, wie Ladezeiten, Time-to-first-byte, First Contentful Paint, Dateigröße der Seite, Uptime, usw. Zudem sollte sich der E-Commerce Owner qualitative Fragen zur aktuellen Technologie stellen, z.B. wie oft werden Shop-Updates veröffentlicht, wie funktioniert das Hosting und ist es skalierbar mit wachsendem Traffic, Produkten und Sprachen. Auch sollte geprüft werden, ob ein gewisses Maß an Sicherheit gewährleistet ist. Letzterer Punkt sollte auch aus DSGVO Gesichtspunkten geprüft werden, da der E-Commerce Owner im schlimmsten Fall rechtliche Probleme bekommen könnte. Zudem werden aus SEO Sicht “Web Vitals” (z.B. Cumulative Layout Shift, Largest contentful Paint, First Input Delay) immer wichtiger.[12]



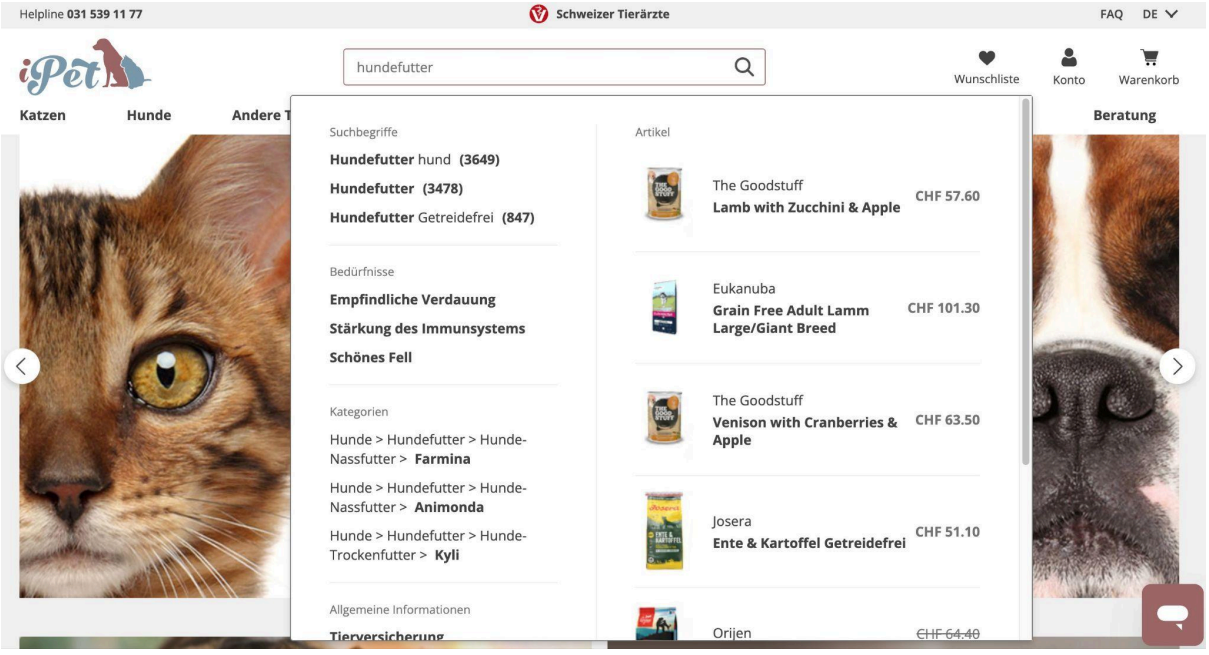
Auf der Seite webpagetest.org können Test zu Ladezeiten sowie Web Vitals erstellt werden

Funktionale Analyse Storefront und E-Mails

Die funktionale Analyse des Storefronts und der E-Mails umfasst alle Funktionen, die für den Endnutzer sichtbar sind. Wichtig ist dabei, alle relevanten Endgeräte zu beschreiben (Desktop, Tablet und Mobiles).

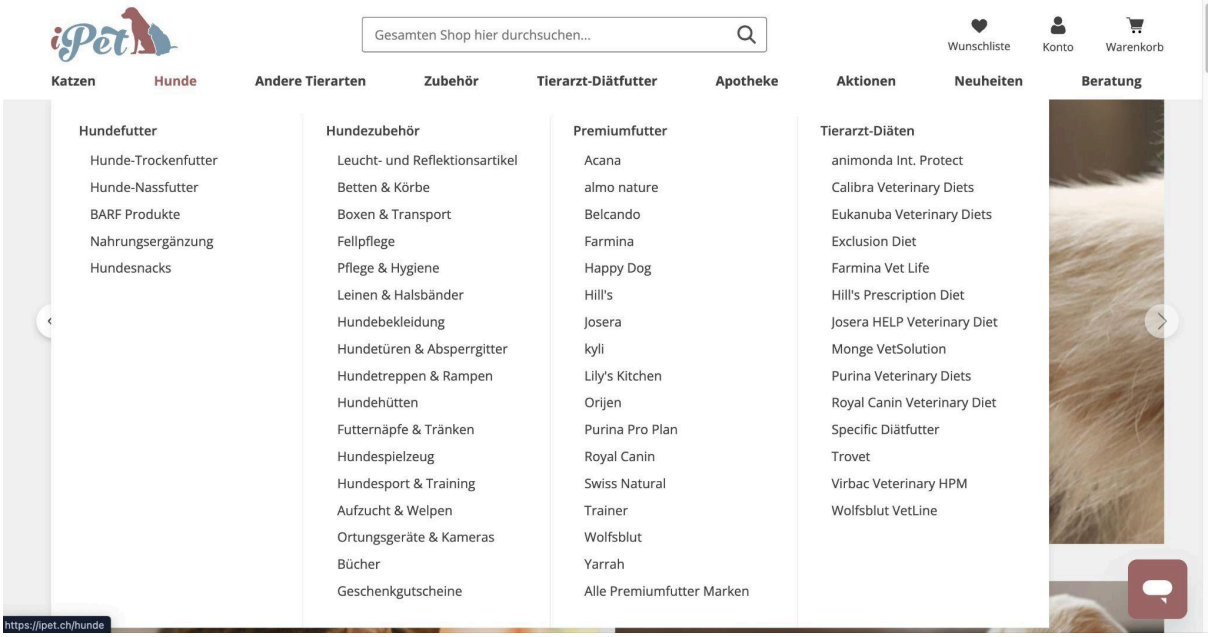
In Bezug auf die Navigation und die Suche im Storefront sollten etwa die folgenden Funktionen berücksichtigt sein:

- Wie funktioniert die Suche? Nach welchen Entitäten wird zu einem Suchbegriff gesucht? Sind es Produkte, Kategorien, Marken, und verwandte Suchbegriffe? Wird auch nach Content-Seiten, wie z.B. Ratgeber, FAQ und Blog Seiten, gesucht? Gibt es eine Fehlertoleranz, sodass z.B. “shirt” die gleichen Suchergebnisse wie “shyrt” liefert? Gibt es neben dem Autosuggest Flyout eine eigene Suchergebnisseite mit Filtern und einer Sortierung?



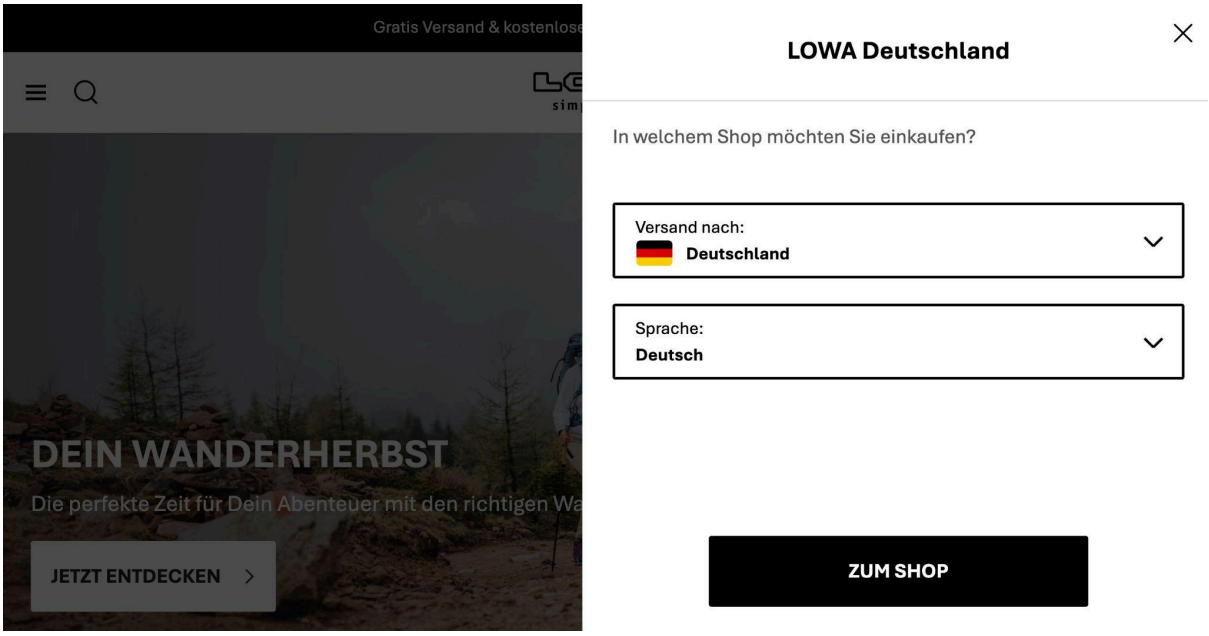
Autosuggest Flyout auf ipet.ch

- Wie funktioniert die Kategorienavigation im Header Bereich? Wie viele Kategorien werden im Storefront angezeigt? Wie viele Ebenen an Unterkategorien gibt es? Gibt es einen Button, der alle Kategorien anzeigt? Handelt es sich nur um Produktkategorien oder werden auch Landing Pages bzw. sonstige Content Seiten und externe URLs verlinkt? Wie sieht die Kategorie Navigation auf mobilen Endgeräten aus?



Kategorienavigation auf ipet.ch

- Gibt es einen Switcher für verschiedene Sprachen, Lieferländer oder Währungen? Wie verhält sich die URL bzw. Domainstruktur nach dem Wechsel? Werden Sub-Folder oder Sub-Domains für die unterschiedlichen Sprachen genutzt? Gibt es im Hintergrund eine Geolocation Logik, die den Nutzer abhängig von der IP-Adresse oder den Browsereinstellungen auf eine bestimmte Sprache führt?



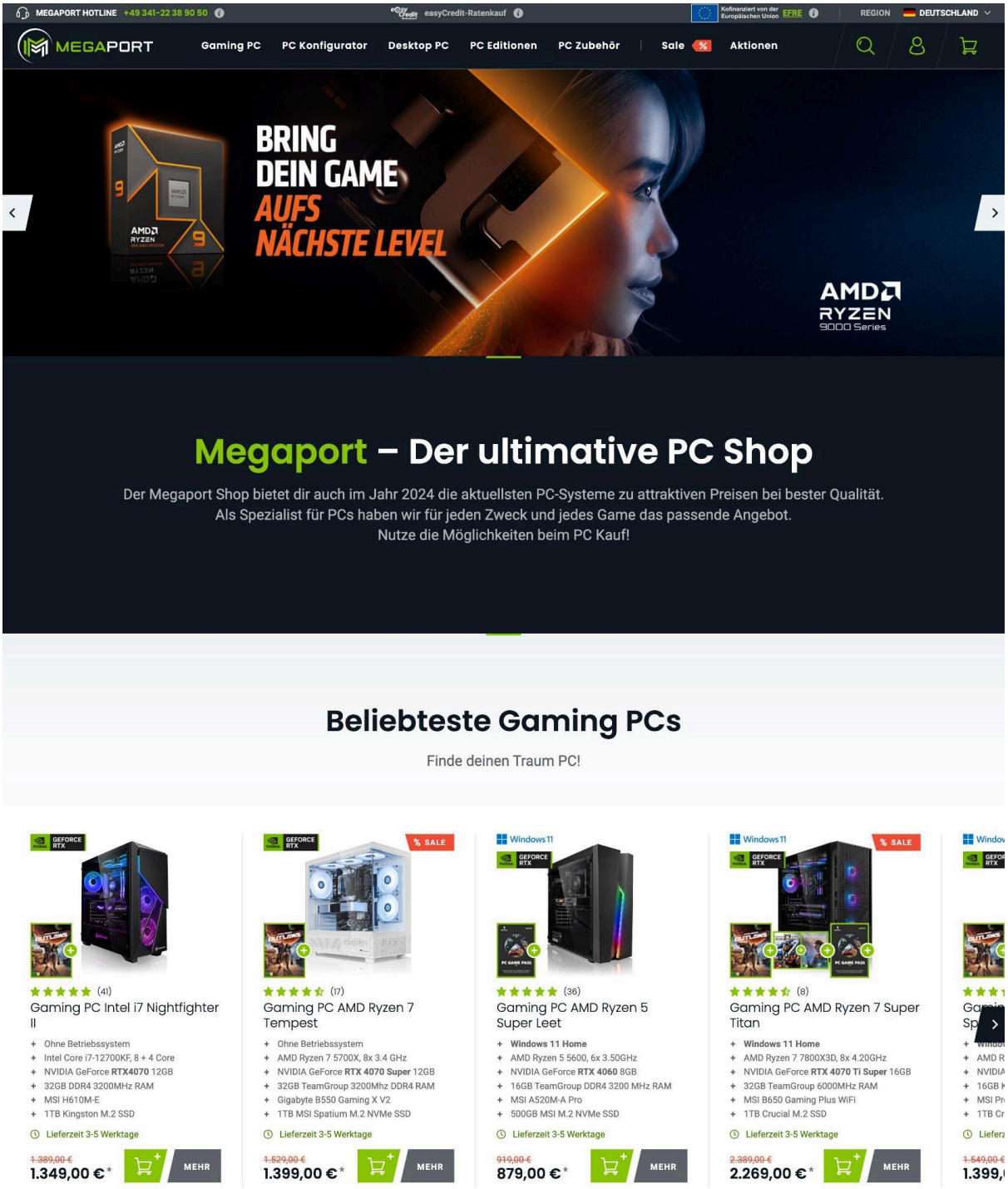
Switcher des Versandlands und der Sprache auf lowa.com

- Wie funktioniert die Footernavigation? Werden die gleichen Inhalte aus der Kategorienavigation übernommen oder handelt es sich um eine separate

Navigation? Werden rechtliche Seiten (z.B. AGB oder Impressum) und Content-Seiten (z.B. Über Uns) verlinkt? Sind Links zu sozialen Medien enthalten?

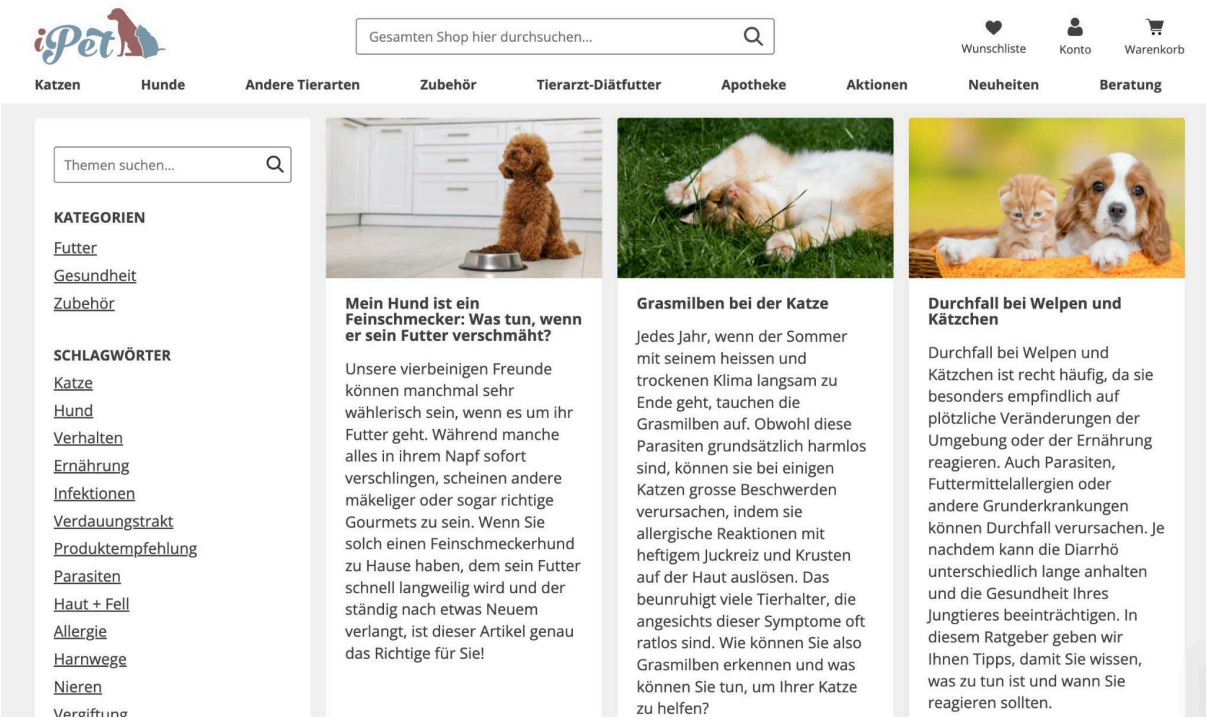
Hinsichtlich der Homepage, Landing-Pages und Content Seiten, sollten vor allem die genutzten Komponenten und deren Responsiveness beschrieben werden. Darunter fallen etwa:

- Welche Teaser Komponenten (Bilder) werden genutzt? Sind es nur große Hero Teaser (Bühnen) auf ganzer Bildschirmbreite oder Container Breite? Gibt es mehrere Bilder, die zeitlich automatisch rotieren oder die mittels Pfeilen vom Nutzer angeklickt werden müssen? Gibt es Teaser Komponenten mit zwei, drei oder mehr Bildern pro Reihe? Sind auf dem Bild Call-to-Action (CTA) Buttons platziert und Text, der markierbar ist?
- Werden neben Bildern ebenfalls Videos genutzt? Woher stammen die Videos? Falls die Videos aus externen Quellen (YouTube, Vimeo, usw.) stammen, sind diese DSGVO-konform über einen User Consent eingebunden?
- Gibt es Textkomponenten? Wenn ja, welche Textformatierungen sind sichtbar (z.B. fett, kursiv, Links)? Gibt es Komponenten mit Text und Teasern in einer Reihe? Gibt es Textparagraphen mit mehreren Spalten?
- Gibt es Commerce Komponenten, mit einem Produktkarussell oder Produktgrid? Welche Informationen sind in der Produktkachel enthalten? Sind es Produktbild, Preis, Produktname, Lagerverfügbarkeit und Bewertungen? Gibt es einen Hover-Effekt, der z.B. das Bild vergrößert? Gibt es einen “in den Warenkorb legen” Button (Add-to-cart)? Wie werden weitere Produkte bei einem Karussell geladen, z.B. per Pfeil oder automatisch rotierend?



Der obere Teil der Homepage mit Hero, Text und Commerce Komponenten auf megaport.de

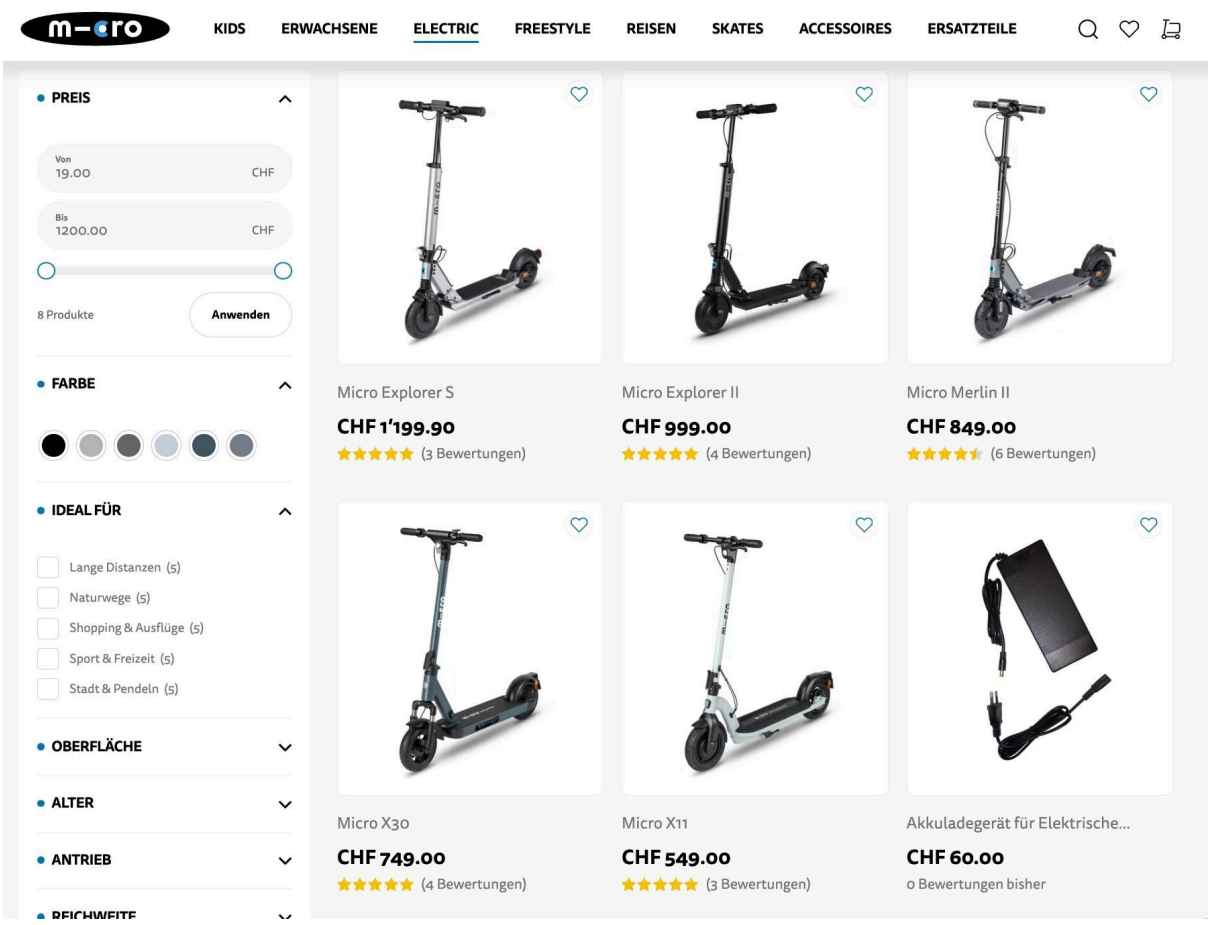
- Werden Content-Seiten kategorisiert oder gibt es ein Untermenü? Gibt es Blog, FAQ oder Ratgeber Funktionen, die nach Content-Seiten suchen, sortieren oder filtern können?



Der Blog (Hunde und Katzen Ratgeber) auf ipet.ch

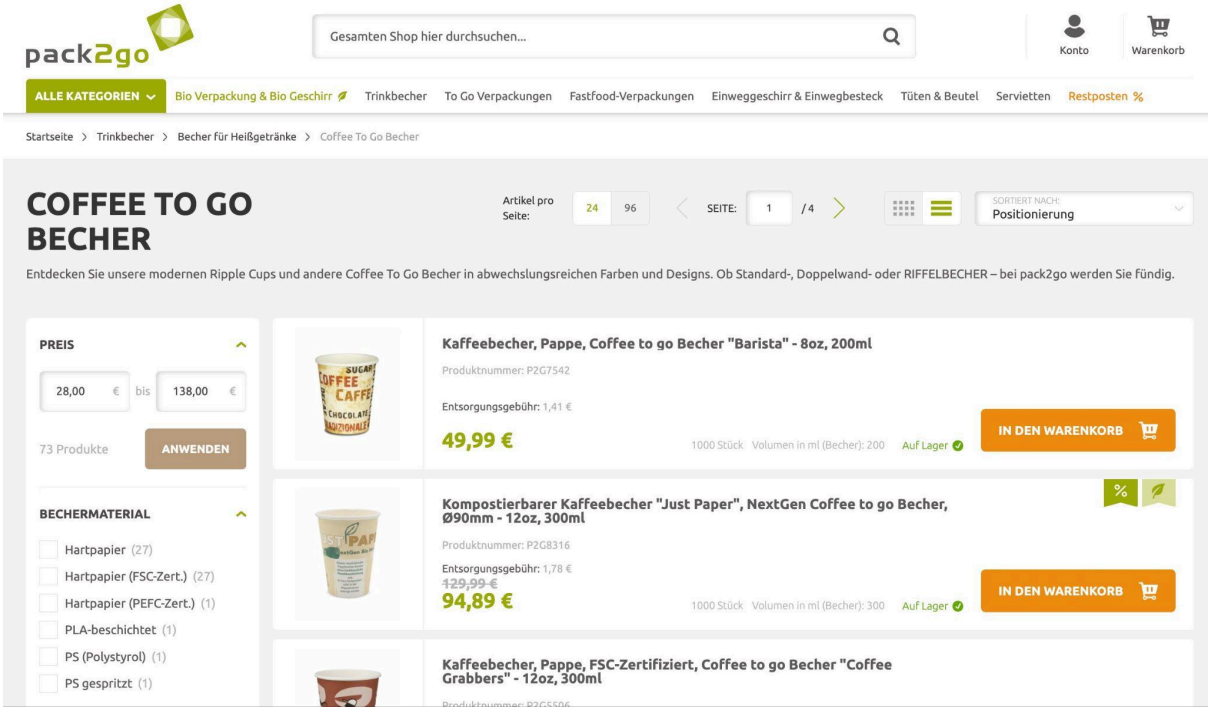
Auf eine Kategorieseite wird der Nutzer seine Produktnavigation mit Filtern und einer Sortierung verfeinern können. Hierbei sollten die folgenden Elemente beschrieben sein:

- Wie sind die Filter gelegen, horizontal über den Produktergebnissen oder vertikal in der Seitenleiste? Welche Attribute werden als Filter (Facet Navigation) genutzt? Wie werden die Filteroptionen genutzt, z.B. mit einem Slider (vor allem bei dem “Preis” Filter), per Input, oder per Checkbox? Kann man innerhalb des Filters suchen, z.B. nach “Markennamen”? Wie sind die Filteroptionen sortiert, alphabetisch oder nach den höchsten Produktergebnissen? Werden Unterkategorien angezeigt?



Kategorieseite inkl. Filter in der vertikalen Seitenleiste auf micro-scooter.com

- Welche Sortieroptionen gibt es? Sind es nur Produktname (alphabetisch) und Preis? Kann man nach Bestsellern (sortiert nach der Anzahl an Verkäufen) sortieren? Gibt es weitere Sortieroptionen, wie z.B. Kundenbewertungen?
- Kann man die Ansicht ändern, bspw. zwischen einem Grid (mehrere Produktkacheln pro Reihe) und List View (ein Produkt pro Reihe)? Kann man die Anzahl der Produkte pro Seite ändern?



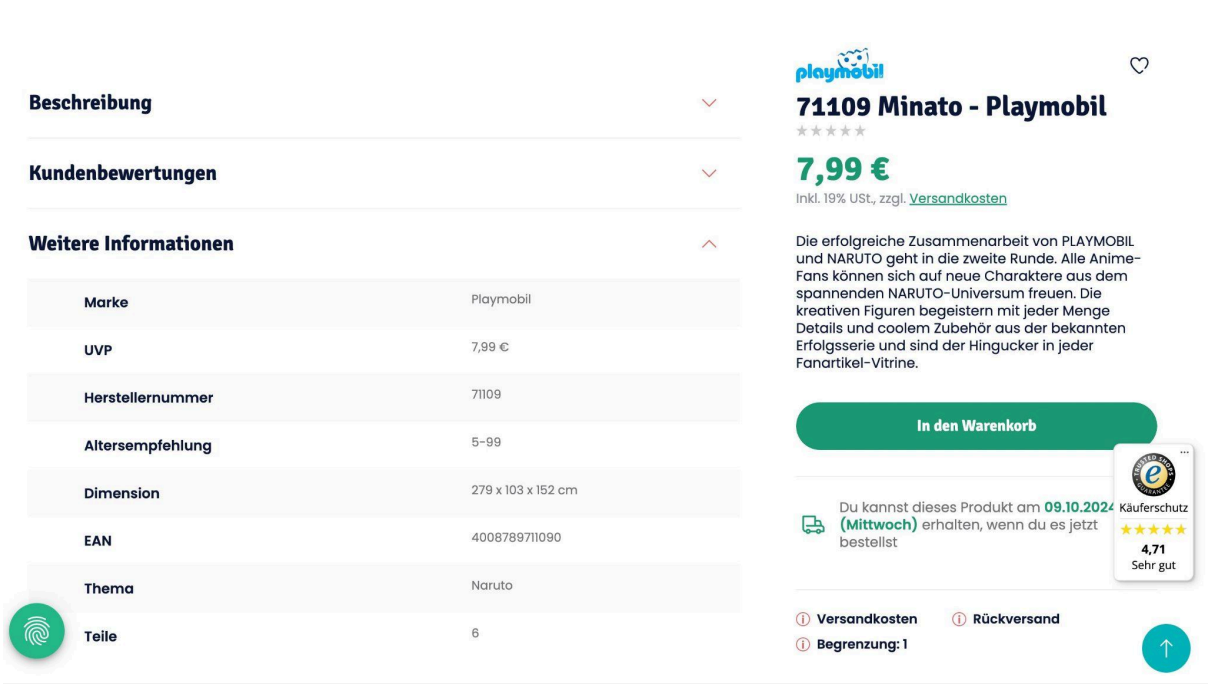
Anpassung der Kategorieansicht (List bzw. Grid View Switcher und Anpassung der Produkte pro Seite) auf pack2go.de

- Wie werden weitere Seiten geladen? Gibt es eine klassische Paginierung (Seite 1, Seite 2, usw.), einen “Mehr Laden” Button (mehr Produkte werden nach Klick geladen) oder einen Infinite Scroll (mit Scrollen werden weitere Produkte geladen)? Wie verhalten sich die Filter Links aus SEO-Sicht bzw. sind die URLs aller Filterkombinationen durch Suchmaschinen indexierbar?
- Gibt es einen Content-Bereich auf der Kategorieseite, z.B. mit Bildern und (SEO relevanten) Texten?

Auf der Produktdetailseite findet der Nutzer deutlich mehr Details zum Produkt. Hier sollte man insbesondere auf die folgenden Funktionen eingehen:

- Wie werden die Produktbilder dargestellt? Gibt es eine Galerie mit mehreren Bildern und kann der Nutzer das Produktbild mit einem Zoom Effekt vergrößern? Können auch Videos statt Bildern dargestellt werden?
- Welche Informationen sind in der Buybox (der Bereich mit Add-to-cart sowie Preisen) enthalten? Sind es Produktname, Kategorie und ggf. die Marke? Wird nur ein Preis genannt oder sind ebenfalls Streichpreise (bzw. Rabatte), UVP, Basispreis (z.B. Preis pro Kilogramm) sowie der nach EU Richtlinie erforderliche 30-Tage-Tiefstpreis genannt[13]? Sind die Preise in netto oder brutto, mit enthaltenen MwSt., ausgezeichnet? Kann man die Produkte konfigurieren, z.B. hinsichtlich Größe und Farbe, oder sogar individualisieren, z.B. mit eigenem Namen auf einem Trikot? Gibt es einen Mengenswitcher mit numerischem Input

oder “+” und “-” Zeichen? Gibt es neben dem Add-to-Cart Button einen Wunschlisten-Button oder eine Vergleichsfunktion?

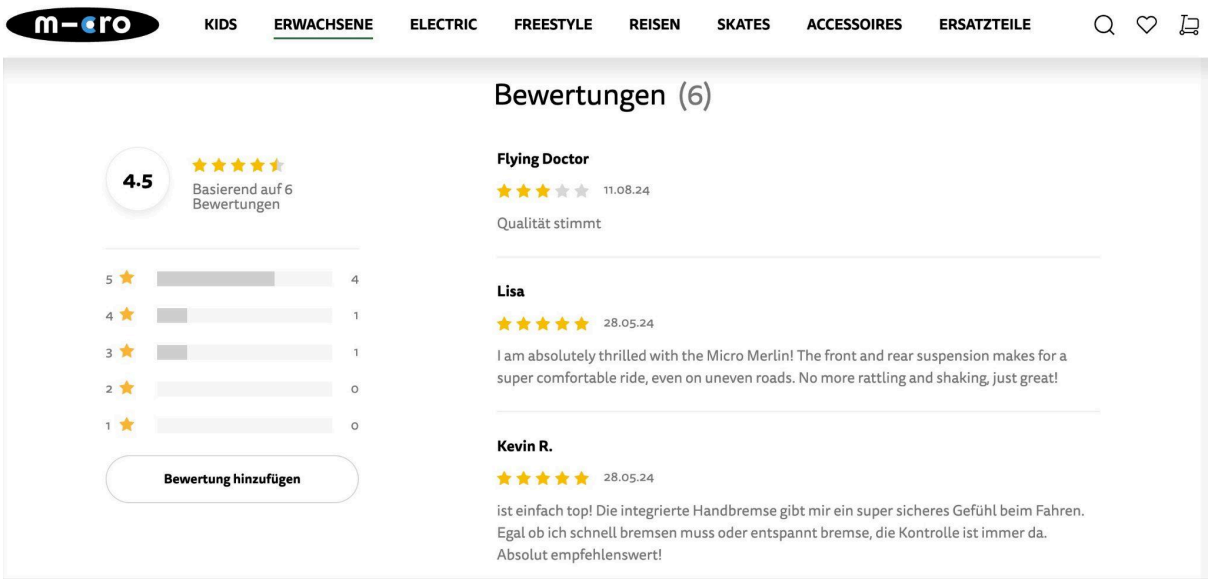


Produktdetailseite inkl. Produktgalerie und Buybox auf toys-for-fun.com

Produktbeschreibung und -details auf toys-for-fun.com

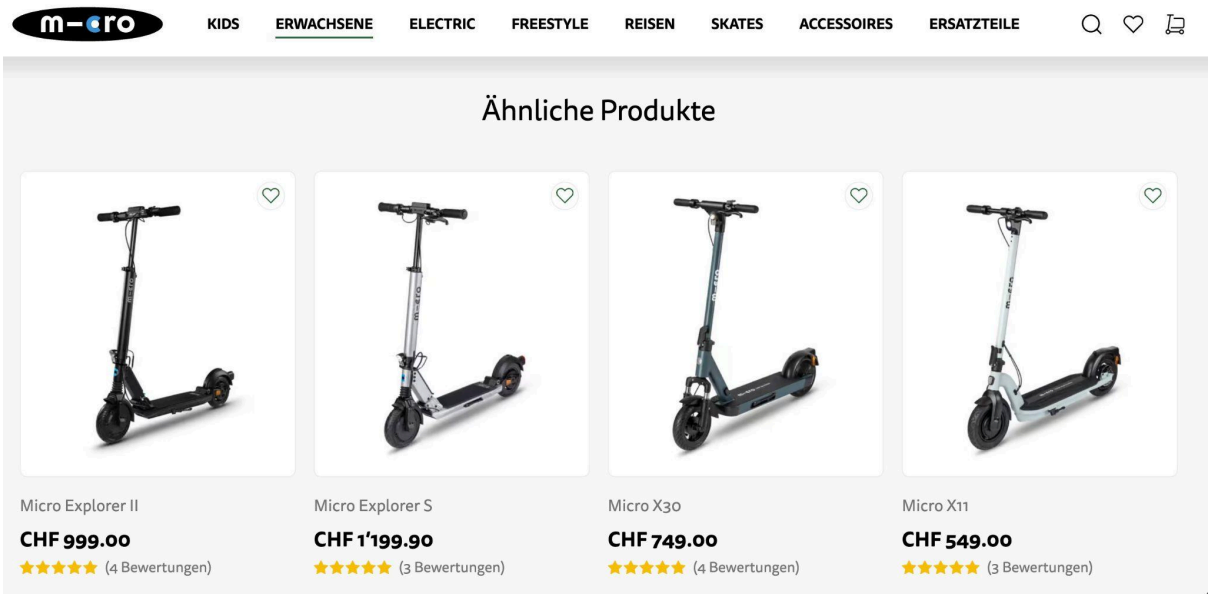
- Ist ein Hinweis für die Lagerverfügbarkeit, Lieferzeiten und Versandkosten enthalten?

- Kann man sich für ein Verfügbarkeitsabonnement einschreiben, sofern das Produkt nicht lagernd ist?
- Welche Informationen sind in den Produktbeschreibung und -details enthalten? Welche Produktattribute (z.B. Ausmaße, Gewicht, Material, Gefahrenkennzeichen, ...) sind sichtbar?
- Sind Kundenbewertungen sichtbar? Wenn ja, welche Skala wird genutzt und gibt es eine Zusammenfassung bzw. einen Durchschnitt? Sind die Bewertungen schriftlich zusammengefasst? Falls das Storefront in mehreren Sprachen verfügbar ist, sind die Bewertungen übersetzt? Kann man als Gast oder eingeloggter Kunde eine Bewertung verfassen oder können es nur verifizierte Käufer?



Produktbewertungen auf micro-scooter.com

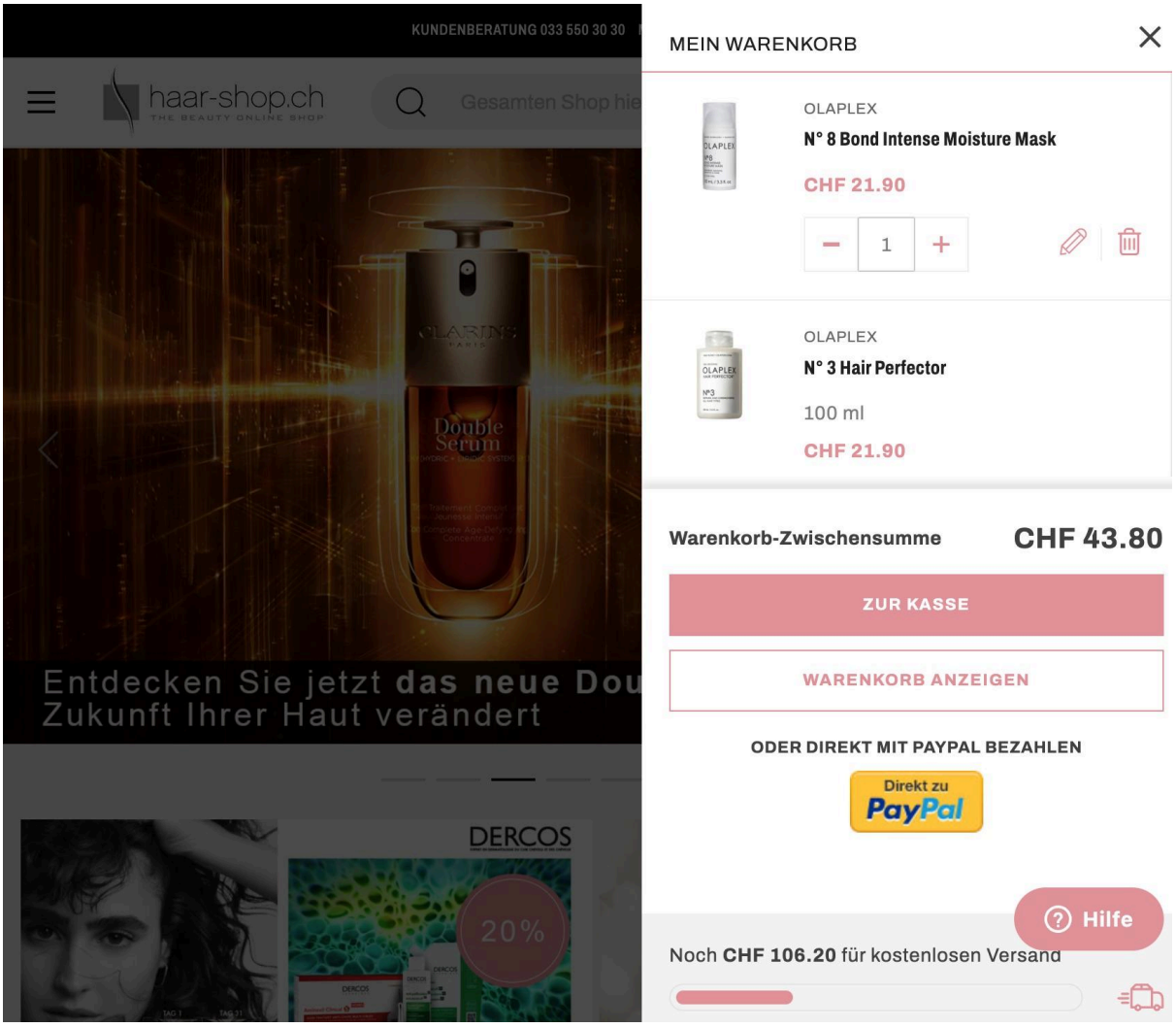
- Gibt es einen Bereich mit weiteren Produktvorschlägen, wie z.B. Zubehör Produkte (eher technische Produkte), Complete-the-Look (Fashion) oder “Kunden kauften auch”?



Weitere Produktempfehlungen auf micro-scooter.com

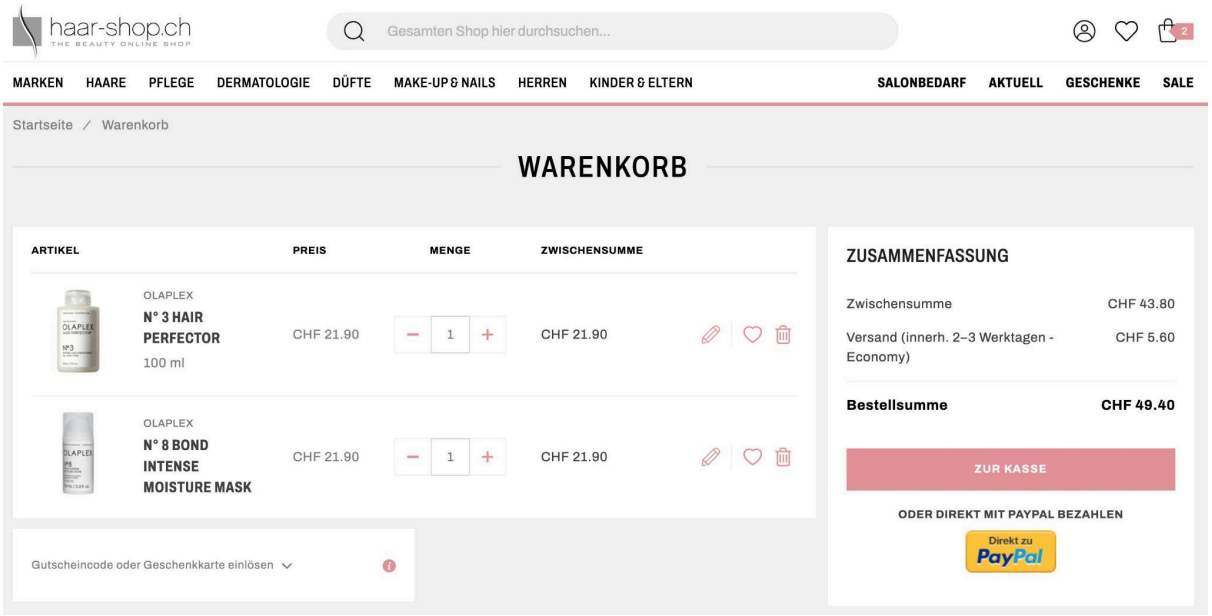
Nachdem der Nutzer Produkte im Warenkorb gelegt hat, wird er im nächsten Schritt der User Journey zum Warenkorb geführt. In diesem Bereich sollten die folgenden Punkte beschrieben sein:

- Gibt es eine Warenkorb-Vorschau (Minicart), die alle Produkte inkl. Mengen und Preise enthält? Wann öffnet sich der Minicart und kann man diesen per Icon im Header eigenständig öffnen? Kann man hier Produkte entfernen oder die Menge ändern? Gibt es eine Zusammenfassung des Gesamtbetrags inkl. Steuern und Versandkosten? Gibt es an dieser Stelle einen Express Check-out (z.B. Amazon pay oder Paypal Express), der den gewöhnlichen Check-out umgeht?



Minicart auf haar-shop.ch

- Welche Informationen sind im Warenkorb enthalten? Sieht der Nutzer eine Tabelle der Produkte inkl. Mengen und Preise pro Stück? Kann der Nutzer hier die Menge ändern oder Produkte gänzlich entfernen?
- Welche Informationen sind in der Zusammenfassung der Bestellsumme enthalten? Sind es Steuern und ggf. Versandkosten?
- Gibt es im Warenkorb einen Express Check-out (z.B. Amazon pay oder Paypal Express), der den gewöhnlichen Check-out umgeht?
- Kann der Nutzer einen Discount Code oder Gutscheine eingeben? Wenn ja, an welcher Stelle erscheint der genutzte Code und der Rabatt in absoluten Beträgen (als weitere Produktposition oder in der Zusammenfassung)?

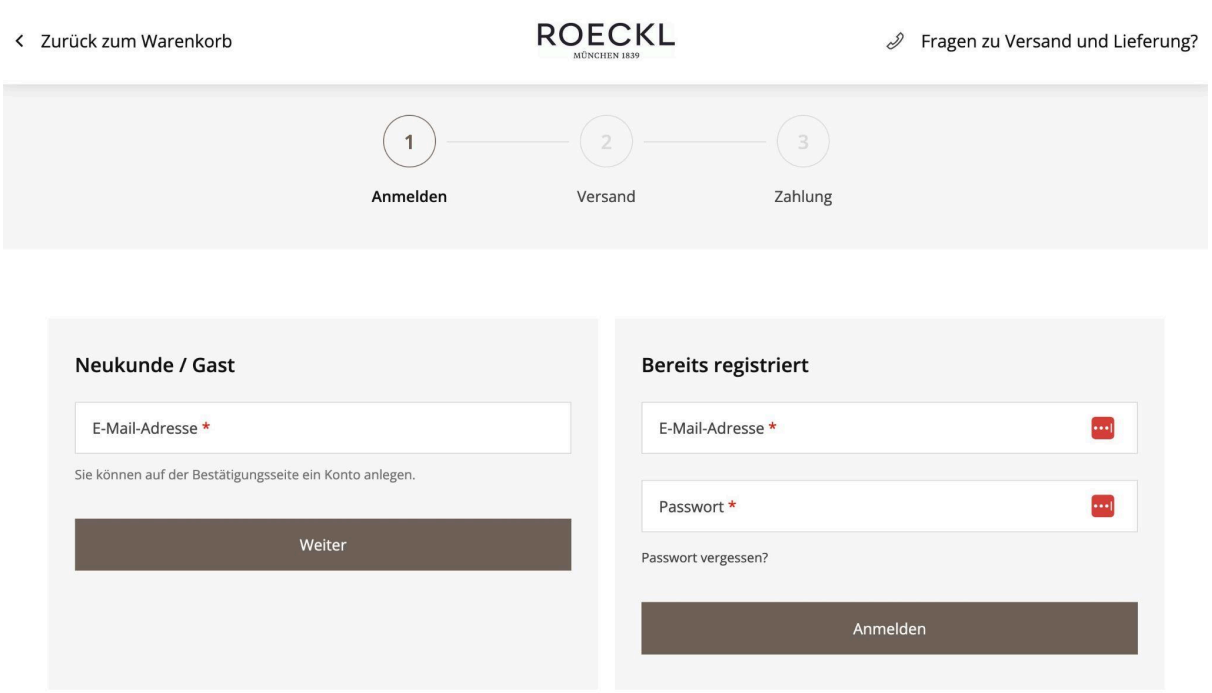


Warenkorbansicht auf haar-shop.ch

- Gibt es ein Produktkarussell mit weiteren Empfehlungen (“Folgende Produkte passen zusammen”)?

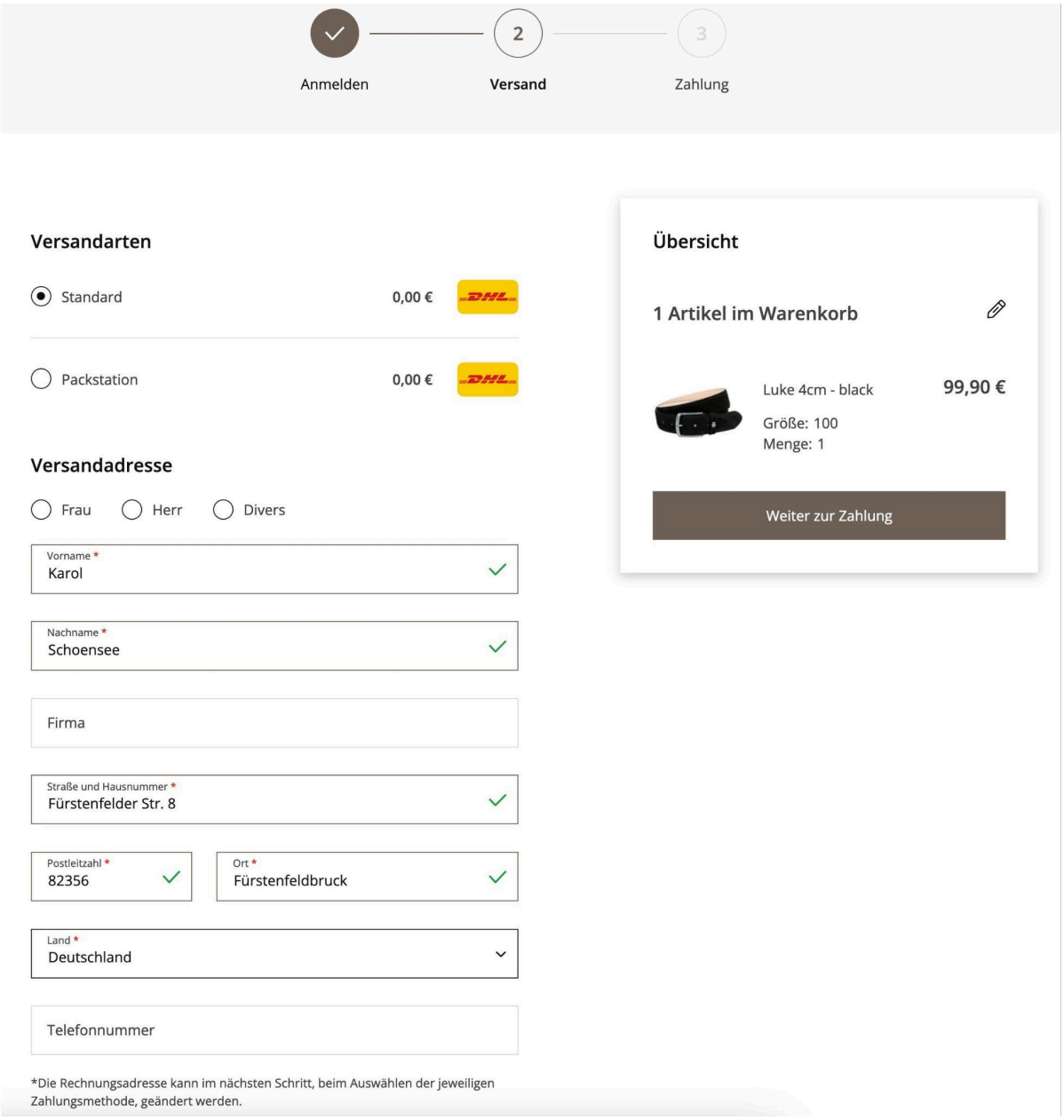
Im Anschluss an den Warenkorb folgt mit dem Check-out der wichtigste Abschnitt der User Journey. Hier sollte beschrieben sein:

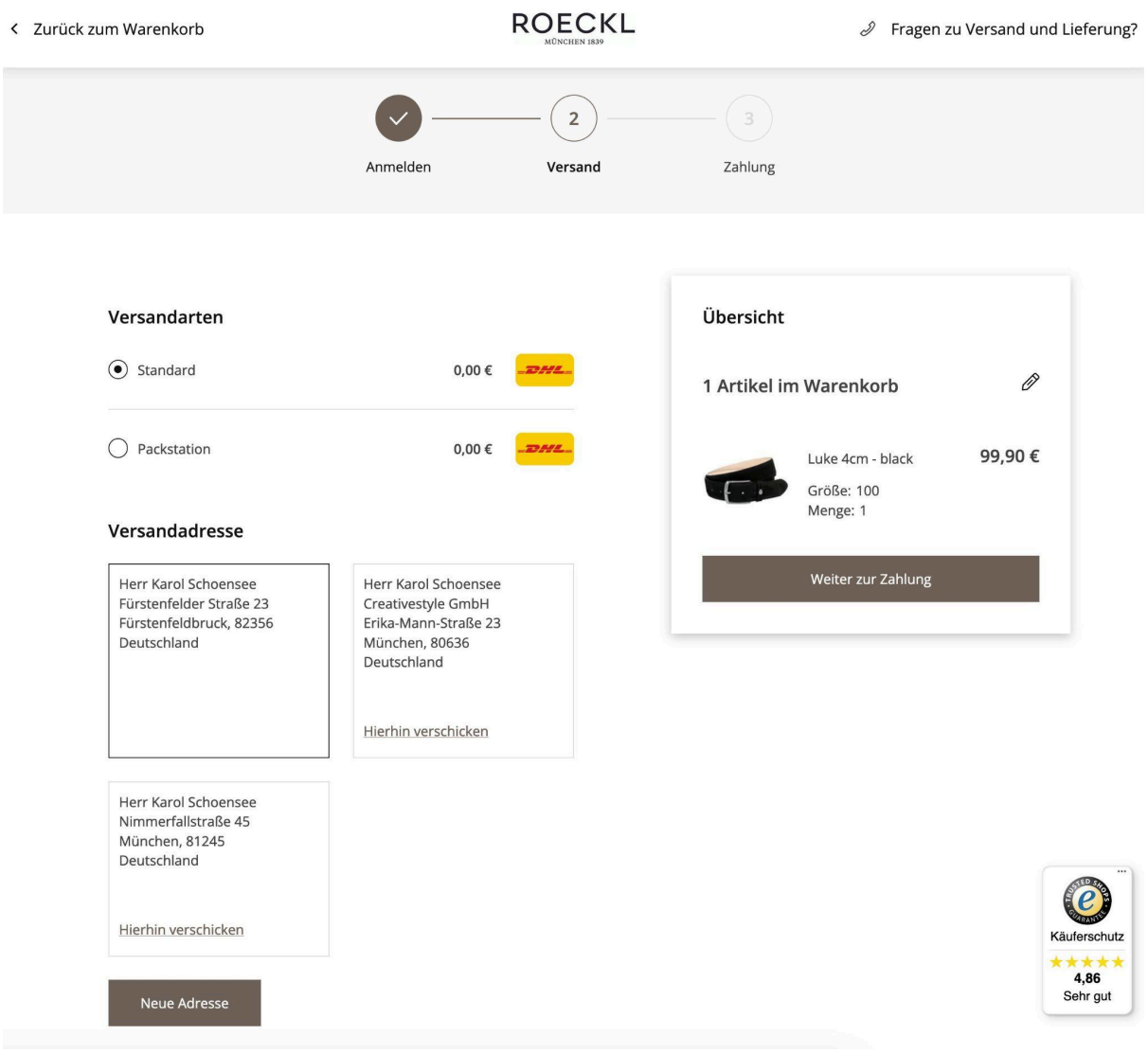
- Aus wie vielen Schritten besteht der Check-out? Ist es ein “1-Step” Check-out, sind es zwei oder drei Schritte?
- Wie kann der Nutzer zwischen einem Gast-Checkout, Kundenlogin und Kundenkonto-Registrierung wählen? An welcher Stelle gibt der Nutzer seine E-Mail-Adresse ein? Gibt es eine Validierung der E-Mail-Adresse? Gibt es ggf. eine Social Login Funktion (Login mit dem Facebook, Google oder Apple Account)?



Check-out Schritt 1 (Login) auf roeckl.com

- Falls der Nutzer als Gast fortfährt und seine Lieferadresse eingibt, welche Felder sind enthalten? Werden Straße und Hausnummer mit zwei Feldern getrennt? Gibt es eine Validierung (oder sogar Autosuggest) für z.B. Straße, Hausnummer, PLZ und Stadt? Kann der Nutzer das Autofill Feature des Browsers nutzen? Gibt es Felder für “Firma” und die “Umsatzsteuer-ID”? Welche Felder sind optional und welche erforderlich?


Anmelden
2
Versand
3
Zahlung
Versandarten
Standard
0,00 €
Packstation
0,00 €
Versandadresse
Frau
Herr
Divers
Vorname *
Karol
Nachname *
Schoensee
Firma
Straße und Hausnummer *
Fürstenfelder Str. 8
Postleitzahl *
82356
Ort *
Fürstenfeldbruck
Land *
Deutschland
Telefonnummer
*Die Rechnungsadresse kann im nächsten Schritt, beim Auswählen der jeweiligen Zahlungsmethode, geändert werden.
Übersicht
1 Artikel im Warenkorb
Luke 4cm - black
99,90 €
Größe: 100
Menge: 1
Weiter zur Zahlung

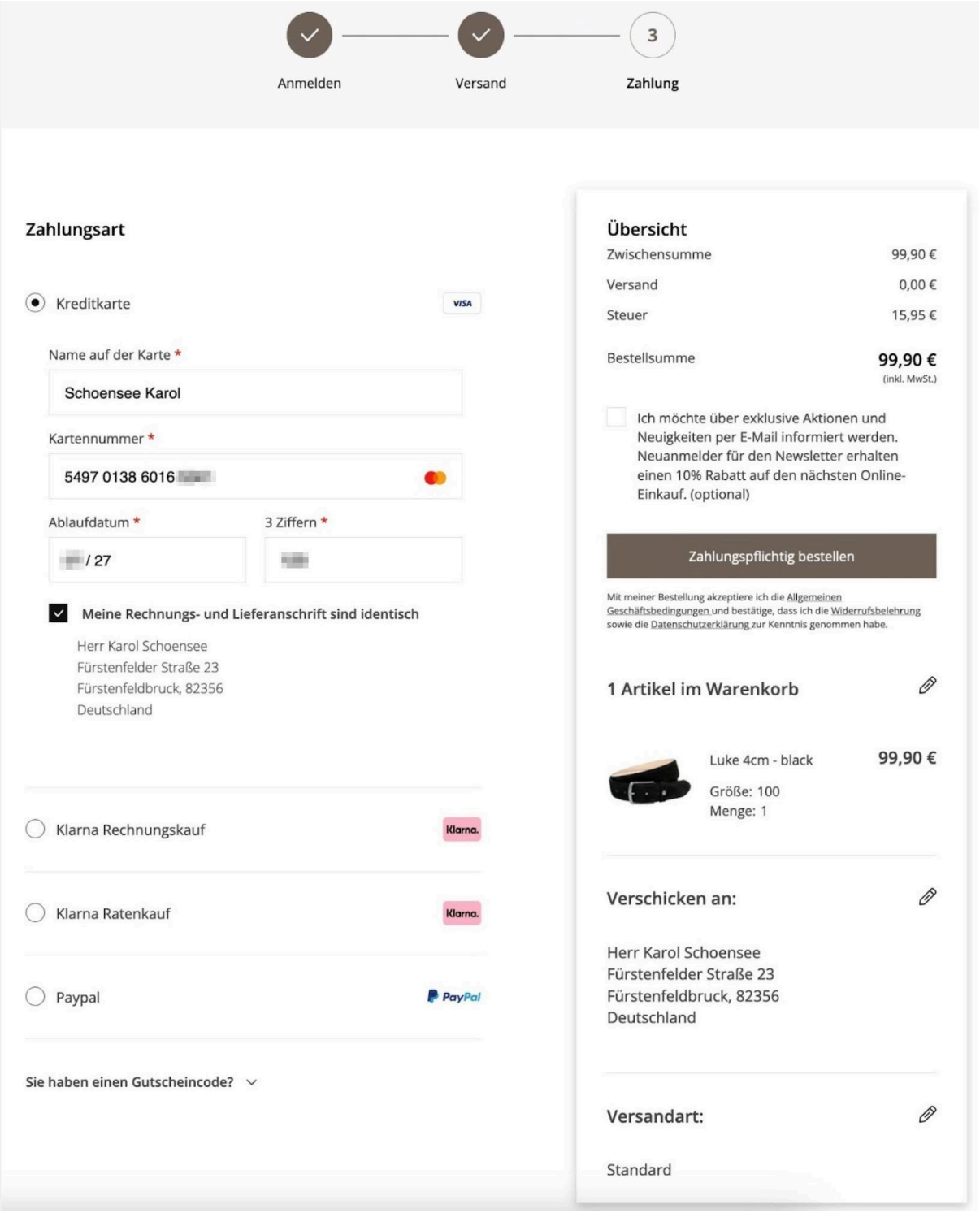


Check-out Schritt 2 als Gastnutzer (Versand und Lieferadresse) auf roeckl.com

Check-out Schritt 2 als eingeloggter Nutzer (Versand und Lieferadresse) auf roeckl.com

- Falls der Nutzer im eingeloggten Zustand fortfährt, kann er seine abgespeicherten Adressen auswählen? Kann er die gespeicherten Adressen ändern, löschen und neue Adressen hinzufügen?
- Kann der Nutzer seine Lieferadresse als Rechnungsadresse übernehmen oder muss er diese neu eingeben?
- Welche Versandmethoden und-kosten sind verfügbar? Gibt es ersichtliche Regeln, welche Versandmethoden verfügbar sind oder ab wann ein kostenfreier Versand gilt? Kann der Nutzer den Versand an eine Packstation auswählen? Gibt es “Express” Versandmethoden oder “Abholung in der Filiale”? Falls es die Versandmethoden Packstation und Abholung gibt, kann der Nutzer nach den gewünschten Filialen bzw. Packstationen suchen? Wie erfolgt die Eingabe der DHL Kundennummer?

- Welche Zahlungsmethoden stehen dem Nutzer zur Verfügung? Gibt es Aufpreise für bestimmte Zahlungsmethoden? Falls Kreditkarten genutzt werden können, findet die Eingabe im E-Commerce Storefront oder auf einer externen Seite bzw. Pop-up statt? Können die Kreditkartendaten für zukünftige Bestellungen abgespeichert werden? Kann der Nutzer das Autofill Feature vom Browser nutzen?

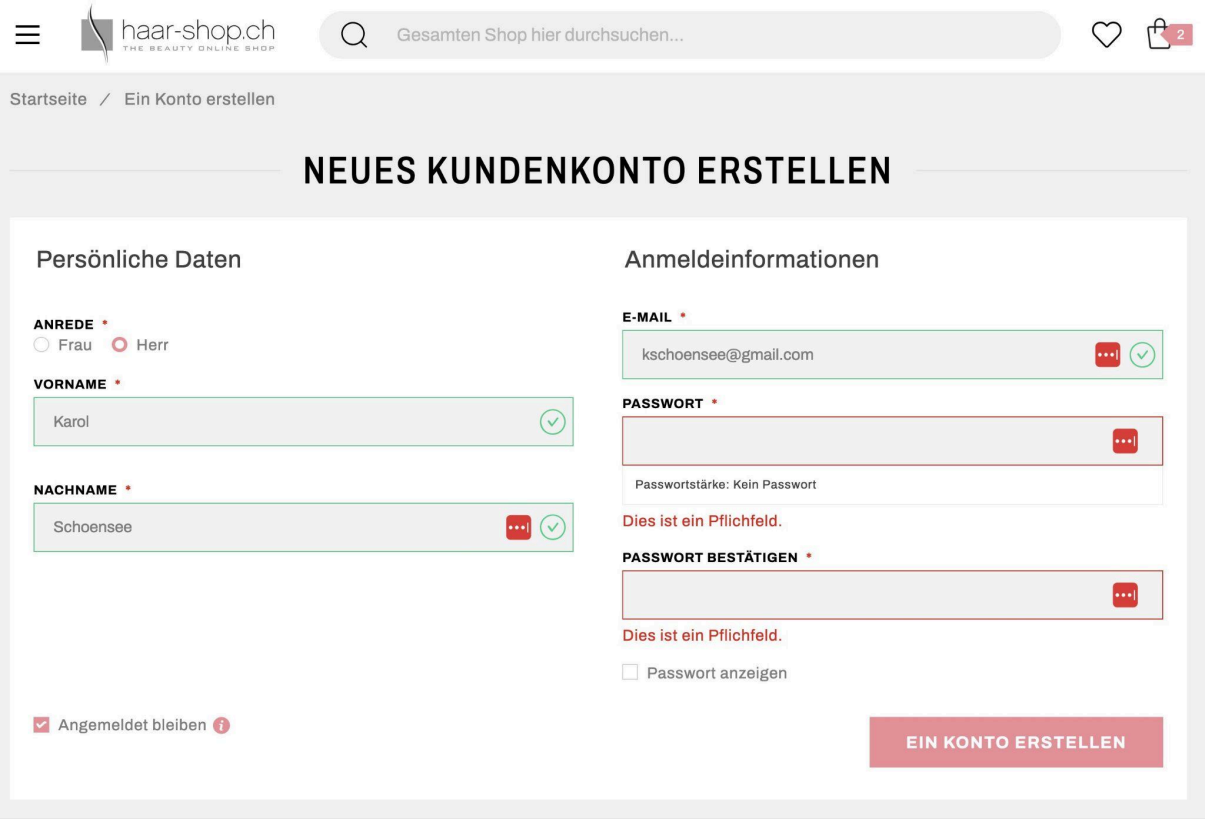


Check-out Schritt 3 (Zahlungsmethode) auf roeckl.com

- Kann der Nutzer einen Discount Code oder Gutscheincode eingeben? Wenn ja, an welcher Stelle erscheint der genutzte Code und der Rabatt in absoluten Beträgen?
- Gibt es eine erforderliche Checkbox zur Einverständniserklärung der AGB, des Widerrufsrechts und der Datenschutzerklärung vor dem Abschluss des Check-outs? Gibt es optionales Feld für das Newsletter Abo?
- Welche Informationen sind auf der Bestellbestätigungsseite enthalten? Kann der Nutzer seine bestellten Produkte und die Beträge einsehen? Kann er seine Bestellung drucken? Gibt es eine Möglichkeit ein Kundenkonto zu erstellen, falls der Nutzer den Check-out als Gast abgeschlossen hat?

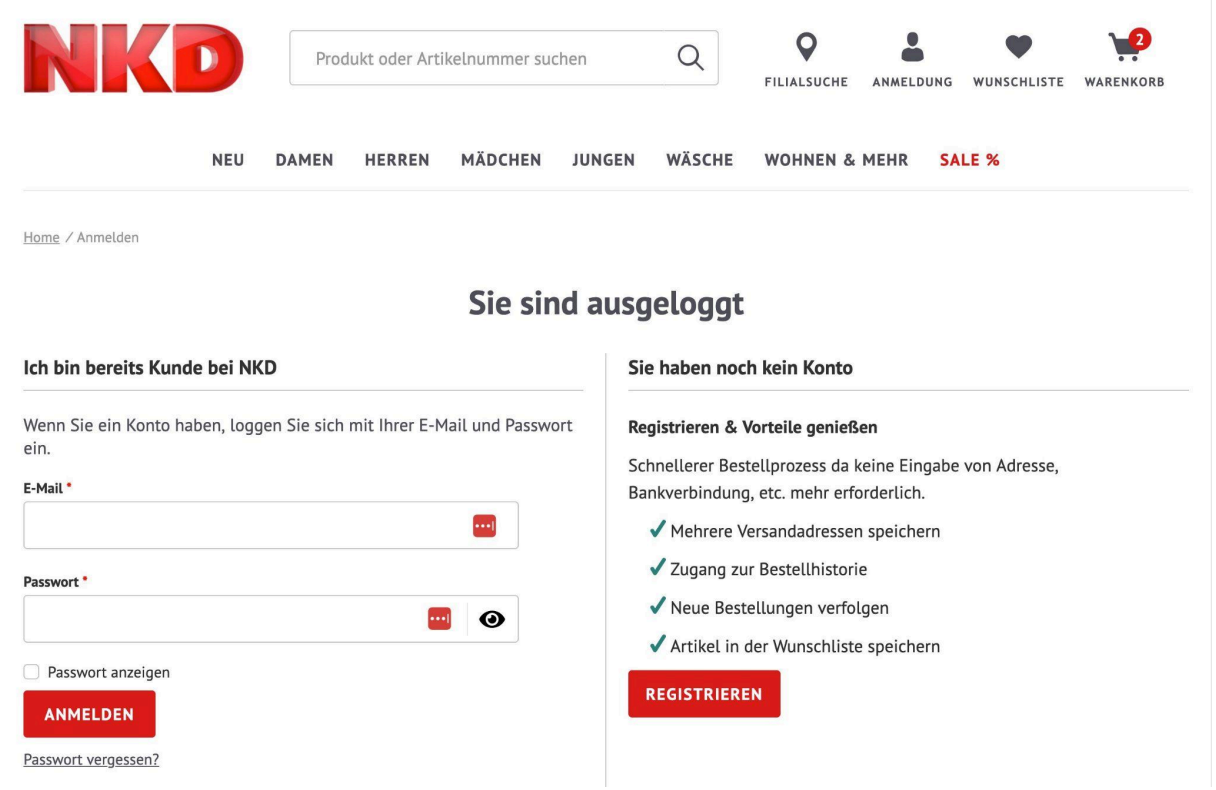
Mit einem Kundenkonto haben Nutzer die Möglichkeit, einen Login zu erstellen und in ihrem persönlichen Kundenbereich Bestellungen einzusehen, die Adressen anzupassen sowie Benachrichtigungen zu verwalten. Hier sollte beschrieben werden:

- Wie erreicht der Nutzer das Kundenregistrierungsformular? Welche Felder sind hier als optional und erforderlich enthalten? Kann sich der Nutzer optional zum Newsletter anmelden? Gibt es Validierungen in den Feldern, z.B. bei der E-Mail-Adresse oder der Telefonnummer? Welche Anforderungen gibt es an die Passwortvergabe? Gibt es einen Double-Opt-in, sodass der Nutzer sein Kundenkonto zunächst per E-Mail aktivieren muss? Kann der Nutzer die Autofill Option vom Browser nutzen? Wird die Registrierung mittels einem Captcha Mechanismus vor Bots geschützt?



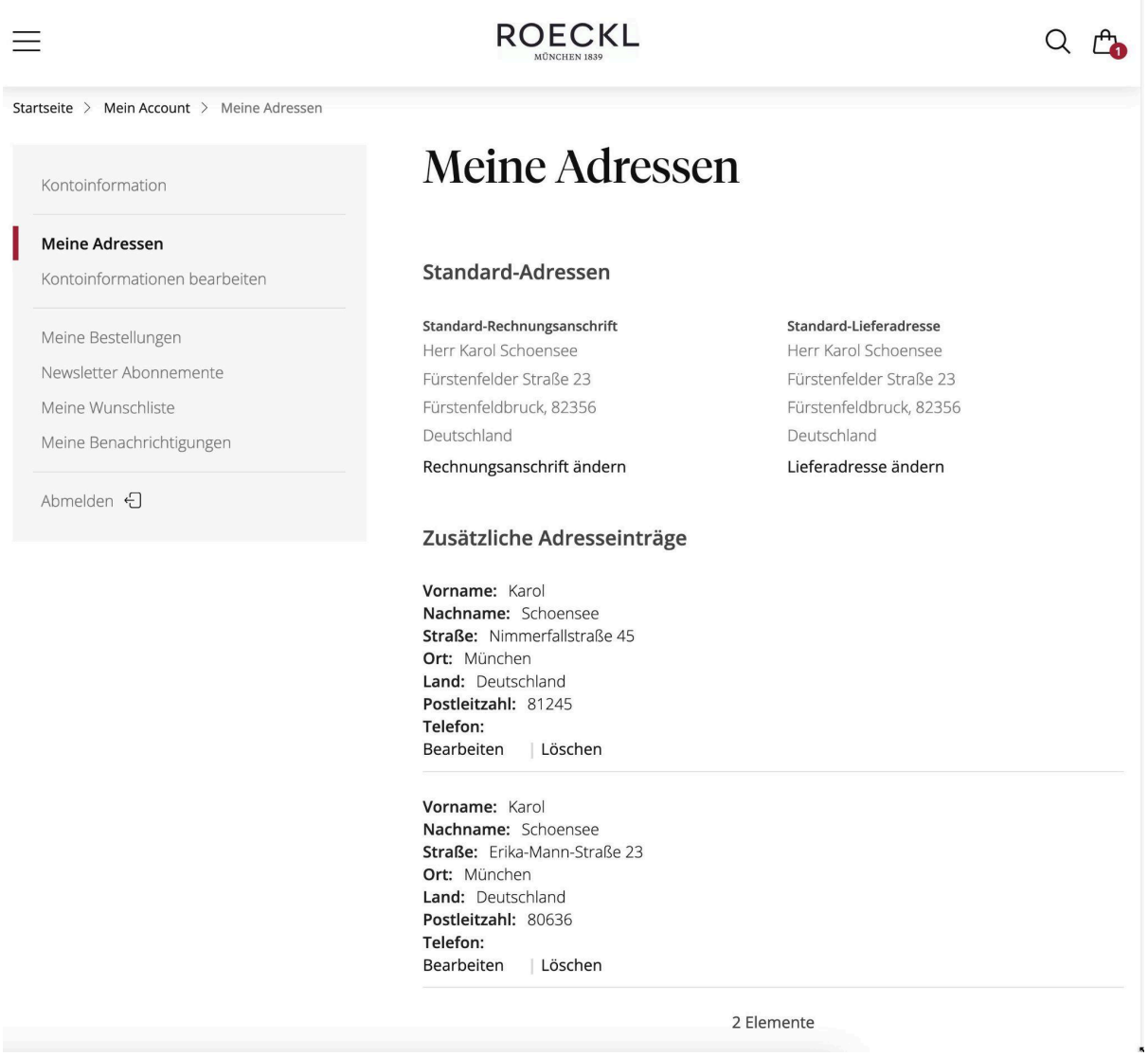
Kundenkonto Registrierung auf haar-shop.ch

- Wie kann sich der Nutzer in sein bestehendes Konto einloggen? Gibt es eine maximale Anzahl an Fehlversuchen, die das Kundenkonto sperrt? Kann der Nutzer sein Passwort anzeigen lassen oder ist es immer trunkiert? Wird dem Nutzer eine “Passwort vergessen” Funktion geboten?

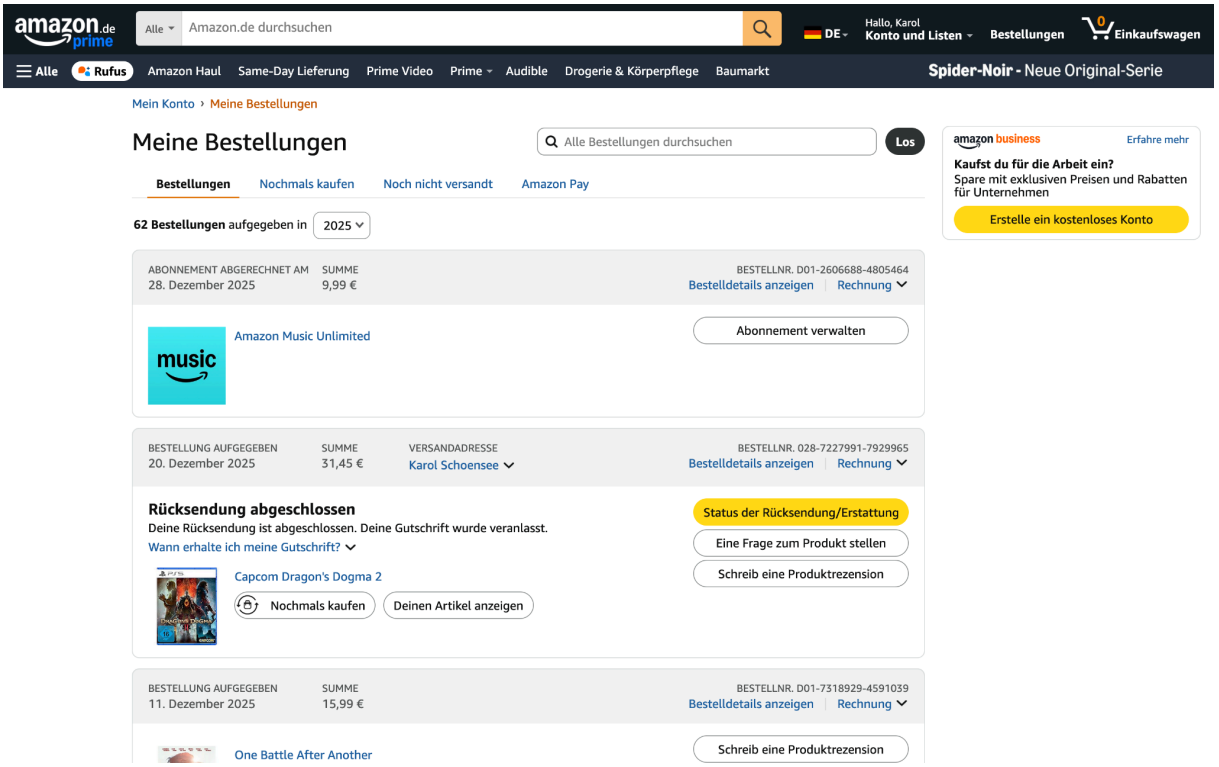


Kundenlogin-Seite auf nkd.com

- Kann der Nutzer seine Benachrichtigungen verwalten, wie z.B. E-Mail, SMS oder Push-Benachrichtigungen? Kann der Nutzer granular steuern, für welche Benachrichtigungen oder Newsletter er sich an- und abmelden möchte? Kann der Nutzer seine geschriebenen Kundenbewertungen und seine Wunschliste einsehen?
- Welche Bereiche stehen dem Nutzer nach erfolgreichem Login zur Verfügung? Kann der Nutzer seine Logindaten (E-Mail-Adresse und Passwort) ändern? Kann der Nutzer bestehende Liefer- und Rechnungsadressen ändern, löschen und neue Adressen hinzufügen? Gibt es, analog zum Check-out, Validierungen der Adressen oder eine Autosuggest-Funktion? Kann der Nutzer seine Bestellhistorie inkl. der Produkte, Mengen, Preise und eventueller Rabatte (Gutscheincodes) einsehen? Sieht der Nutzer ebenfalls den Status der Bestellungen, wie z.B. “im Versand”, “storniert” oder “geliefert”? Wird dem Nutzer eine Möglichkeit geboten, die Rechnung und den Versand Tracking Link einzusehen? Gibt es eine einfache und schnelle Möglichkeit, die Bestellung oder einzelne Produkte weiterzubestellen?



Adressbuch im Kundenbereich auf roeckl.com



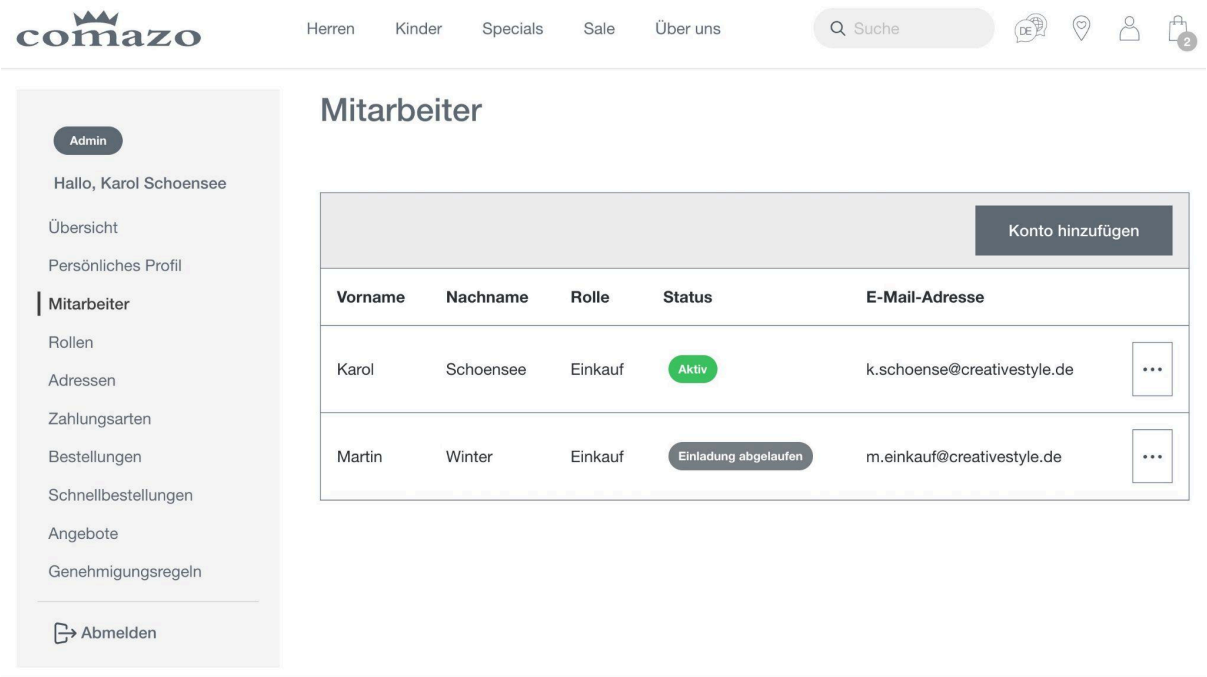
Bestellhistorie im Kundenbereich auf amazon.de

- Gibt es ein Kontaktformular, über das der Nutzer den E-Commerce Owner bzgl. Fragen oder Beschwerden kontaktieren kann? Welche Felder sind in diesem Kontaktformular enthalten und gibt es ggf. Validierungen oder ein Captcha als Spam-Schutz?

Die bisher besprochenen Bereiche und Funktionen können als Grundfunktionen eines Online-Shops angesehen werden, vor allem im Business-to-Consumer (B2C) Kontext. Falls die E-Commerce Plattform auf gewerbliche Kunden, also private Unternehmen, Verbände, Vereine, und sonstige Organisationen, ausgerichtet ist, sollten ebenfalls Funktionen aus dem Business-to-Business (B2B) Kontext beschrieben sein. Dazu gehören etwa:

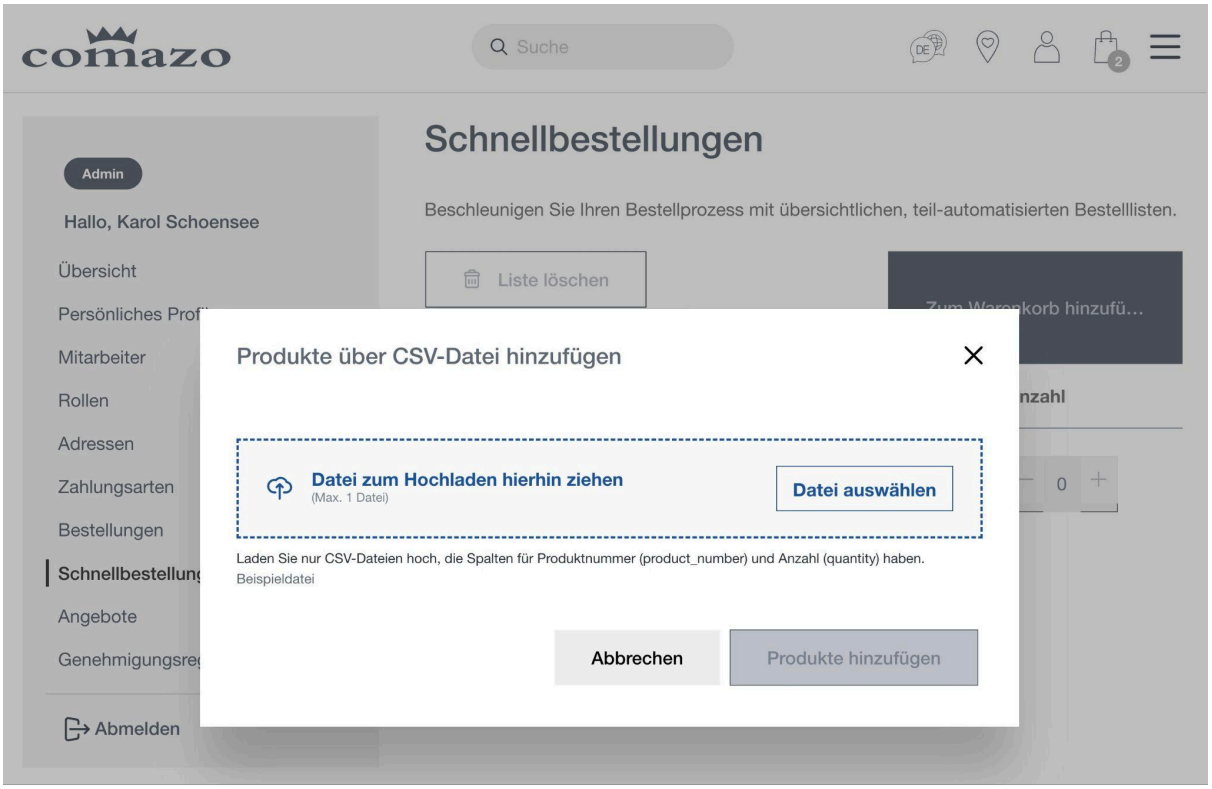
- Gibt es im Katalog (also auf der Kategorie- und Produktdetailseite) Staffelpreise, bei denen die Stückpreise gestaffelt mit höherer Menge abnehmen? Haben die eingeloggten Kunden jeweils individuelle Preise?
- Sind die Preise in netto oder brutto (inkl. MwSt.) aufgeführt? Gibt es im Check-out eine Logik, z.B. mit den Feldern “Firma” und “Umsatzsteuer-ID”, die Steuern vom Gesamtbetrag abzieht?
- Gibt es eine Freigabe der Kundenregistrierung, z.B. mittels Upload von einem Gewerbeschein oder Ähnlichem?

- Hat der Kunde eine Möglichkeit, den Warenkorb zu verhandeln? Ist diese Möglichkeit erst ab einer gewissen Stückzahl der enthaltenen Produkte oder vom Warenkorbwert abhängig? Kann der Nutzer bei der Angebotsanfrage bereits Stückpreise angeben und ggf. eine Nachricht hinterlassen? Kann der Nutzer in seinem Kundenbereich die Angebotsanfragen inkl. dem Warenkorb, dem Status und der Rückmeldung vom E-Commerce Owner einsehen?
- Kann der eingeloggte Kunde weitere Mitarbeiter einladen oder Unterkonten erstellen? Wie unterscheiden sich die einzelnen Unterkonten voneinander? Sind es jeweils unterschiedliche Namen, E-Mail-Adressen, und Lieferadressen? Bezieht sich die Rechnungsadresse auf alle Bestellungen, inkl. aller Unterkonten?
- Haben die Unterkonten bestimmte Rechte und Rollen? Denkbar ist, dass z.B. ein bestimmter Nutzer ("Assistenz") den Warenkorb nur befüllen, aber den Check-out nicht abschließen kann. Ein Nutzer mit höheren Rechten ("Geschäftsführung") kann hingegen den Check-out abschließen.



Mitarbeiter Management im Kundenbereich auf comazo.de

- Kann der Nutzer den Warenkorb per Upload einer Datei, z.B. CSV oder Excel, befüllen? Wenn ja, welche Informationen werden in der Datei erwartet? Sind es nur die SKU und die Stückzahl?

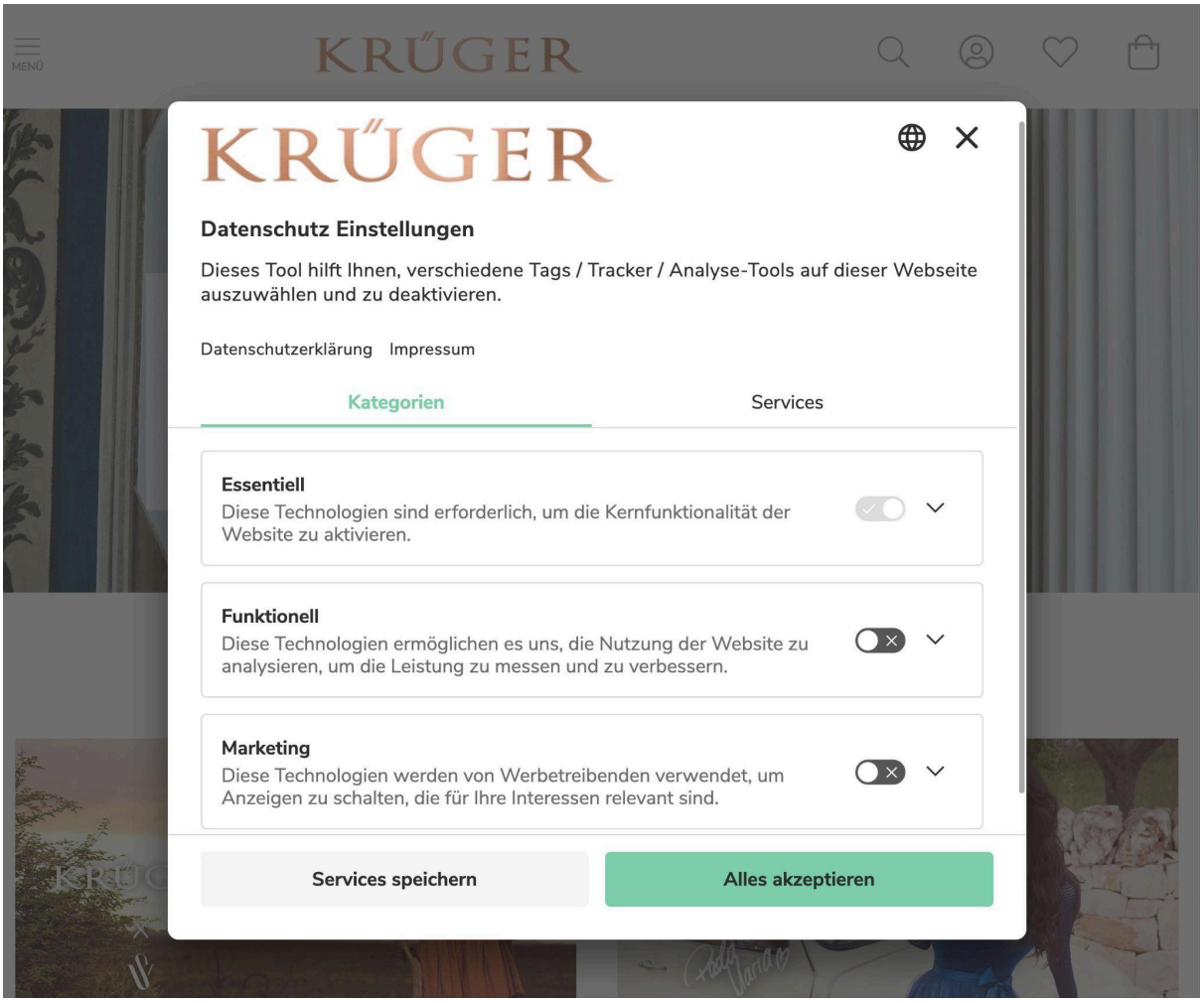


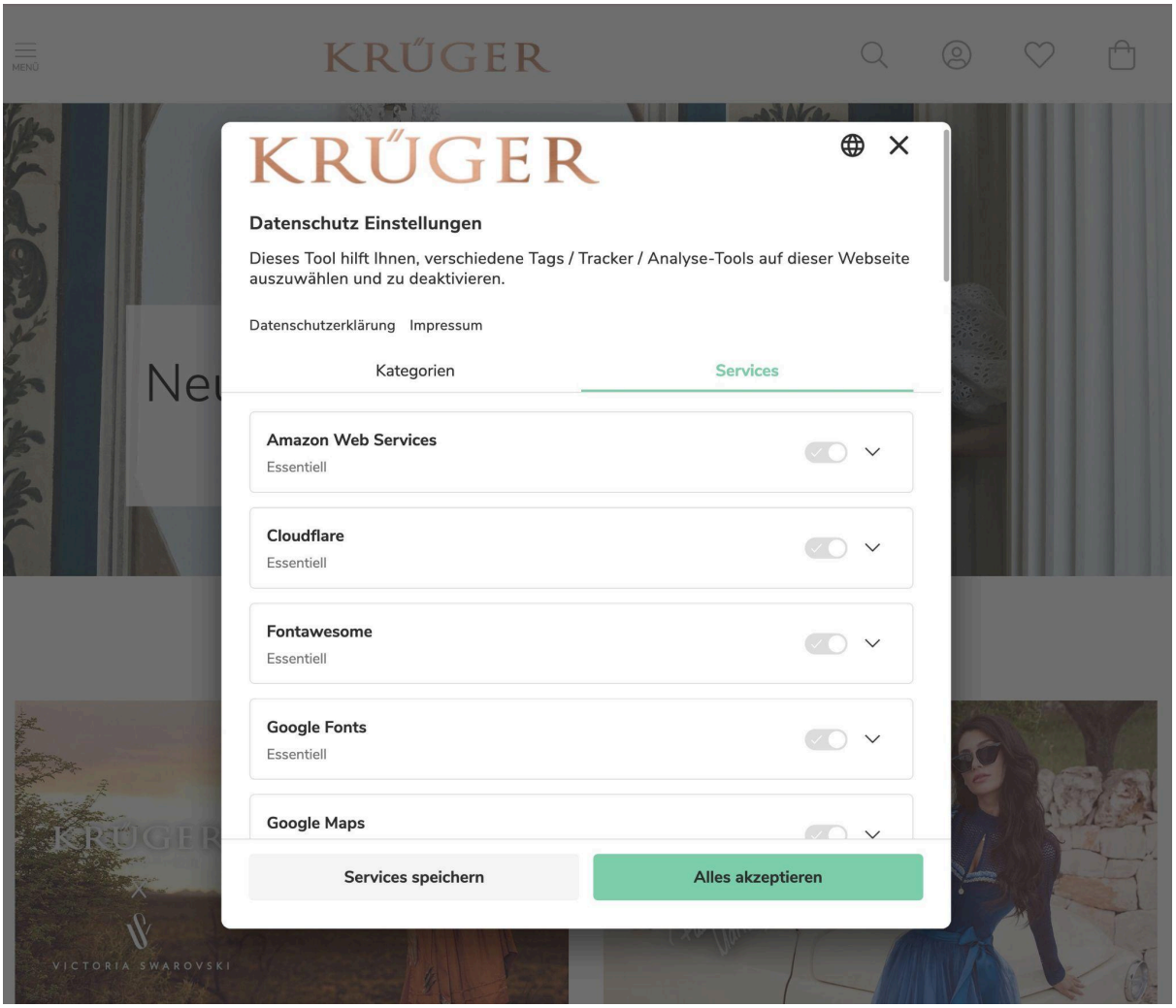
Schnellbestellung per CSV Import im Kundenbereich auf comazo.de

- Gibt es im Kundenbereich oder an sonstiger Stelle einen “Self-Service” Bereich, in dem der Nutzer z.B. passende Ersatzteile finden, Datenblätter, Garantieblätter oder andere Dokumente einsehen kann? Kann der Kunde ein Software-Update seiner Produkte bzw. ein Training buchen?

Neben den offensichtlichen Funktionen im Storefront sollten ebenso die weniger offensichtlichen, aber aus rechtlicher Sicht wichtigen, DSGVO relevanten Funktionen nicht fehlen. Darunter gehören etwa:

- Gibt es zum Ersteinstieg ein User Consent, bei dem der Nutzer seine Einwilligung für die Nutzung von Cookies und ähnlichen Technologien Dritter (z.B. Google Analytics oder Tracking Marketing Pixel) geben kann? Kann der Nutzer diese Dienste kategorisiert, z.B. nach “Essentiell”, “Marketing”, “Funktionell”, “Statistik”, ablehnen oder akzeptieren? Kann der Nutzer alternativ die einzelnen Dienste seperat ablehnen oder akzeptieren?

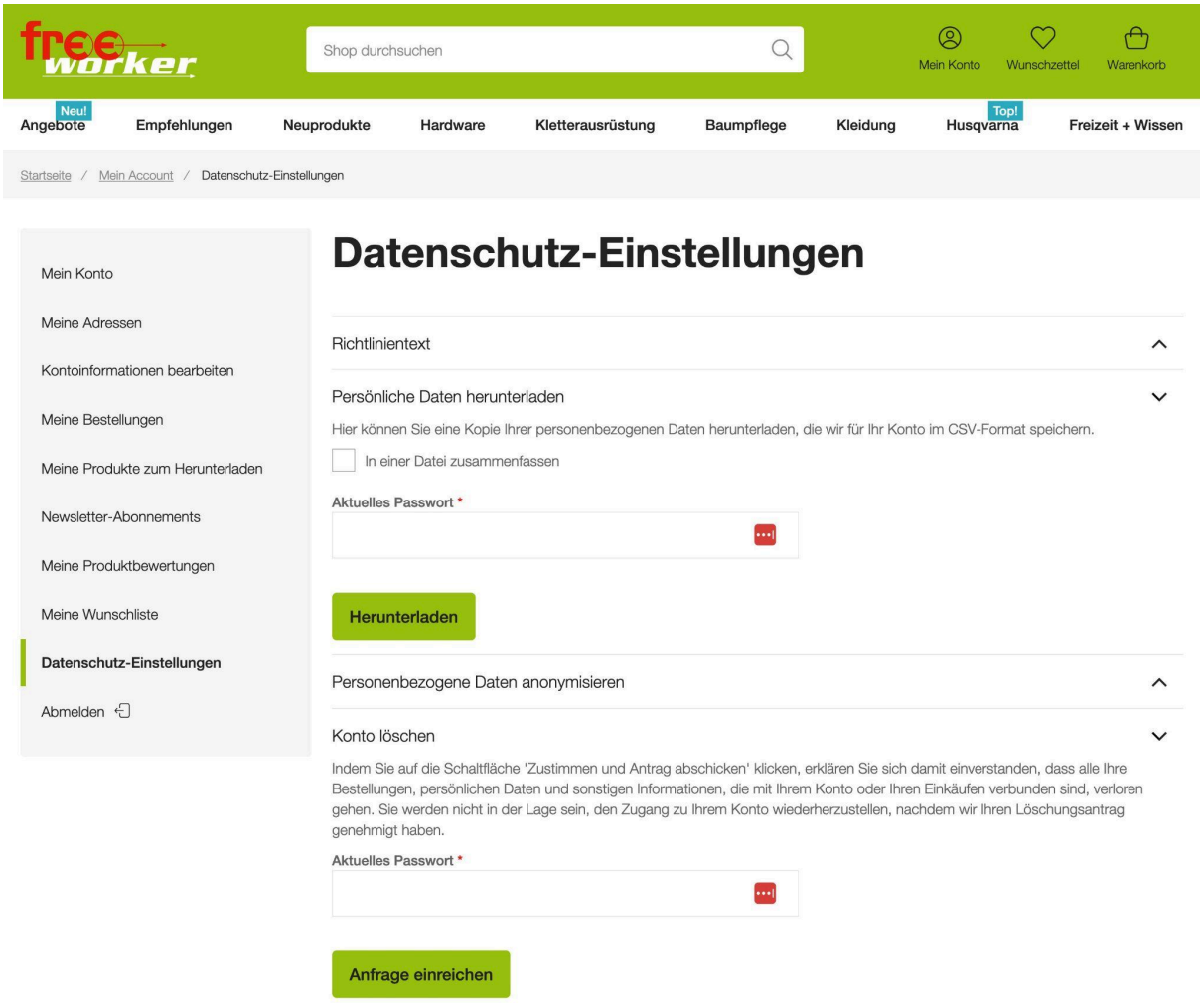

MENÜ
KRÜGER
KRÜGER
Datenschutz Einstellungen
Dieses Tool hilft Ihnen, verschiedene Tags / Tracker / Analyse-Tools auf dieser Webseite auszuwählen und zu deaktivieren.
Datenschutzerklärung
Impressum
Kategorien
Services
Essentiell
Diese Technologien sind erforderlich, um die Kernfunktionalität der Website zu aktivieren.
Funktionell
Diese Technologien ermöglichen es uns, die Nutzung der Website zu analysieren, um die Leistung zu messen und zu verbessern.
Marketing
Diese Technologien werden von Werbetreibenden verwendet, um Anzeigen zu schalten, die für Ihre Interessen relevant sind.
Services speichern
Alles akzeptieren



User Consent kategorisiert auf krueger-dirndl.de

User Consent nach einzelnen Services auf krueger-dirndl.de

- Hat der Nutzer im Sinne von DSGVO Kapitel 3 eine Möglichkeit, seine personenbezogenen Daten einzusehen, zu löschen und umzuziehen?
- Kann der Nutzer nachträglich, z.B. im Kundenbereich, seine Datenschutzeinstellungen sowie seine Benachrichtigungen (z.B. Newsletter) ändern?



Datenschutzeinstellungen im Kundenbereich auf shop.freeworker.de

Schließlich sollte auch geprüft werden, welche Benachrichtigungen vom E-Commerce System selbst an den Nutzer abgeschickt werden. Darunter fallen E-Mails, SMS sowie Push-Benachrichtigungen. Einige Beispiele dafür sind:

- Verschickt das E-Commerce System nach erfolgreichem Check-out eine Bestellbestätigung? Welche Informationen sind hier enthalten? Sind es die Daten des Nutzer (z.B. Adresse, Name, usw.), die bestellten Artikel sowie die Gesamtbeträge inkl. Versandkosten und ggf. Rabatte (genutzter Rabattcodes)?

Karol Schoensee,

Vielen Dank für Ihre Bestellung bei Freeworker. Sobald Ihr Paket verschickt wurde, senden wir Ihnen den Link zur Sendungsverfolgung per E-Mail. Wenn Sie Fragen bezüglich Ihrer Bestellung haben, können Sie uns über die E-Mail-Adresse info@freeworker.de kontaktieren oder rufe uns an +49 (0) 8105 27 27 27. Unsere Öffnungszeiten sind Mo - Fr: 08:00 - 18:00 Uhr.

Ihre Bestellung #1000239925

In Auftrag gegeben am 06.10.2024, 13:04:20

Zahlungsinformationen

Karol Schoensee
Fürstenfelder [redacted]
Fürstenfeldbruck, [redacted]
Deutschland
T: 0176181845

Versand-Informationen

Karol Schoensee
Fürstenfelder [redacted]
Fürstenfeldbruck, [redacted]
Deutschland
T: 0176181845

Zahlungsart

Vorkasse

ACHTUNG: Überweisung erst nach Erhalt unseres Vorkassenbelegs tätigen!

Versandart

Versandmethoden - DPD bis 5 Kg: Lieferzeit ~2 Tage

Artikel	Anzahl	Preis
Toolprotect Helmhalter Protos Auf Lager Nur noch 4 auf Lager! Es kann passieren, dass ein anderer Interessent den Bestellprozess schneller durchläuft und die Anzeige der Restbestände sich auf dem Weg durch die Kasse noch ändert. Artikelnummer: KTP-PH-GY	1	**56,60 €**
Climb Now Work Later Hoodie Artikelnummer: HOCN-GY-M Kleidergröße M Farbe grau	1	**75,60 €**

Zw.summe	132,20 €
Versand & Bearbeitung	8,20 €
Gesamt (Netto)	**140,40 €**
MwSt. (19%)	26,68 €
Steuer	26,68 €
Gesamt (Brutto)	**167,08 €**

Karol Schoensee,

Vielen Dank für die Bestätigung Ihres Freeworker Kundenkontos.

Verwenden sie diese Zugangsdaten während der Zahlungsabwicklung oder auf der Mein Benutzerkonto Seite um sich einzuloggen:

- **E-Mail:** [illegible]
- **Passwort:** *Passwort, das beim Erstellen des Kontos festgelegt wurde*

Passwort vergessen? Klicken Sie hier um es zurückzusetzen.

Wenn Sie sich in Ihr Benutzerkonto einloggen können Sie:

- Die Bestellung schneller durchführen
- Status der Bestellungen prüfen
- Frühere Bestellungen anzeigen
- Alternative Lieferadressen speichern (für den Versand an mehrere Familienmitglieder und Freunde)

Freeworker
St. Gilgen 15
82205 Gilching

+49 (0) 8105 27 27 27

Mo - Fr: 08:00 - 18:00 Uhr

Climb up high! **Freeworker**

Bestellbestätigung per E-Mail auf shop.freeworker.de

Bestätigung der Kontoerstellung per E-Mail auf shop.freeworker.de

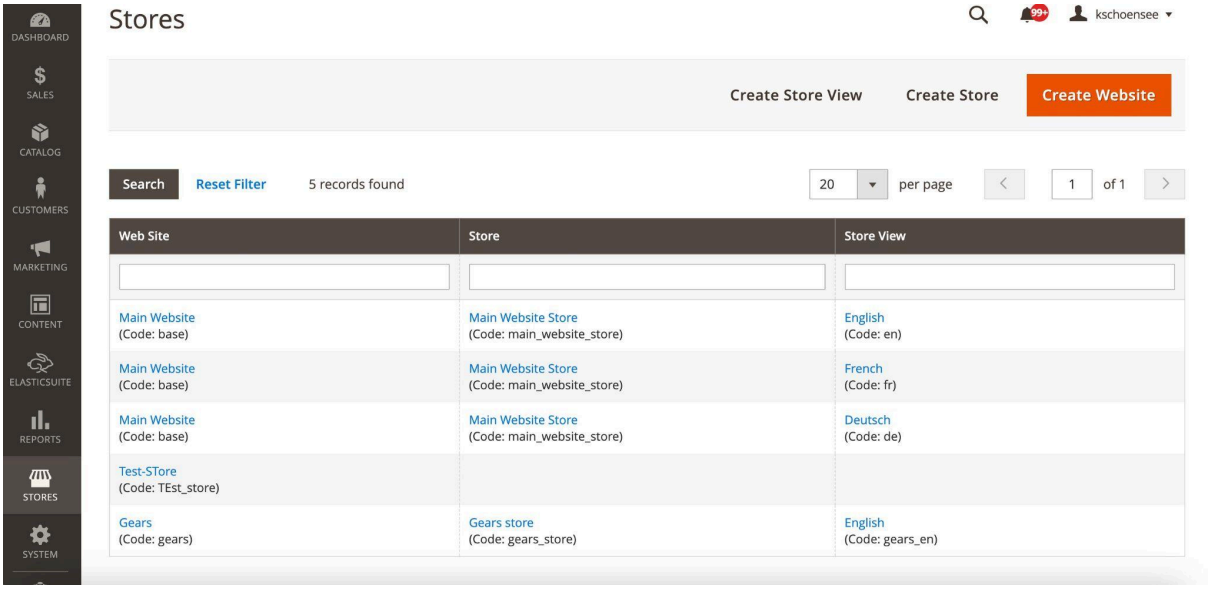
- Verschickt das E-Commerce System eine Nachricht nach erfolgtem Versand inkl. Tracking Link, oder ist es ein anderes System?
- Verschickt das E-Commerce System die Rechnung oder wird diese ggf. dem Paket beigefügt oder von einem anderen System verschickt?
- Verschickt das E-Commerce System eine Nachricht bei Nutzung der “Passwort vergessen” Funktion?
- Wird eine Double-Opt-in Nachricht verschickt, falls der Nutzer ein Kundenkonto anlegen oder sich zum Newsletter anmelden möchte?
- Was passiert im Falle von Beschwerden oder Retouren? Werden Nachrichten vom E-Commerce System oder einem anderen Dienst verschickt?

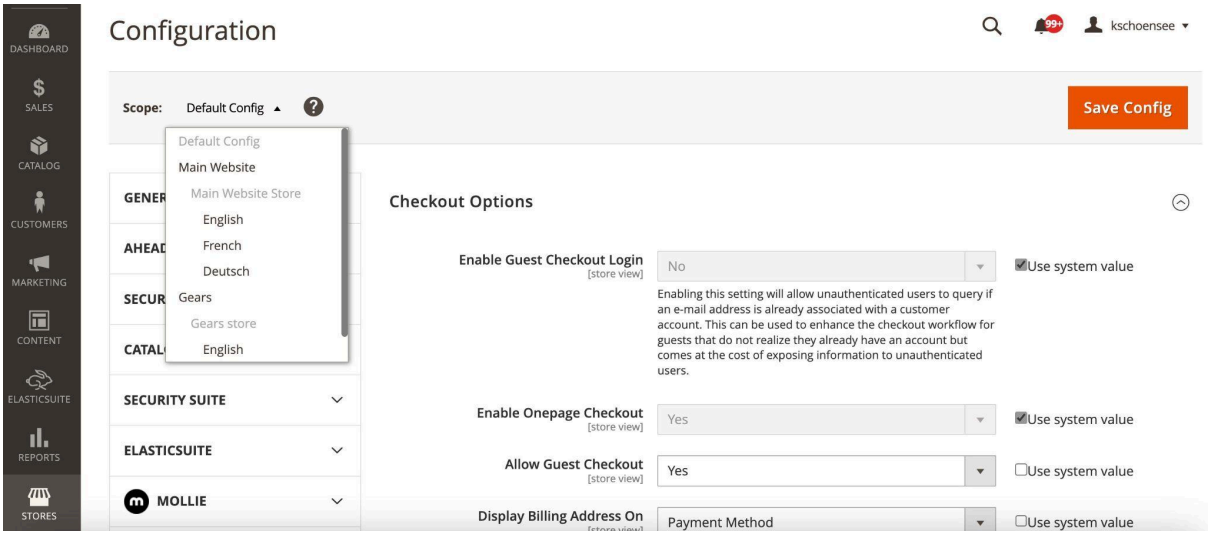
Funktionale Analyse Admin Bereich

Die funktionale Analyse des E-Commerce Admin Bereichs umfasst alle Funktionen, die nur für Administratoren des Systems mittels Login zugänglich sind.

Dabei sollte vor allem geprüft werden, die aktuelle E-Commerce-Plattform mandantenfähig ist oder mehrere "Sub-shops" unterstützt. Darunter fallen etwa die folgenden Aspekte:

- Kann man in einer E-Commerce-Instanz bzw. in einem Admin Bereich mehrere Mandanten oder Sub-Shops erstellen? Wenn ja, worin unterscheiden sich diese? Gibt es mehrere Ebenen oder eine Hierarchie der Sub-Shops (z.B. kann die tiefste Ebene nur unterschiedliche Sprachen oder Währungen unterscheiden, während die oberste Ebene die gesamte Shop-Konfiguration unterscheiden kann)?

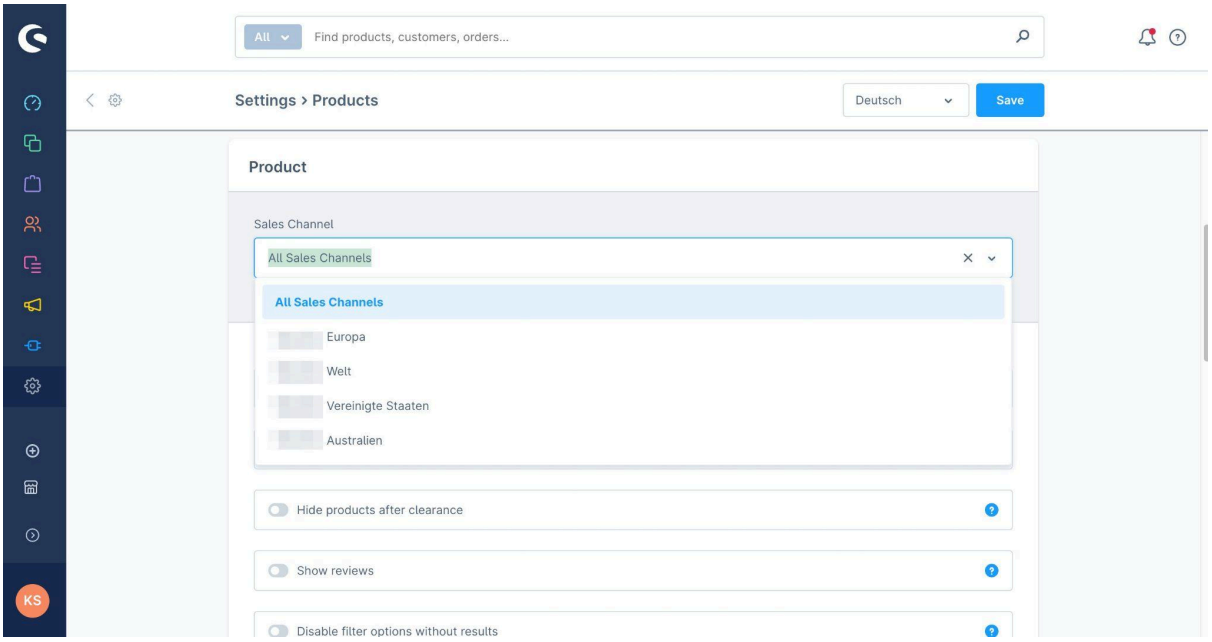




Die Multi-Store Architektur in Magento 2

Auswahl des Mandanten zur Konfiguration in Magento 2

- Können die einzelnen Sub-Shops unterschiedliche Storefront Templates (bzw. Themes) nutzen? Wie werden diese durch öffentliche Nutzer im Storefront erreicht? Sind es z.B. gänzlich unterschiedliche Top-Level-Domains oder werden ggf. nur unterschiedliche URL Sub-Folder unterstützt?
- Können die Sub-Shops gänzlich andere Kataloge (Attribute, Kategorien und Produkte) nutzen?
- Können lediglich bestimmte Shop Konfigurationen (z.B. der Check-out, wie Versand- und Zahlungsmethoden und Steuern) pro Sub-Shop gesteuert werden oder können grundsätzlich alle Konfigurationen pro Sub-Shop gesteuert werden?
- Unterstützt der Shop mehrere Währungen und Sprachen?
- Gibt es eine Defaultebene, deren Konfiguration auf die Sub-Shops vererbt wird?

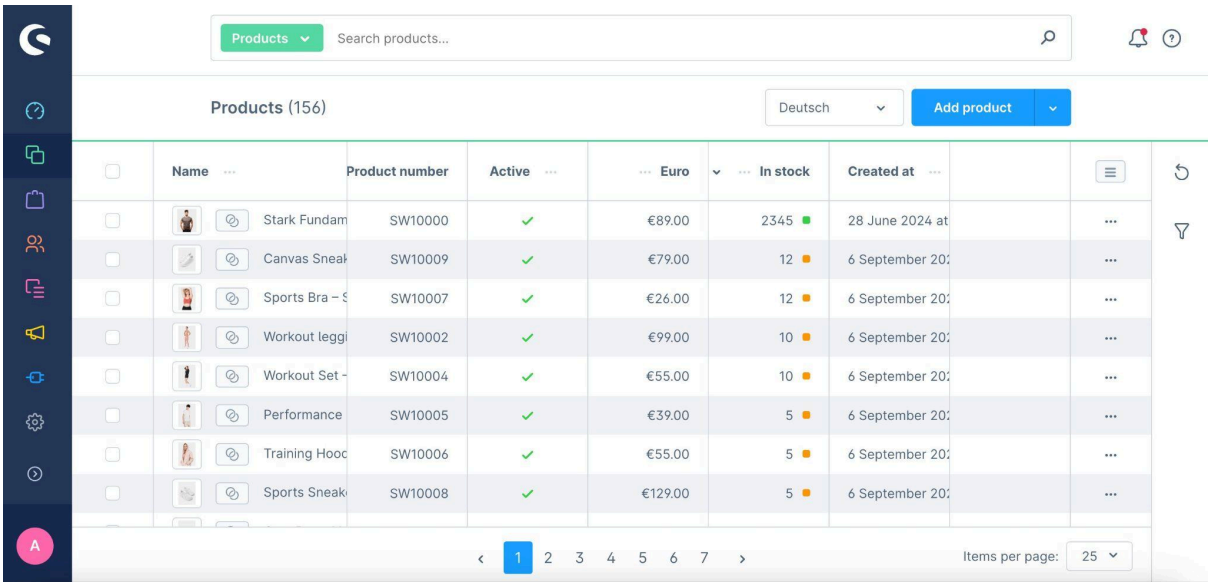


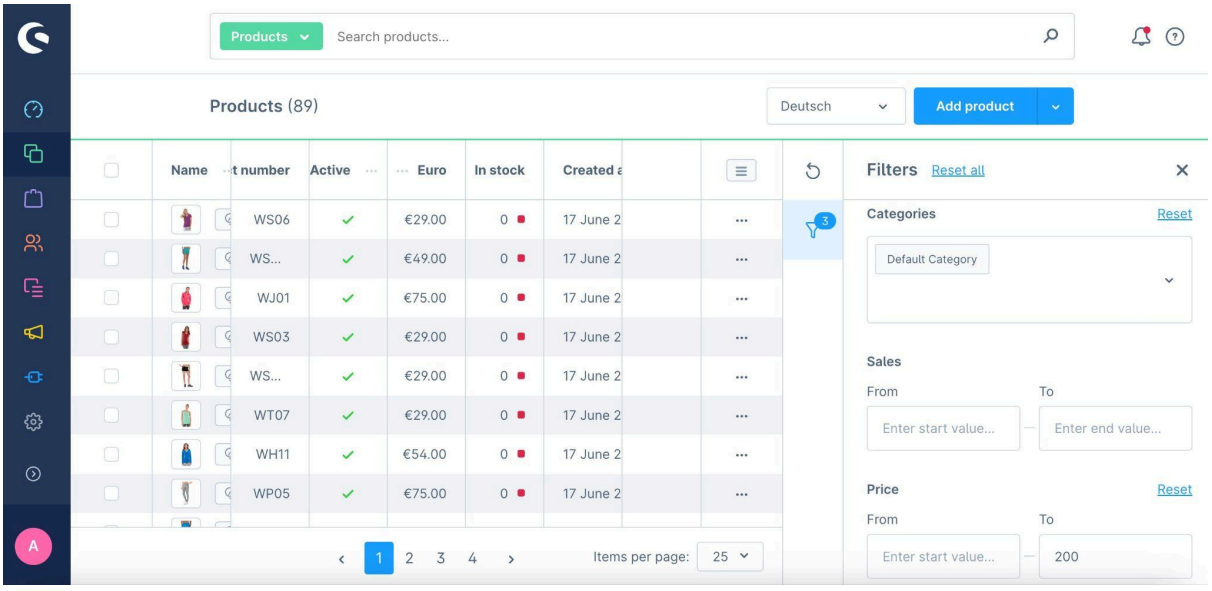
Konfiguration der Produktdarstellung im Mandanten (Sales Channel) "Storefront" und den Vererbungen aus der Defaultebene in Shopware 6

Einer der wichtigsten Bereiche der E-Commerce System Administration ist die Produktverwaltung. Hier sollte man alle im E-Commerce System befindlichen Produkte einsehen und bearbeiten können. Falls ein PIM System die Produkte steuert, sollten die entsprechenden Produkte, deren Attribute und Medien (z.B. Produktbilder) dennoch im Admin Bereich einsehbar sein. Bei der Analyse sollten die folgenden Punkte beachtet werden:

- Wie sieht die Produktübersicht aus und welche Funktionen gibt es? Kann man über eine Suche nach z.B. Produktname, SKU oder einem anderen Attribut gezielt

Produkte suchen? Gibt es eine Sortierung, die z.B. neuesten (oder ältesten) Produkte anzeigt oder alphabetisch nach Produktname oder SKU sortiert? Gibt es Filter, um z.B. nur “aktive” Produkte zu sehen?

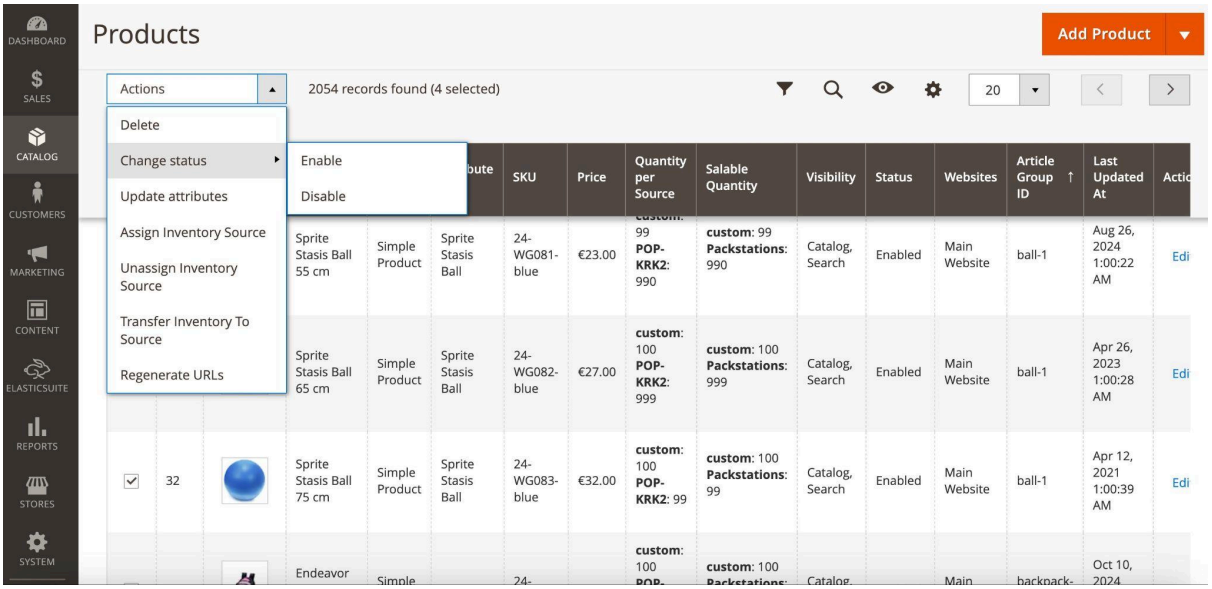


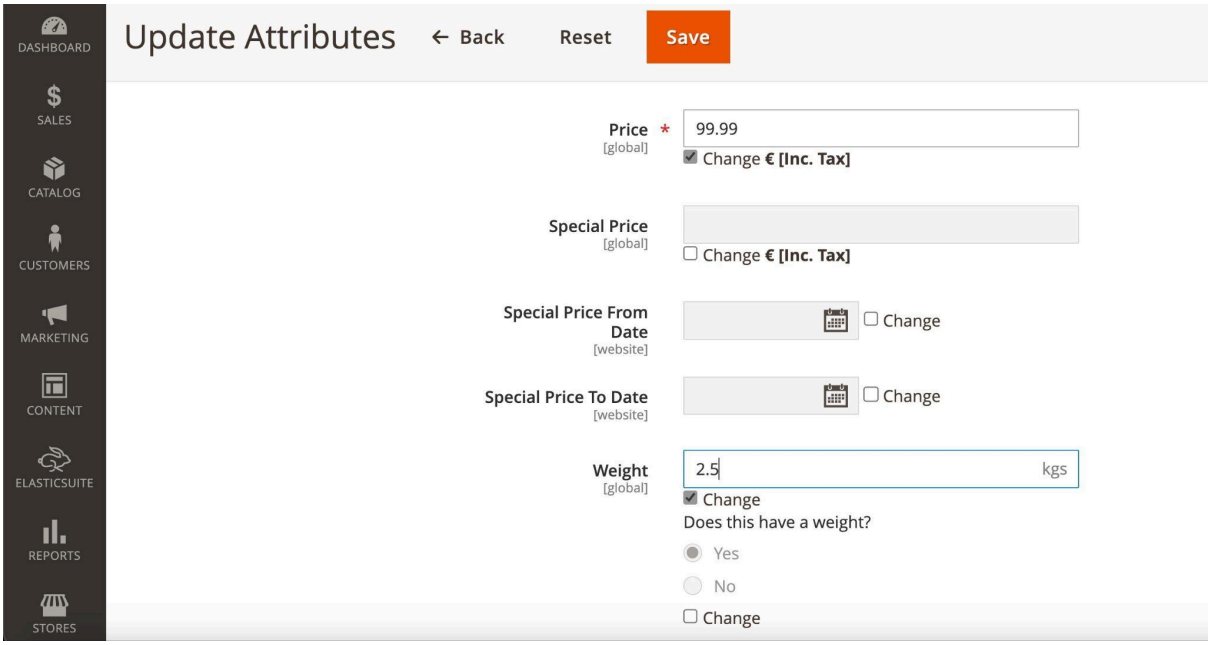


Produktübersicht inkl. Suche in Shopware 6

Produktübersicht mit angewandten Filtern in Shopware 6

- Gibt es eine Möglichkeit, mehrere Produkte gleichzeitig (“Mass Action” oder “Bulk Edit”) zu editieren? Können die entsprechenden Produkte gefiltert oder per Checkbox definiert werden? Welche Änderungen können hier getätigt werden? Können die Produkte lediglich gelöscht oder hinsichtlich des Status (inaktiv / aktiv) geändert werden? Oder können grundsätzlich alle Produktattribute geändert werden (z.B. Preis oder Produkteigenschaften wie “Material”)?

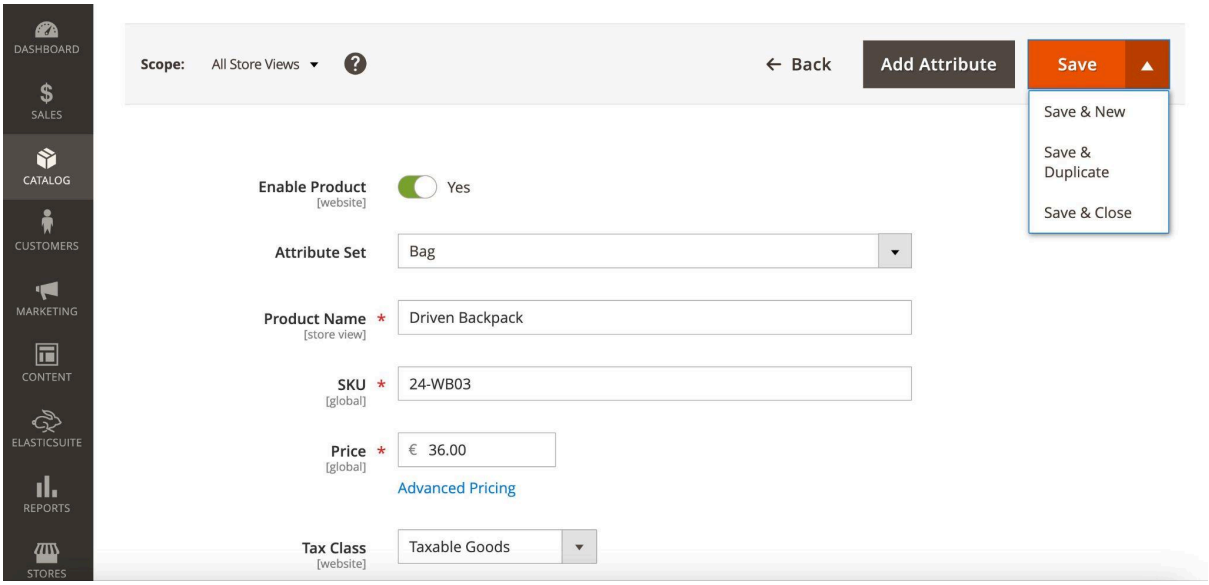




Produktauswahl für "Mass Actions"(z.B. Produkte löschen oder deren Status ändern) in Magento 2

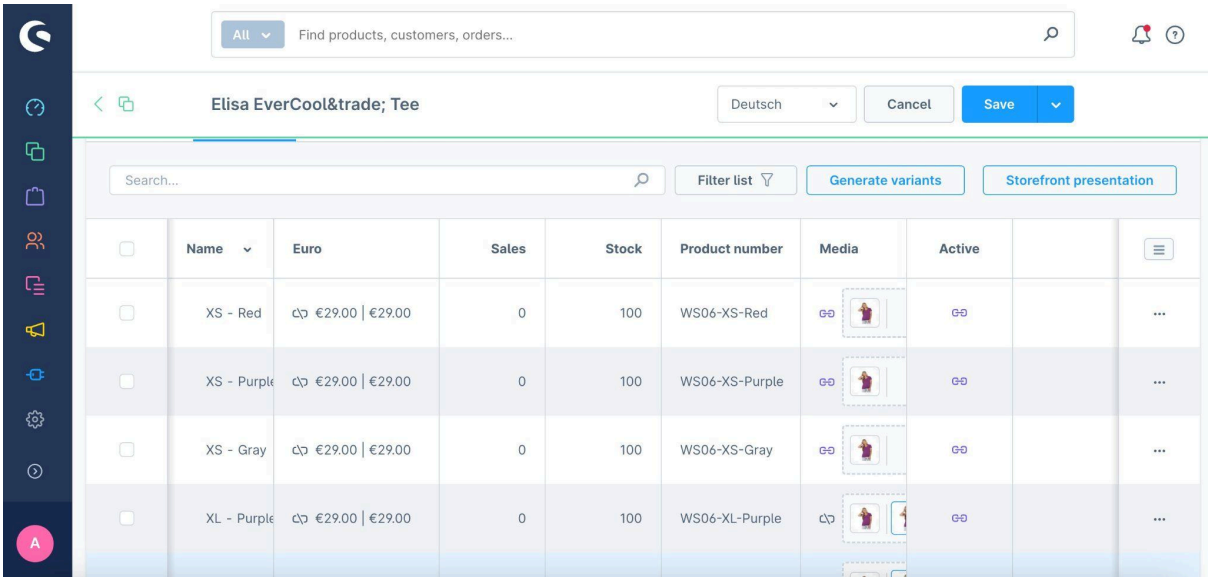
Änderung der Attribute "Preis" und "Gewicht" für mehrere Produkte in Magento 2

- Welche Spalten (bzw. Produktattribute) werden in der Produktübersicht dargestellt? Kann man beliebige Attribute ein- oder ausblenden sowie deren Reihenfolge steuern?
- Kann man Produkte exportieren und importieren?
- Welche Attribute kann der Admin in der Detailansicht einzelner Produkte sehen? Sind es lediglich SKU, Produktname, Status, Beschreibung, Warenbestand, Steuerklasse, URL und Produktbilder? Kann der Admin weitere Attribute eigenständig hinzufügen? Kann er bestehende Produkte duplizieren und die Werte von Duplikaten einzeln steuern?



Verwaltung einzelner Produkte in Magento 2

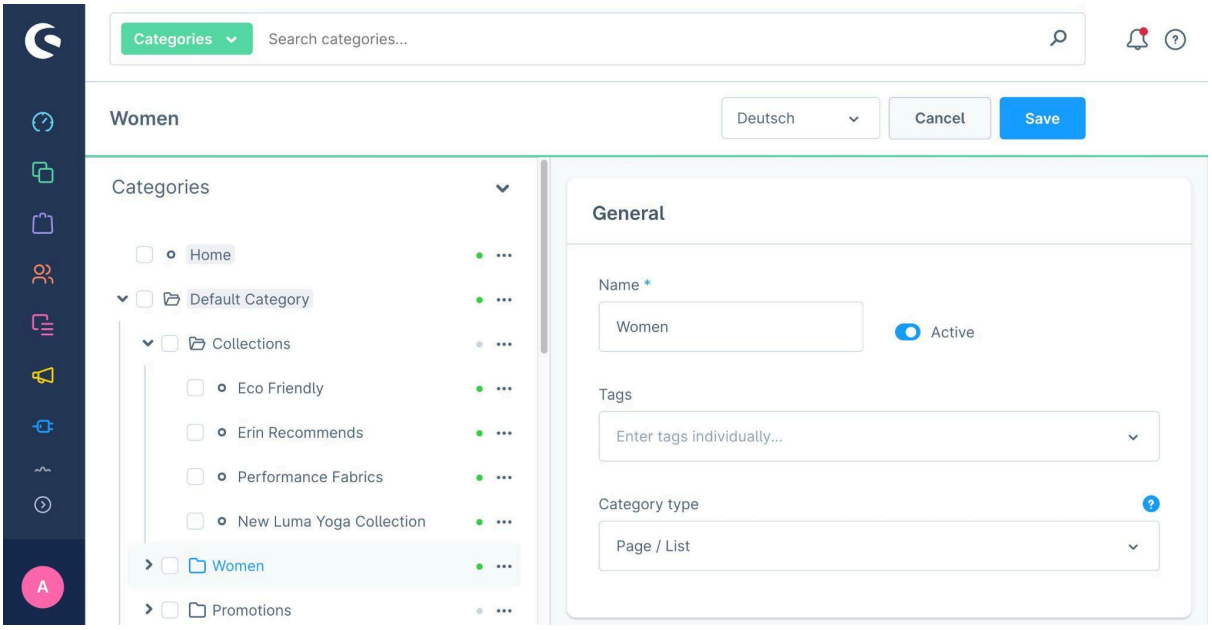
- Welche Produkttypen werden im E-Commerce System unterstützt? Sind es lediglich “simple” Produkte ohne weitere Ausprägungen, wie z.B. ein Buch oder eine DVD? Gibt es “konfigurierbare” Produkte mit mehreren Varianten, wie z.B. einen Schuh in mehrere Größen? Wenn ja, kann man mehrere Varianten anhand eines Basis-Produkts generieren? Können in den jeweiligen Varianten einzelne Attribute vererbt oder überschrieben werden? Gibt es “Bündelprodukte”, die physisch aus mehreren simplen Produkten bestehen? Wenn ja, wie kann der Preis dieser Bundles gesteuert werden? Gibt es “digitale Produkte”, wie z.B. eine Versicherung oder eine Software, die einen gesonderten Check-out bedürfen?



Verwaltung von Varianten eines konfigurierbaren Produkts in Shopware 6

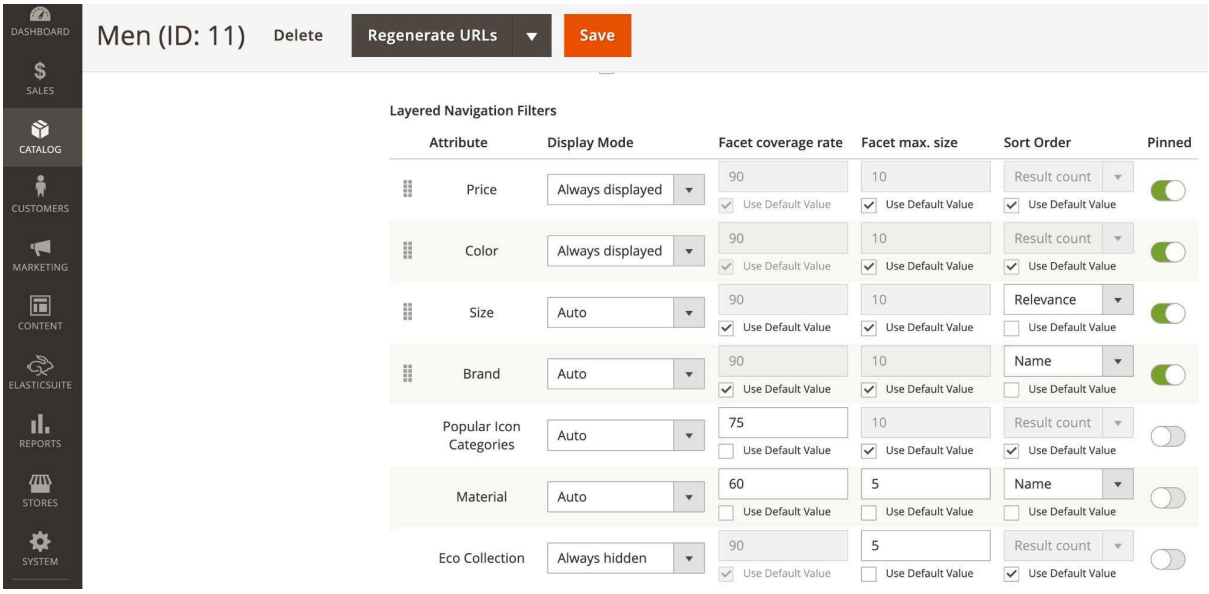
Ein weiterer wichtiger Bestandteil des Katalogs sind die Kategorien. Bei der Beschreibung der Kategorienverwaltung sollte man die folgenden Punkte beachten:

- Kann man eigenständig Kategorien erstellen und diese in einem Kategoriebaum mit mehreren Kategorie-Ebenen einordnen? Wie viele Ebenen an Unterkategorien werden vom System unterstützt?



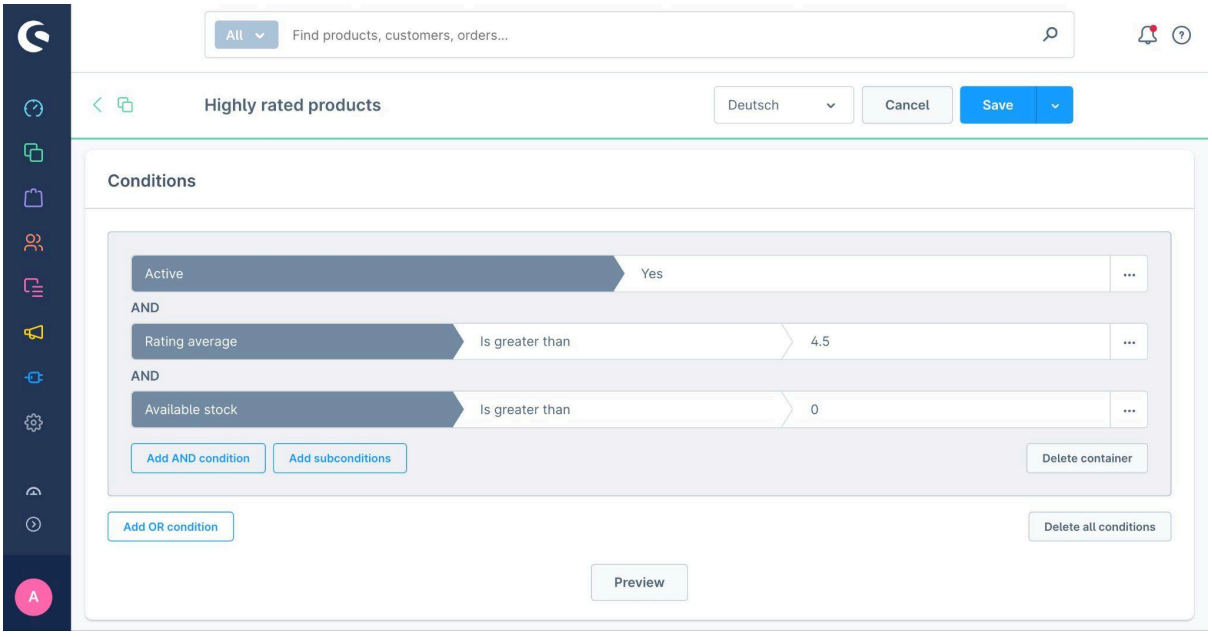
Verwaltung von Kategorien in Shopware 6

- Gibt es eine “Root Category” (bzw. Kategorie Ebene 0), die die Pflege von mehreren Katalogen bzw. Kategoriebäumen erlaubt? Kann man diese Root Category einem bestimmten Mandanten zuordnen?
- Welche Einstellungen kann der Admin in der jeweiligen Kategorie tätigen? Kann er diese aktivieren / deaktivieren und ggf. in der Navigation im Storefront verstecken? Kann man SEO relevanten Einstellungen, wie z.B. Meta Tags oder URL Keys, setzen? Können pro Kategorie die Sortierung und die Filter definiert werden? Kann man die Sortierung der Produkte manuell anpassen oder regelbasiert manipulieren (z.B. “neue Produkte” weiter oben anzeigen)?



Filtereinstellungen in Magento 2

- Wie funktioniert Produktzuweisung in den jeweiligen Kategorien? Muss jedes Produkt manuell zugewiesen werden oder gibt es eine regelbasierte Zuweisung, sodass Produkte mit einem bestimmten Attributwert der Kategorie automatisch zugeordnet werden? Passiert diese regelbasierte Zuweisung dynamisch, d.h. wird die Kategorie stetig mit neuen oder geänderten Produkten aktualisiert?



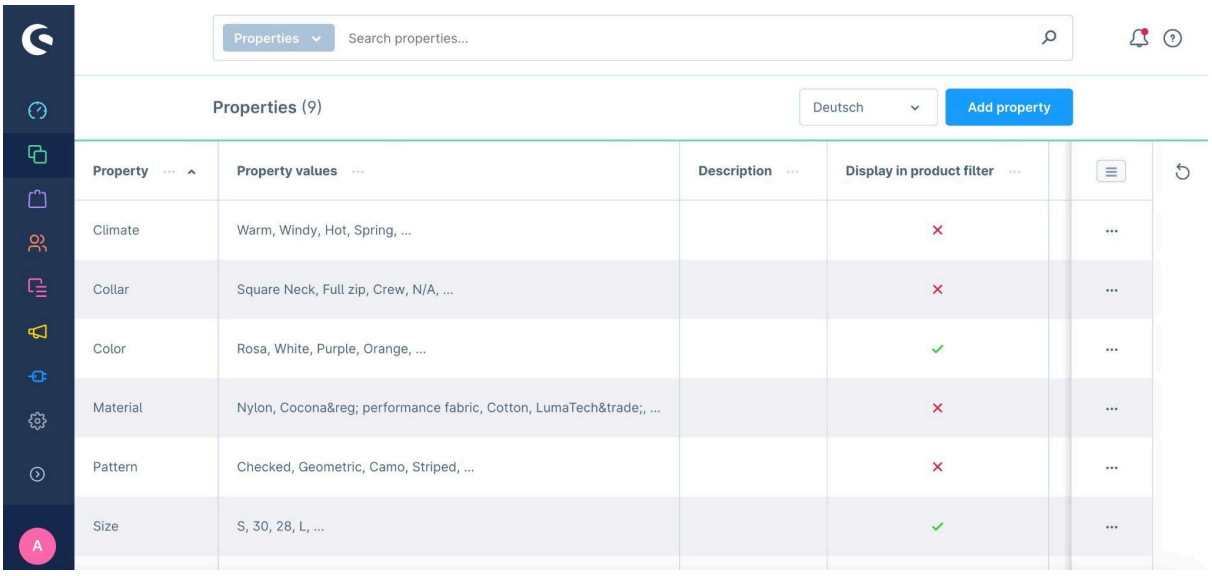
Erstellung von dynamischen Produktgruppen für Kategorien in Shopware 6

- Kann man neben der Produktzuweisung ebenso Content, wie z.B. ein Kategoriebild oder eine Beschreibung pflegen? Wird hier der gleiche Pagebuilder genutzt, der für gewöhnliche Content Seiten oder Landing-Pages genutzt wird?

- Erlaubt die Kategorieverwaltung auch eine Pflege von Menüs für Content Zwecke? Werden beispielsweise der Footer Bereich und die Footer Links über die Kategorieverwaltung gepflegt?

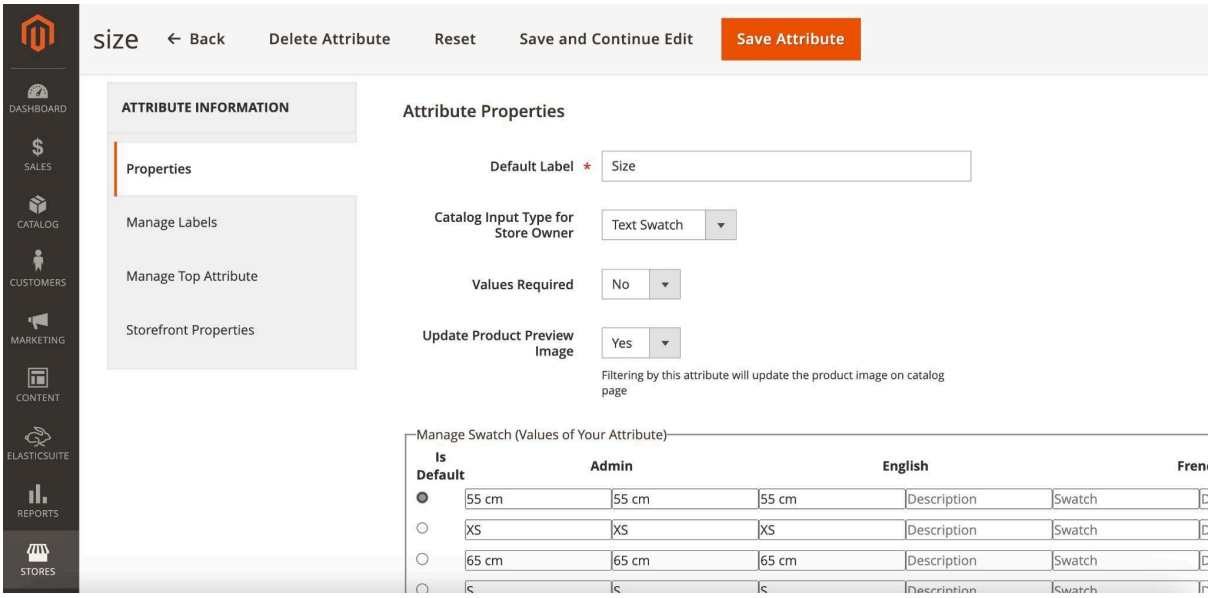
Der letzte Bestandteil der Katalogpflege sind die Attribute. Dabei sollten die folgenden Aspekte beschrieben sein:

- Welche Attributtypen werden im E-Commerce System unterstützt? Einige Beispiele können sein:
- Input Attribute, wie etwa numerische Werte ("Gewicht in KG"), Preise, Datum bzw. Zeit, simpler textlicher Input ("Produktname") oder Rich Content mit HTML Mark-up
- Visuelle Attribute, wie etwa Bild (2. Produktbild, Thumbnails auf Produktkacheln) oder Farbe
- Select-Attribute, wie etwa Checkbox (Ja / Nein), Einfachauswahl ("Schuhgröße")[14] oder Mehrfachauswahl ("Material")[15]



Verwaltung von Attributen in Shopware 6

- Welche Einstellungen kann der Admin im jeweiligen Attribut tätigen? Kann man die Sichtbarkeit auf der Kategorieseite (als Filter) und auf der Produktdetailseite steuern? Kann man eine Validierung setzen oder das Attribut als "erforderlich" bei der Produkterstellung verlangen?

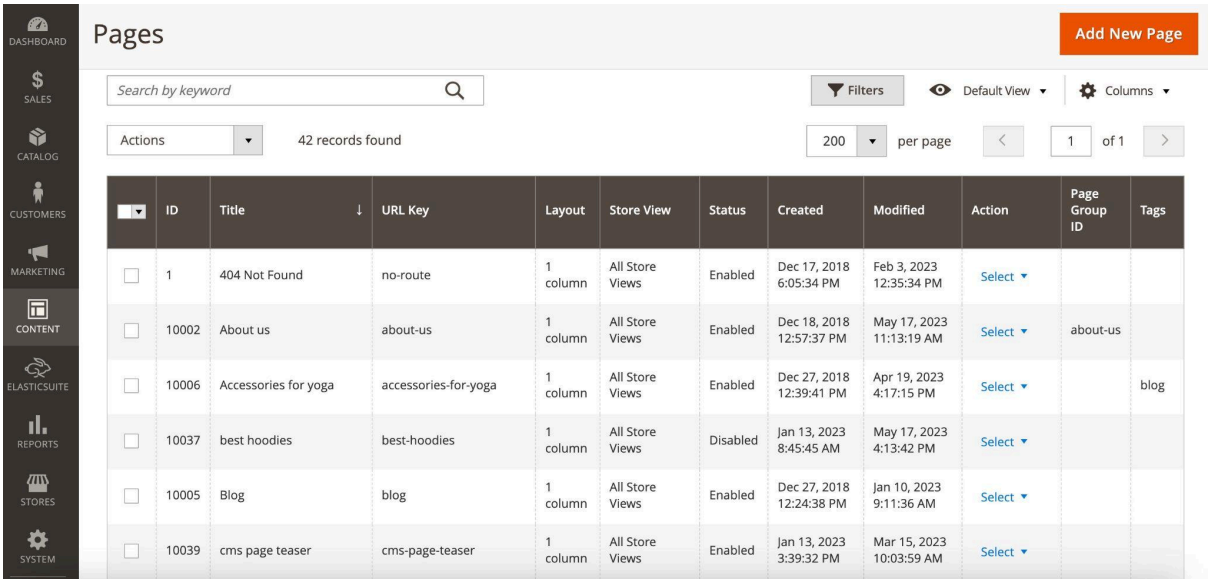


Einstellungen von Attributen in Magento 2

- Kann der Admin eigenständig Attributwerte einpflegen, z.B. bei dem Attributtyp Einfach- oder Mehrfachauswahl (wie “Größen”)? Kann man Attributwerte importieren und exportieren? Können diese in mehrere Sprachen übersetzt werden?
- Kann man Attribute in Attributsets oder Produktfamilien zusammenfassen? D.h. es gibt jeweils unterschiedliche Attribute für Produkte in den Attributsets “Schuhe” und “Shirts”.

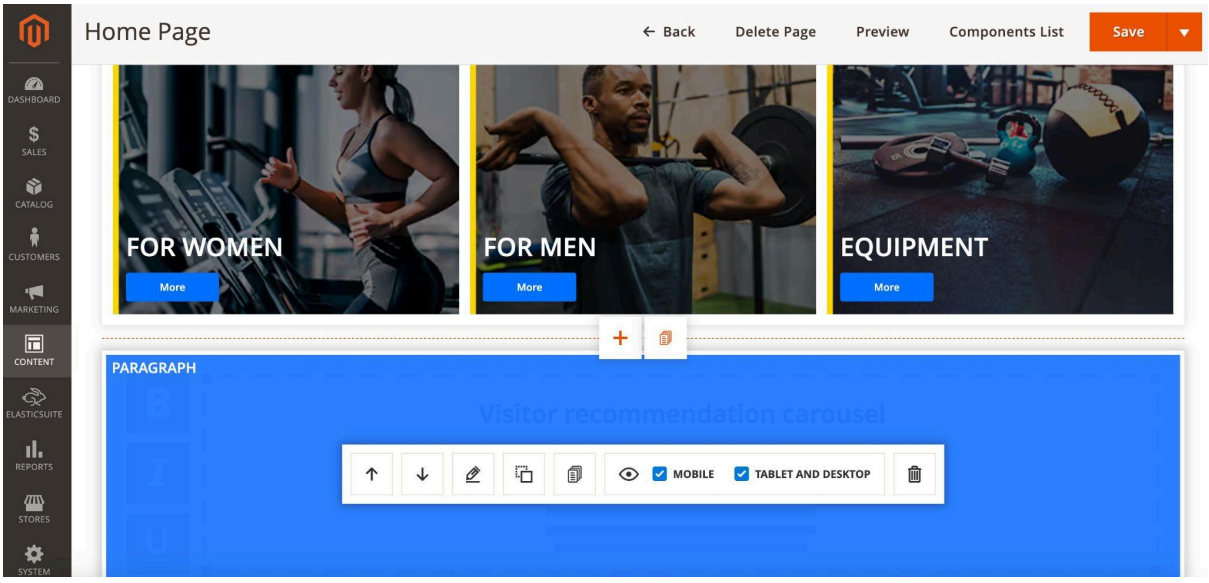
Ein weiterer Aspekt bei Storefront Inhalten sind Content-Seiten, wie z.B. die Homepage, Landing-Pages, Blogs, FAQs und rechtliche Seiten (AGB, Impressum, Datenschutzerklärung). Bei der Pflege von Content Seiten sollte man die folgenden Funktionen beschreiben:

- Gibt es eine Übersicht der existierenden Content-Seiten? Gibt es hier (ähnlich zu der Produktübersicht) eine Suche (z.B. nach Titel der Seite), Filter (z.B. nur aktive Seiten) und eine Sortierung (z.B. neue Seiten), die dem Admin Nutzer die Verwaltung erleichtern? Kann man hier ebenfalls Mass Actions nutzen, um mehrere Seiten z.B. zu aktivieren oder einem Sub-Shop zuzuschreiben?



Übersicht von Content-Seiten in Magento 2

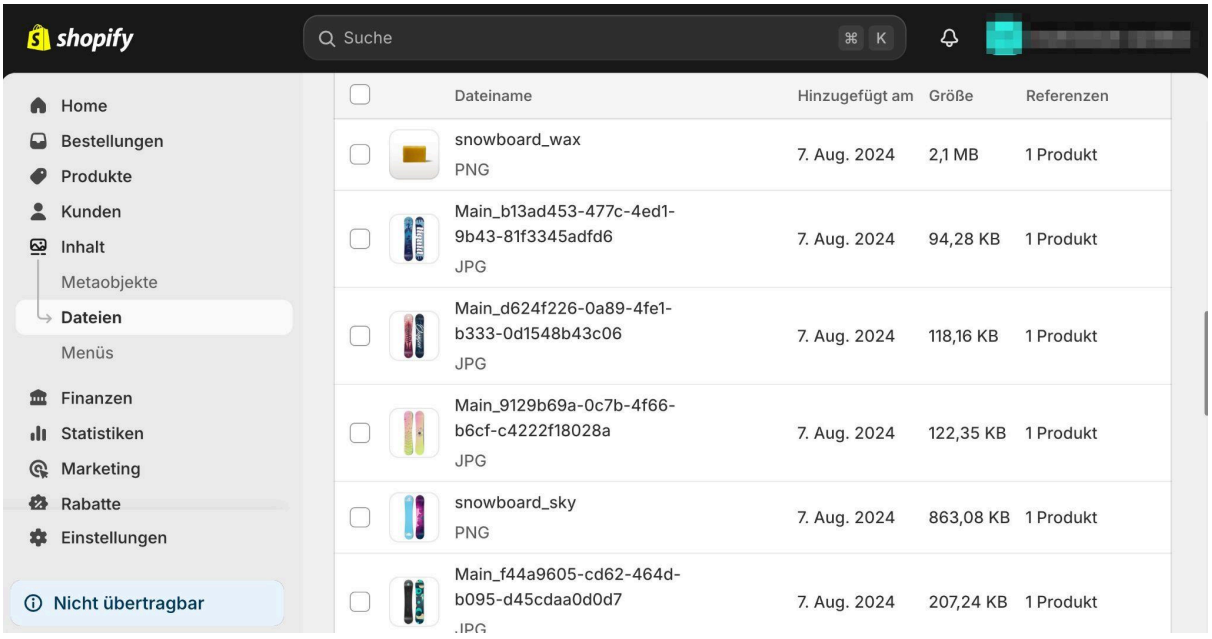
- Wie kann man Content-Seiten miteinander verknüpfen bzw. im Storefront verlinken, bspw. im Footer? Gibt es eine Möglichkeit, Menüs oder Kategorien aus Content-Seiten zu erstellen? Gibt es ggf. eine Tagging-Funktion, um Content-Seiten miteinander zu verknüpfen?
- Welche Einstellungen kann der Admin in einer Content-Seite tätigen? Kann er neben dem Seitentitel, SEO relevante Einstellungen, wie z.B. Meta Tags oder URL Key setzen? Kann er die Sichtbarkeit des Contents zeitlich schalten oder für diese für bestimmte Kunden definieren? Wie funktioniert die Übersetzung des Contents? Können Content Seiten dupliziert oder kopiert und angepasst werden?
- Welche Möglichkeiten bietet der Pagebuilder bzw. Editor des E-Commerce Plattform? Kann man mehrere Templates erstellen (rechtliche Seiten, Landing-Pages, Blog Seiten, ...), die im Anschluss mit Inhalten befüllt werden? Gibt es vordefinierte Content Komponenten, wie z.B. Hero Teaser, Image Teaser, Produktkarusselle, Text Paragraphen oder Ähnliches? Kann man bei Teaser Komponenten mit Bildern den Kontrast zur Lesbarkeit von Texten anpassen? Kann man die vordefinierten Komponenten nur zeilenweise oder auch spaltenweise einsetzen? Können die Komponenten gezielt für bestimmte Viewports wie Desktop, Tablet und Mobile angepasst oder sogar deaktiviert werden? Können einzelne Komponenten kopiert oder dupliziert werden? Kann die Sichtbarkeit einzelner Komponenten zeitlich geschaltet oder für Kunden gesteuert werden? Kann man die Komponenten per CSS eigenständig ohne Code Änderungen anpassen? Werden Entwicklerkenntnisse benötigt, um den Pagebuilder zu bedienen?

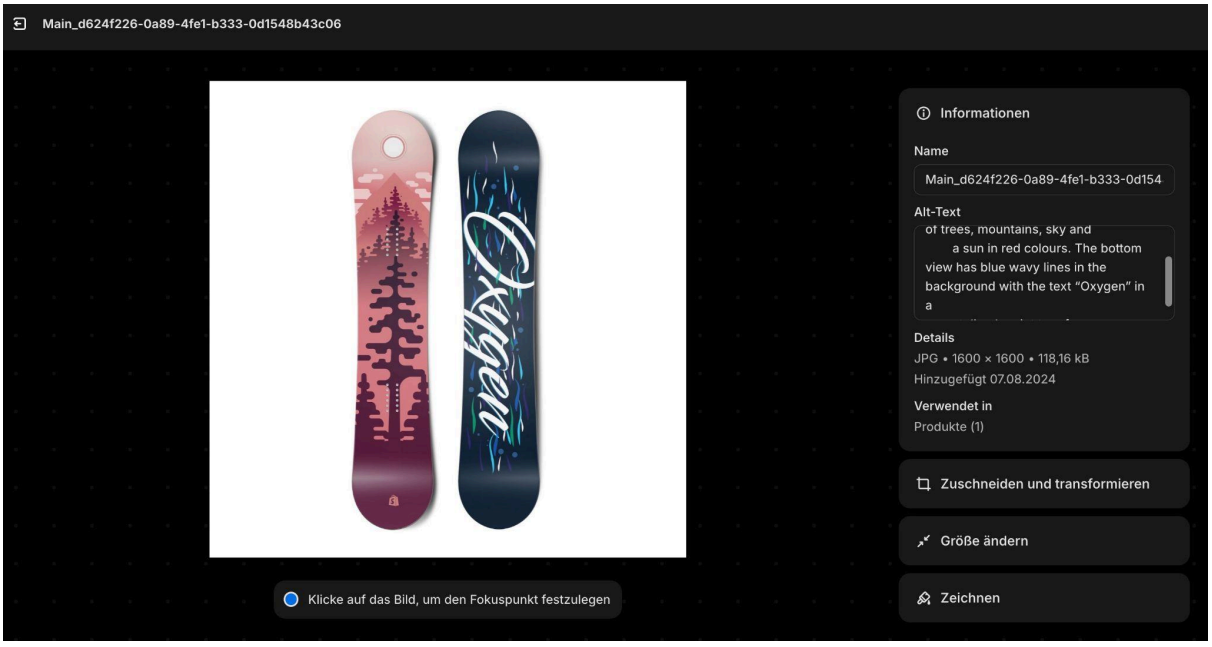


MageSuite Pagebuilder für Magento 2

Im Rahmen der Content Pflege ist die Medienbibliothek und die Verwaltung von Bildern, Videos und anderen Dateien ein wichtiger Bestandteil. Hier sollte man auf die folgenden Punkte eingehen:

- Wie ist die Medienbibliothek organisiert? Kann man mittels einer Suche nach einzelnen Dateien suchen? Gibt es eine Sortierung, um die Medien z.B. alphabetisch oder nach Erstellungsdatum zu sortieren? Kann man Filter nutzen, um bestimmte Dateiformate (JPG, PNG, PDF, MP4 usw.) zu wählen? Gibt es unterschiedliche Ansichten, z.B. mit Vorschaubild oder als Listenansicht?
- Kann man die Medien in Ordnern gruppieren, z.B. nach Saison, Kollektion oder einem anderen Use-Case? Wie viele Ebenen an Ordnern sind möglich?
- Gibt es eine Validierung vom Datei-Upload, z.B. nach Viren, bestimmten Dateiformaten oder Größe? Welche Dateiformate werden in der Medienbibliothek unterstützt?

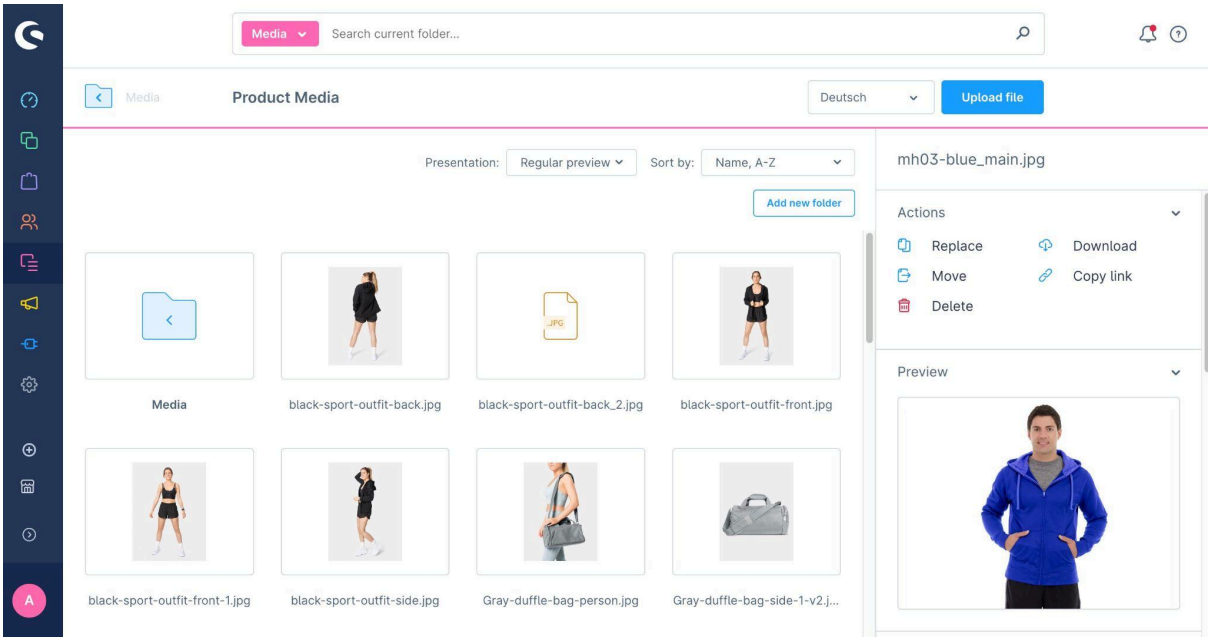




Medienbibliothek in Shopify

Bearbeitung von Medien in Shopify

- Welche Aktionen kann der Admin mit Medien tätigen? Kann er die Medien löschen und in andere Ordner verschieben? Können existierende Medien ersetzt werden, sodass die Relationen zu Produkten, Kategorien oder Content-Seiten bestehen bleiben und nur das Bild getauscht wird? Kann der Admin den Download-Link kopieren, um diesen bequem in einem HTML Tag als Download einzufügen?

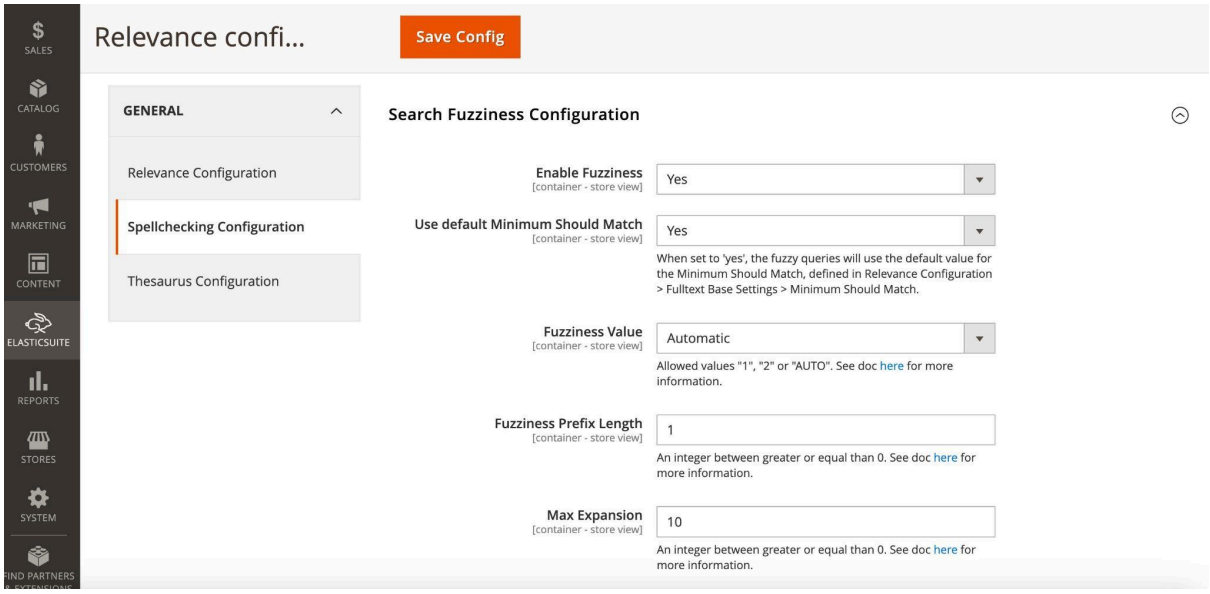


Verwaltung von Medien in Shopware 6

- Kann der Admin die Meta Tags, wie z.B. den Image-Alt Tag, ändern? Gibt es eine Tagging-Funktion, die z.B. im Rahmen eines Blogs genutzt werden kann? Kann der Admin sehen, an welcher Stelle (z.B. Produkt oder Kategorie) das Bild im Storefront genutzt wird?

Ein weiterer wichtiger Aspekt der Admin Einstellungen sind die Funktionalität und Konfiguration der Suche. Hierbei sollte man die folgenden Punkte beschreiben:

- Kann der Admin Nutzer definieren, welche Entitäten in der Suche als Ergebnis geliefert sollen? Kann der Admin Nutzer definieren, ob Suchbegriffe, Produkte, Kategorien, Content-Seiten und ggf. bestimmte Attribute (z.B. “Marken”) als Ergebnisse geliefert werden sollen?
- Kann der Admin Nutzer Einstellungen zur Fehlertoleranz hinterlegen und z.B. definieren, wie viele Zeichen als “Fehler” gelten?

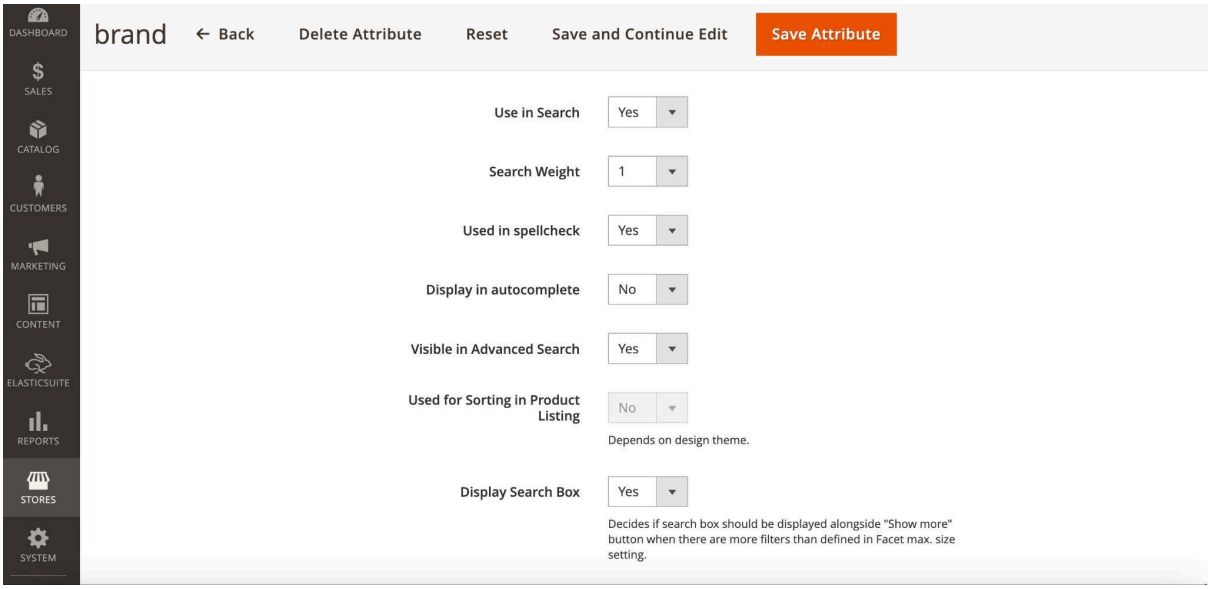


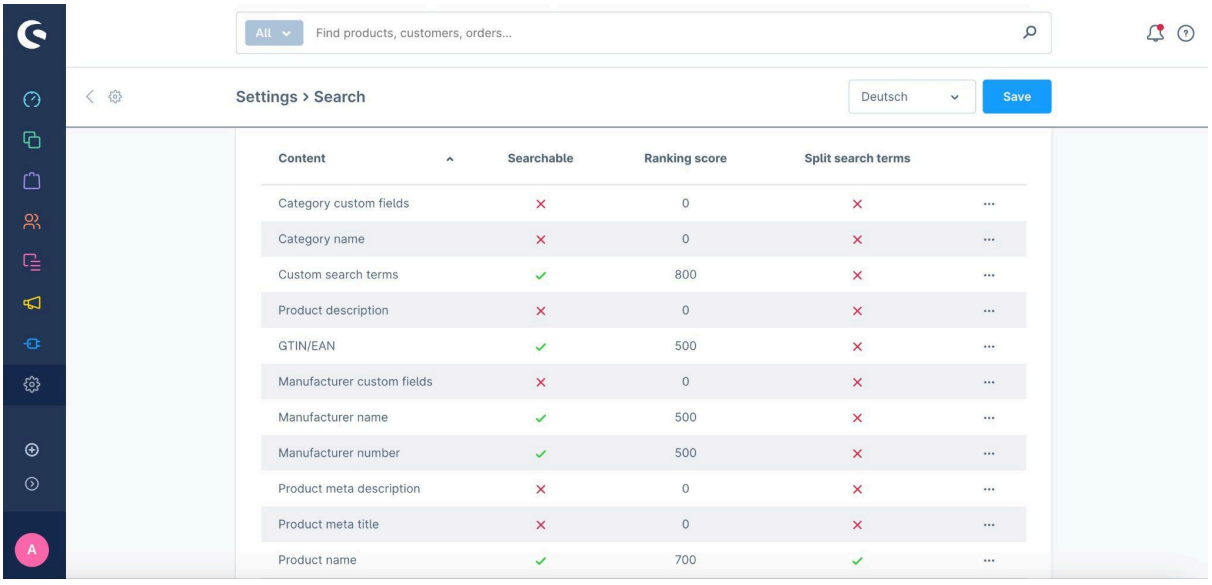
Konfiguration der Fehlertoleranz der ElasticSuite Suche in Magento 2

- Kann der Admin Nutzer Synonyme[16] und Erweiterungen[17] definieren? Kann man auf bestimmte Suchbegriffe einen Redirect auf z.B. eine Kategorie-, Content- oder Produktseite setzen?

Synonyme und Erweiterungen der ElasticSuite Suche in Magento 2

- Kann man Produktattribute hinsichtlich Ihrer Relevanz für die Suche gewichten, sodass z.B. “Produktname” eine höhere Relevanz als “Produktbeschreibung” hat? Kann man bestimmte Attribute von der Suche ausschließen? Kann man ggf. ganze Wörter aus der Suche ausschließen?

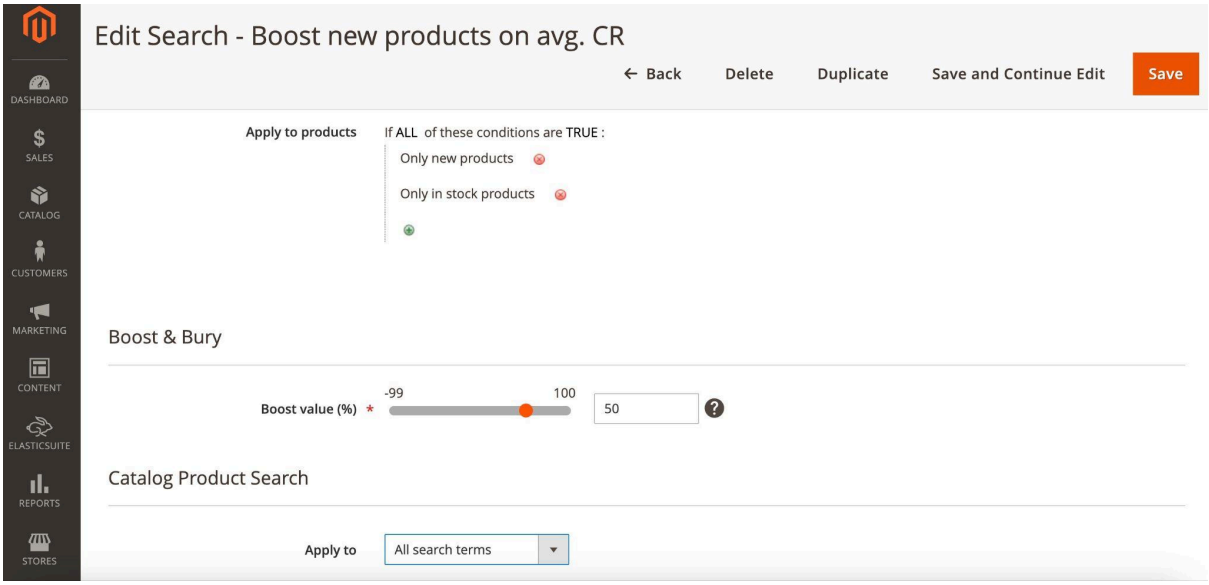




Konfiguration des Attributes "brand" für die Suche in Magento 2

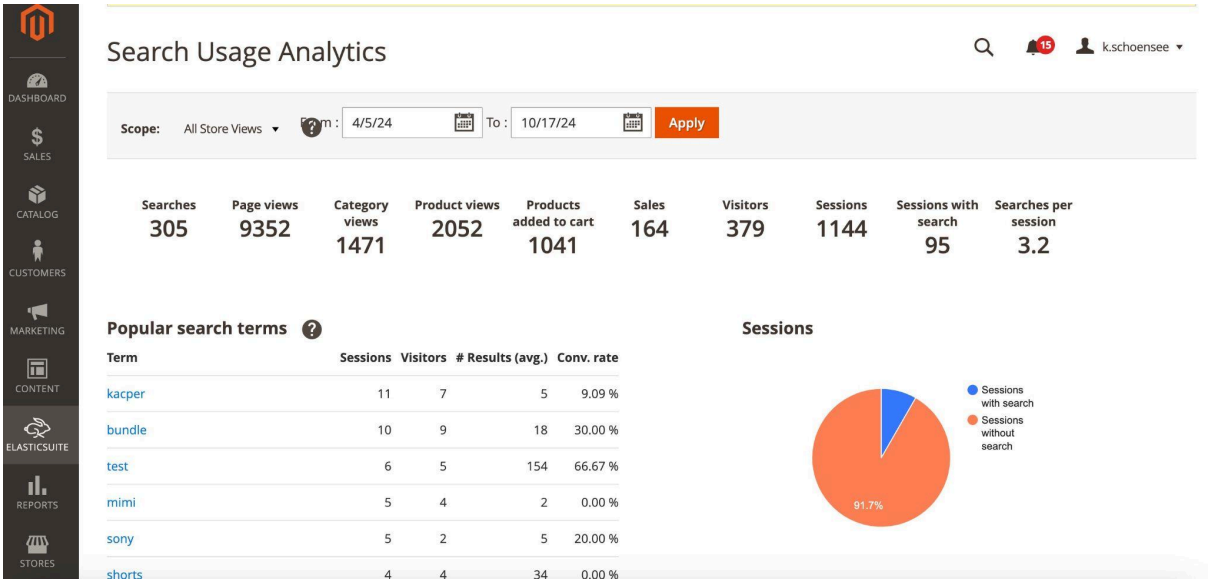
Konfiguration der Suchrelevanz von Produktattributen in Shopware 6

- Kann der Admin Nutzer Suchergebnisse regelbasiert manipulieren, sodass z.B. neue Produkt mit hohem Warenbestand weiter oben gelistet sind, während Produkte mit niedrigem Warenbestand weiter unten gelistet sind? Kann man diese Regeln für bestimmte Suchbegriffe definieren?



Konfiguration von "Optimizer" der ElasticSuite Suche in Magento 2

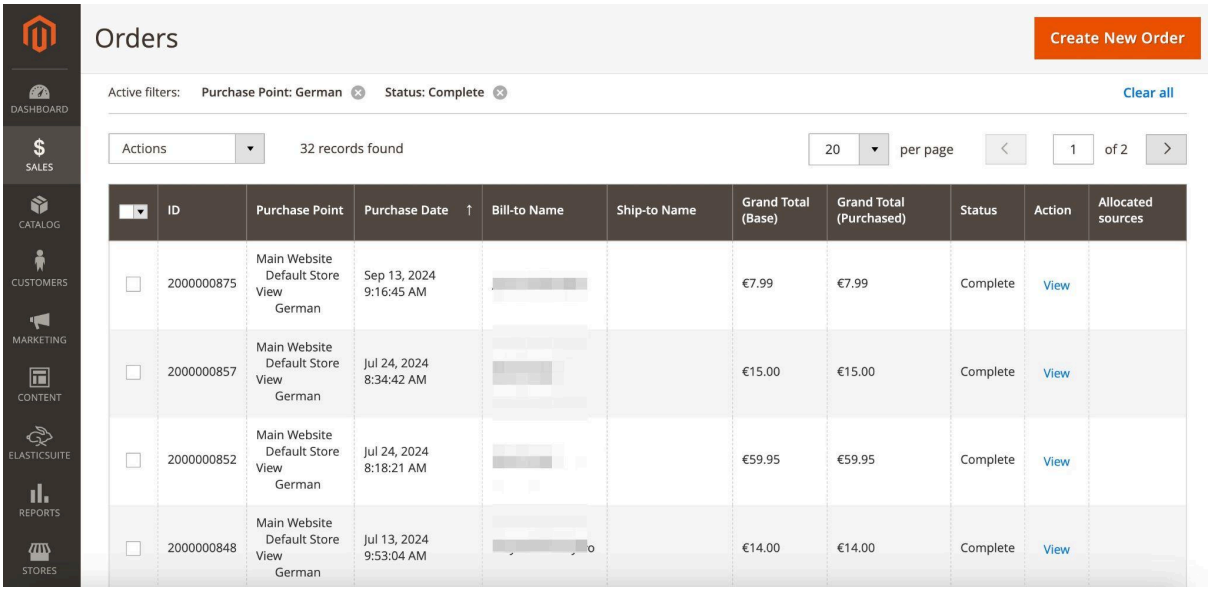
- Gibt es eine Auswertung der genutzten Suchbegriffe?



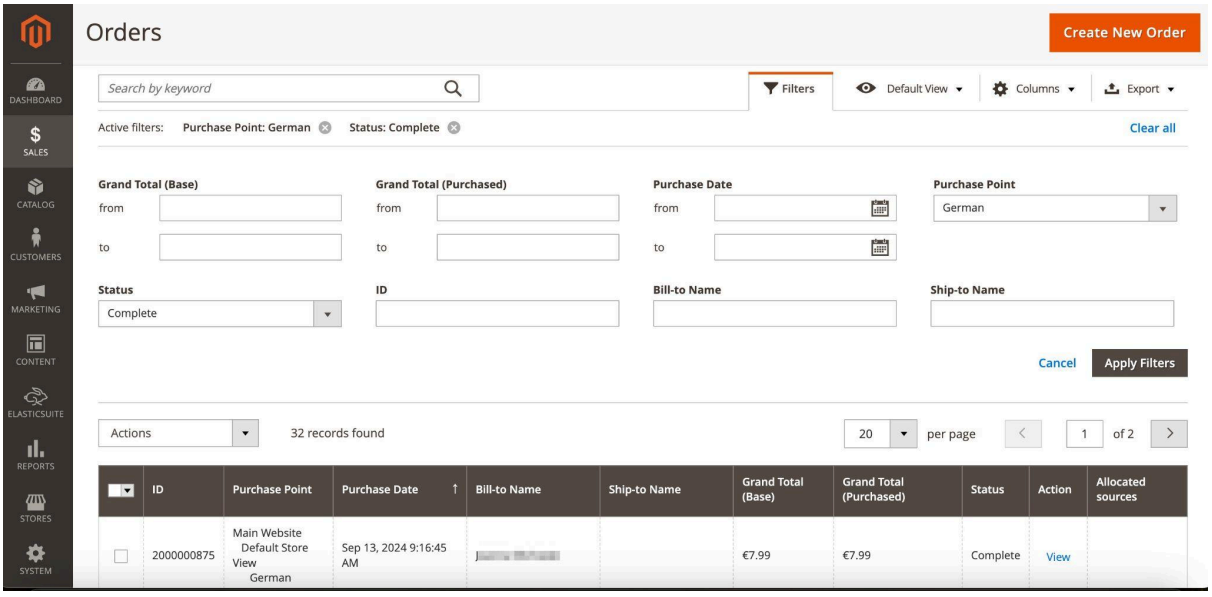
Analyse der ElasticSuite Suche in Magento 2

Das Herzstück eines E-Commerce Systems sind Bestellungen. Dabei sollten etwa die folgenden Funktionen beschrieben werden:

- Gibt es eine Übersicht der Bestellungen? Kann man hier eine Suche nutzen, z.B. nach Order ID oder Kundenname? Gibt es eine Sortierung, z.B. nach den neuesten Bestellungen oder dem Warenkorbwert? Kann man Filter nutzen, um die Bestellungen, z.B. nach einer Zahlungsmethode, einzugrenzen?

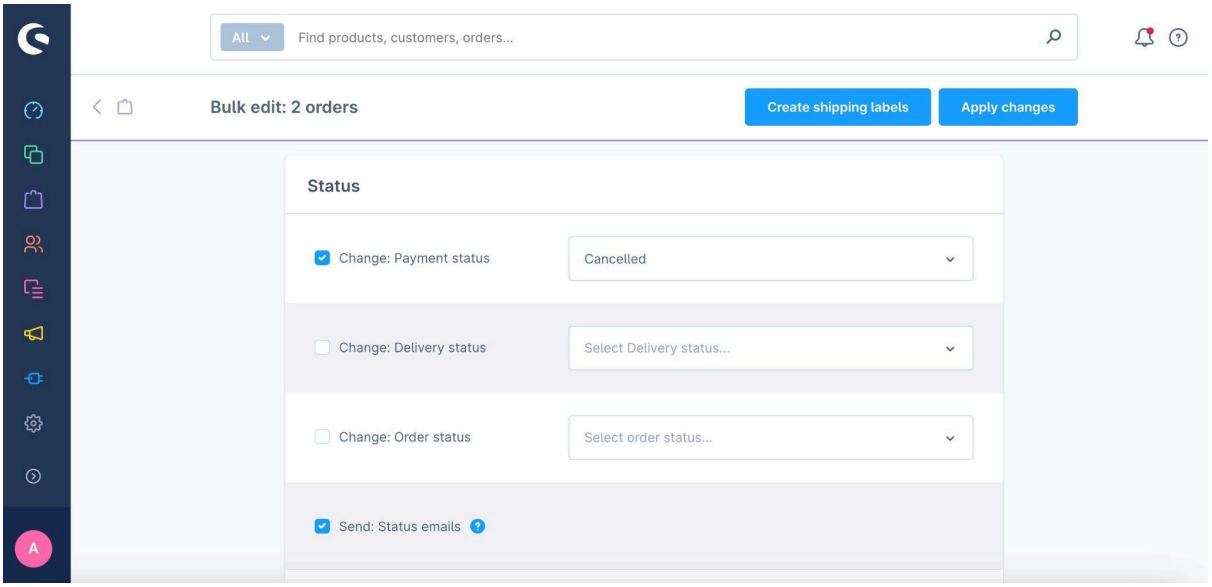


Bestellübersicht mit angewandten Filtern in Magento 2



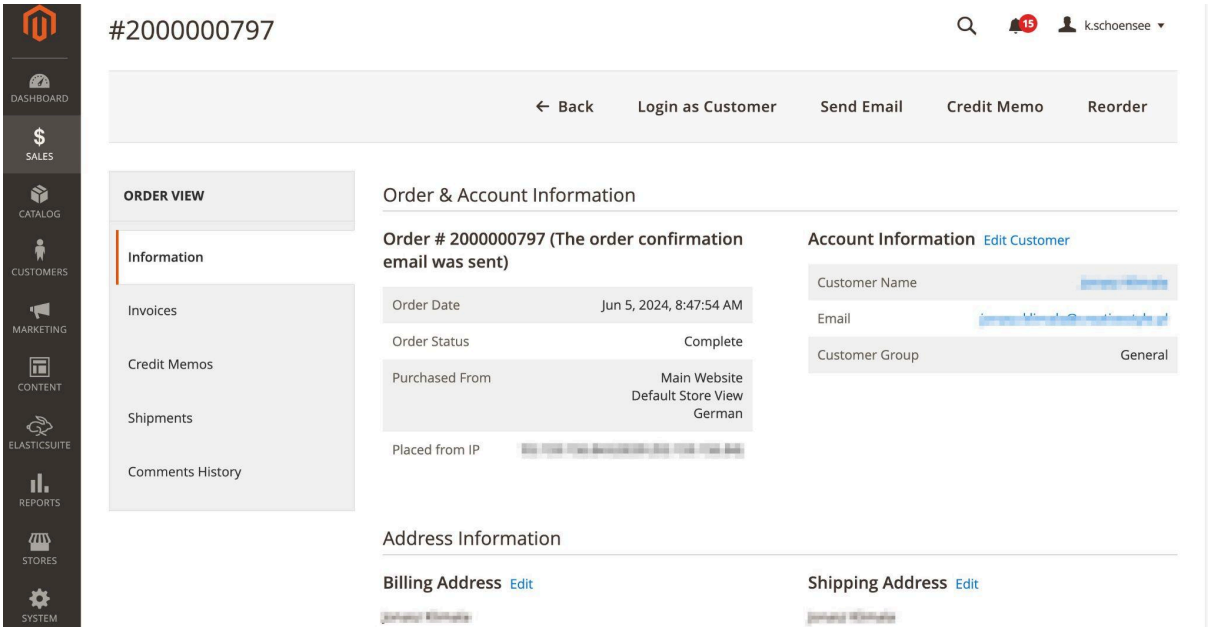
Nutzung von Filtern in der Bestellübersicht in Magento 2

- Kann man die Bestellungen exportieren, z.B. im CSV oder Excel Format? Kann der Admin Nutzer das Exportprofil editieren und z.B. Spalten entfernen oder hinzufügen?
- Kann der Admin Nutzer Bestellungen per Mass Action editieren? Kann man hier den Status der Bestellung ändern und diese z.B. auf “storniert” oder “abgeschlossen” setzen? Kann man Dokumente, wie z.B. eine Rechnung oder einen Lieferschein erstellen? Können E-Mails, wie z.B. eine Bestellbestätigung, manuell verschickt werden? Kann man Bestellungen löschen?

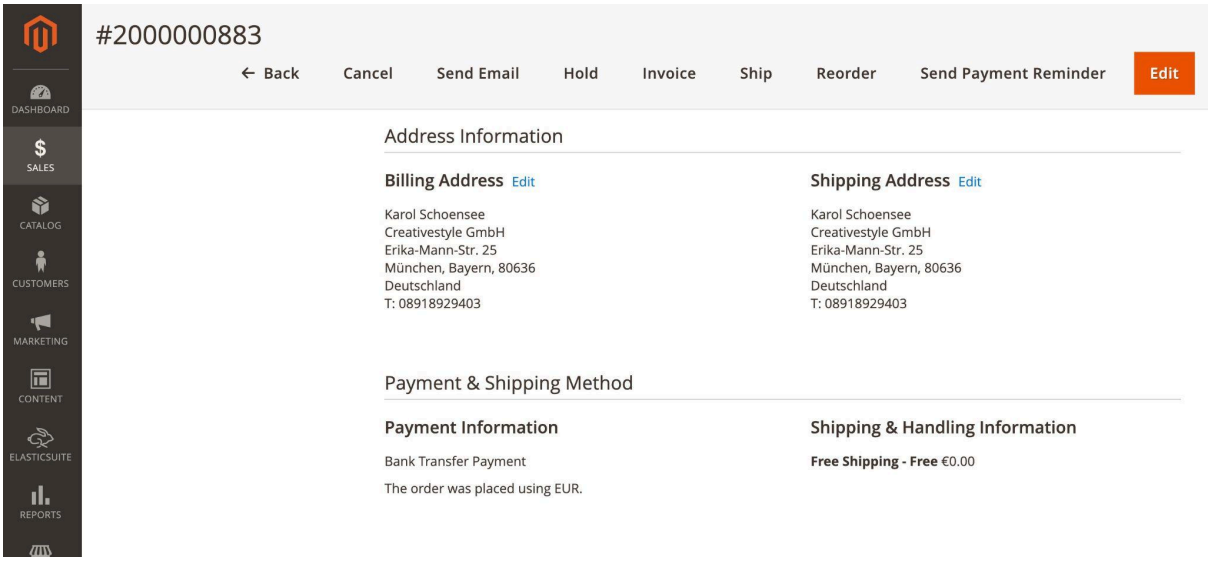


Bulk Edit von Bestellungen in Shopware 6

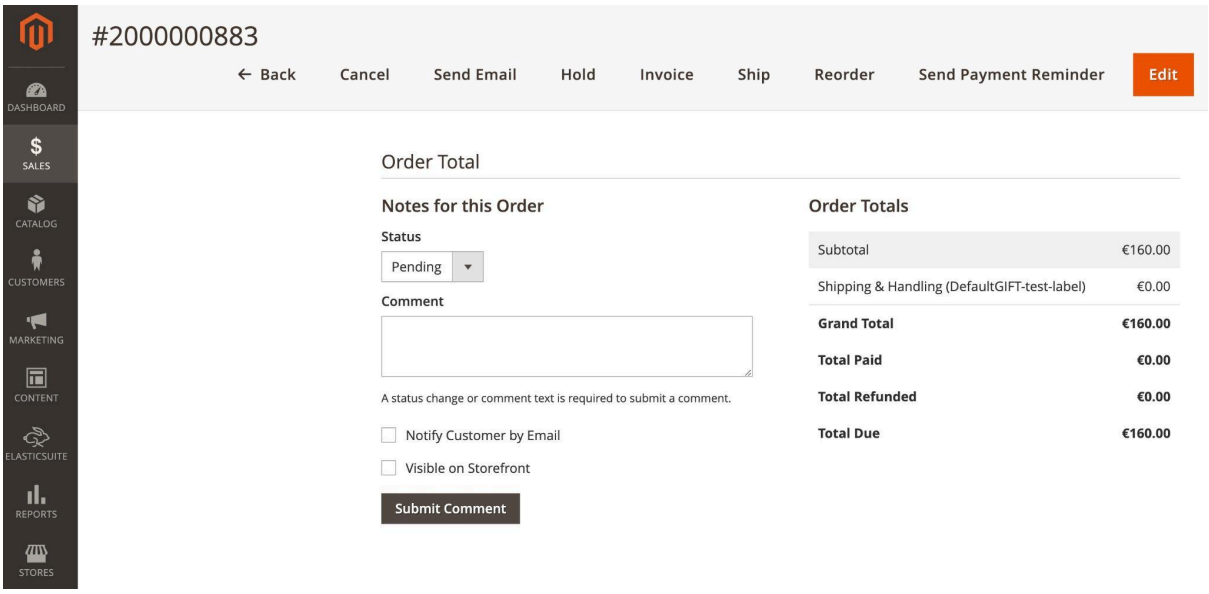
- Welche Informationen können auf der Bestelldetail-Ansicht eingesehen werden? Sind es der Bestellstatus, Zahlungs- und Versandmethode, die Kundendaten (z.B. Name und Adresse) sowie Warenkorb, Steuern und ggf. angewandte Rabatte? Kann man die hier enthaltenen Informationen, wie den Warenkorb, nachträglich ändern?



Order Details in Magento 2 (1 / 3)

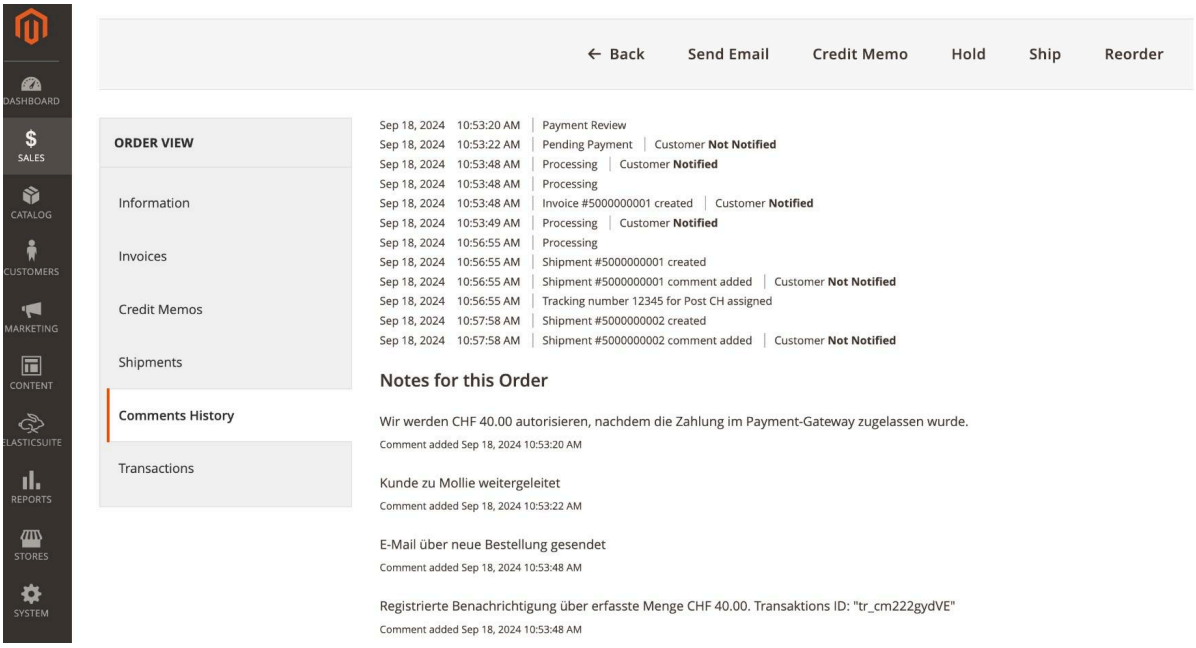


Order Details in Magento 2 (2 / 3)



Order Details in Magento 2 (3 / 3)

- Erfolgt eine Synchronisation mit dem Payment Service Provider und kann man den Status sehen, z.B. Zahlungseingang erfolgt? Kann man die Transaktion über den Admin Bereich synchronisiert mit dem Payment Service Provider stornieren oder eine Gutschrift erstellen?

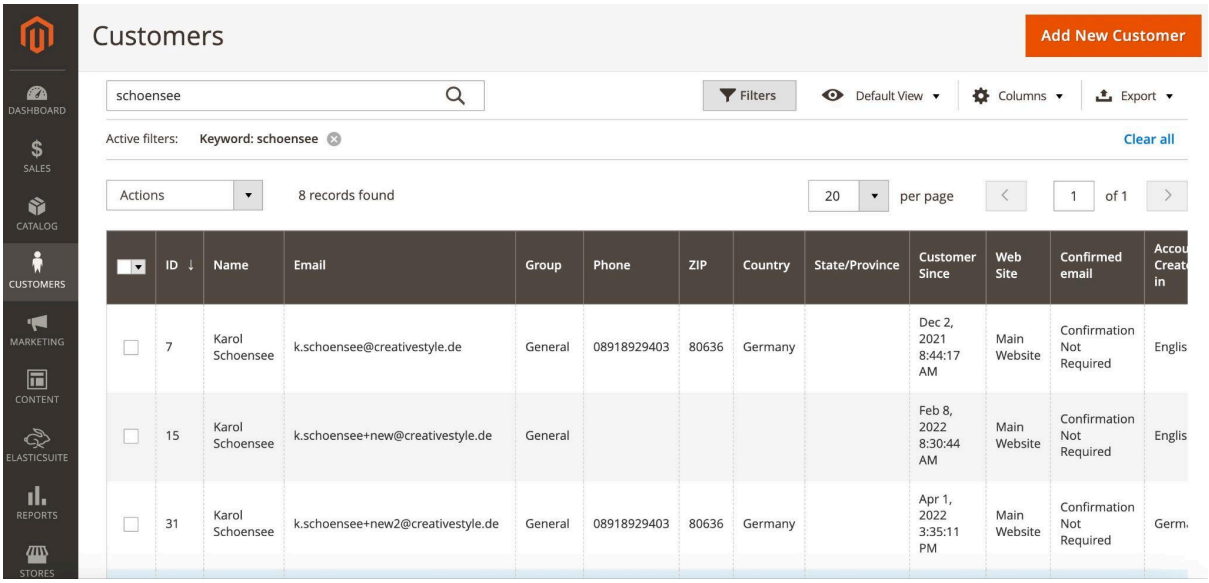


Order Historie inkl. Payment Synchronisation in Magento 2

- Kann man für eine Bestellung eine Rechnung erstellen? Wird diese als PDF per E-Mail an den Kunden versandt? Können pro Bestellung mehrere Rechnungen erstellt werden?
- Kann man für eine Bestellung eine Versandmeldung inkl. Tracking Link, z.B. für DHL, erstellen? Wird diese als PDF per E-Mail versandt? Können pro Bestellung mehrere Teillieferungen erstellt werden?

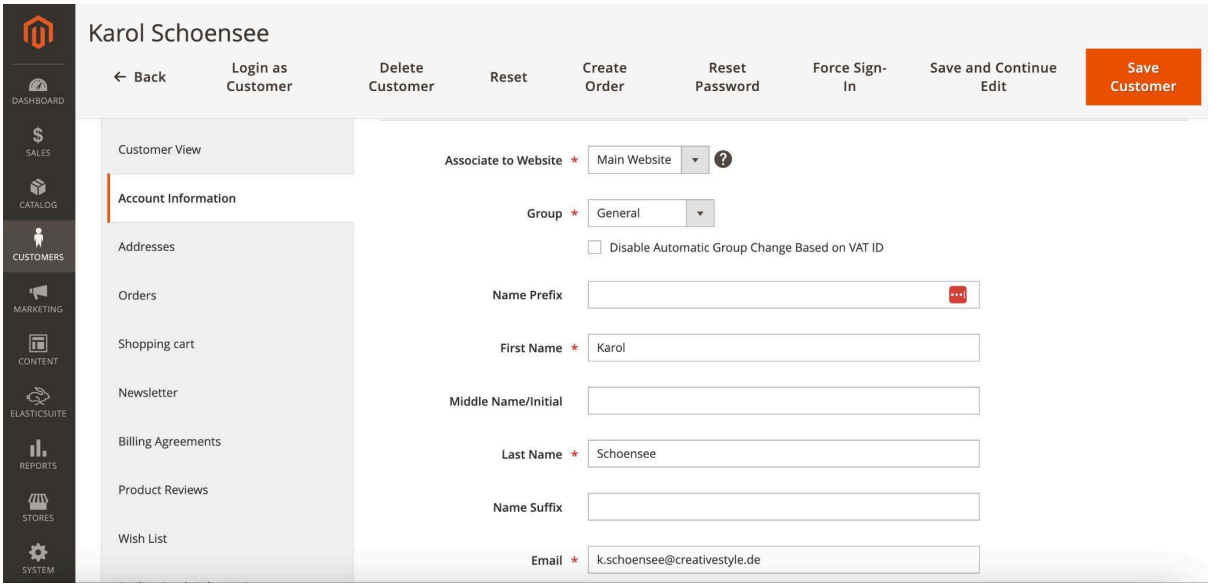
Neben der Verwaltung von Bestellungen ist ebenfalls die Verwaltung von Kunden und deren Konten ein wichtiger Aspekt eines E-Commerce Systems. Hier sollten etwa die folgenden Funktionen beschrieben werden:

- Gibt es eine Übersicht der Kunden? Kann man hier eine Suche nutzen, z.B. nach Name oder Adresse? Gibt es eine Sortierung, z.B. nach den neuesten Kundenkonten oder nach alphabetischer Reihenfolge? Kann man Filter nutzen, um z.B. nur nach “aktiven” Kunden einzugrenzen?



Übersicht von Kundenkonten in Magento 2

- Kann man die Kunden exportieren, z.B. im CSV oder Excel Format? Kann der Admin Nutzer das Exportprofil editieren und z.B. Spalten entfernen oder hinzufügen?
- Kann man Kunden in eine Kundengruppe oder ein Segment ordnen? Wofür können diese Kundengruppen genutzt werden? Sind es Steuerregeln, Sichtbarkeit von Content, Kategorien und Produkten oder Preis- und Rabattregeln?
- Kann der Admin Nutzer Kunden per Mass Action editieren? Kann man hier den Status z.B. auf “inaktiv” setzen oder die Kundengruppe ändern? Können E-Mails, wie z.B. eine “Passwort vergessen” E-Mail, verschickt werden? Kann man Kunden löschen?
- Kann man in einem Kundenkonto die Daten, wie z.B. die Adresse, ändern? Können die getätigten Bestellungen, Wunschlisten, Warenkörbe und die verfassten Produktbewertungen eingesehen werden? Kann der Admin Nutzer den Kunden in einen Newsletter einschreiben? Kann man sich über das Admin-Konto als Nutzer einloggen, das damit verbundene Storefront aus Kundensicht einsehen und z.B. für ihn Bestellungen tätigen oder den Warenkorb befüllen? Wenn ja, gibt es eine entsprechende DSGVO-konforme Freigabe des Kunden?

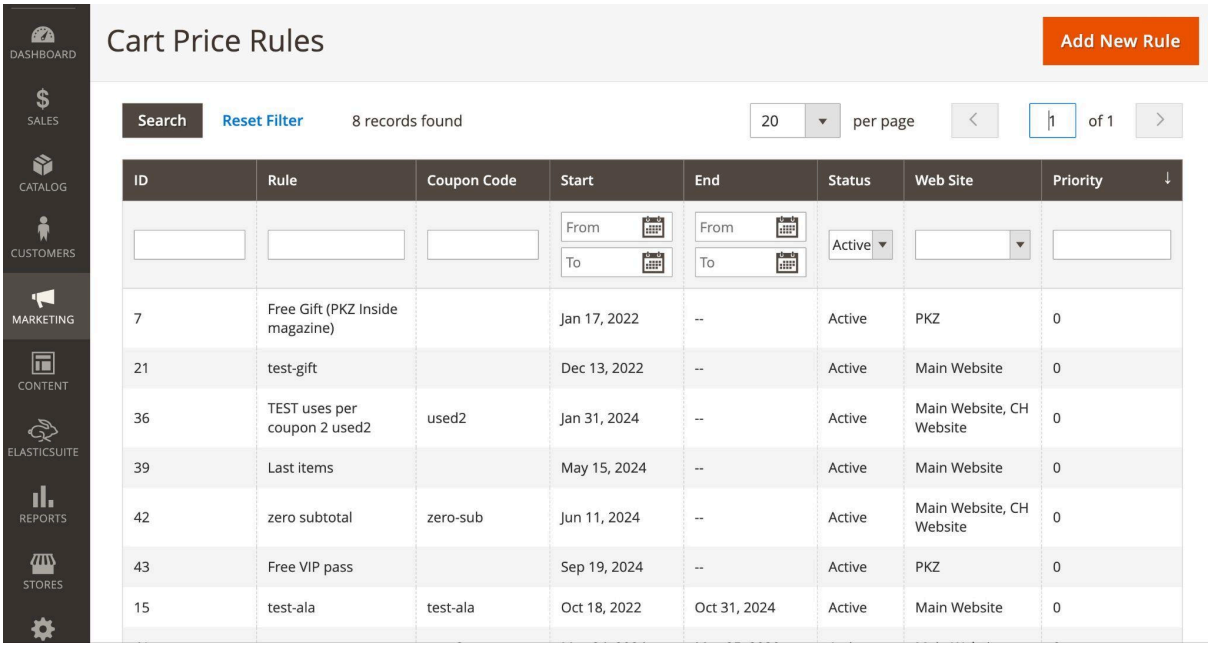


Verwaltung vom Kundenkonto in Magento 2

- Werden Kunden nur einem bestimmten Mandanten bzw. Sub-Shop zugeordnet und nutzbar oder werden die Konten mit allen Mandanten geteilt? Falls ein Kundenkonto mit mehreren Mandaten geteilt wird, sind alle Bestellungen der anderen Mandanten ebenfalls sichtbar?

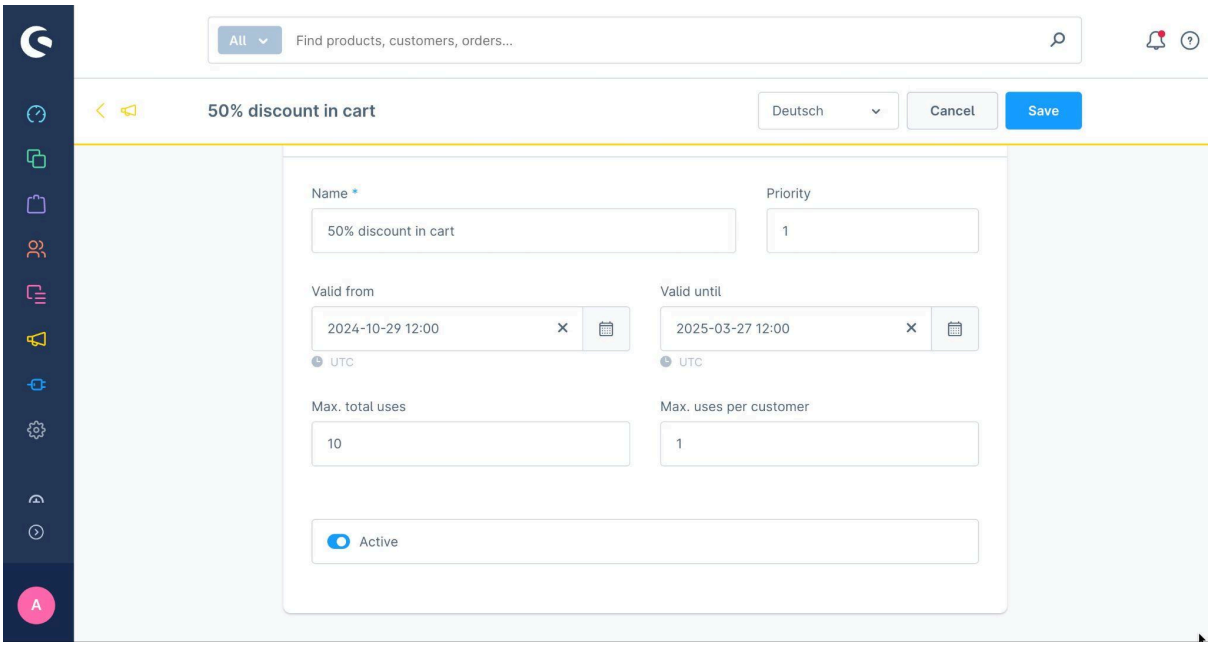
Viele Online-Shops bieten die Möglichkeit, Rabatte und Preisregeln zu nutzen. Bei der Verwaltung und Konfiguration dieser sollte man auf die folgenden Punkte eingehen:

- Gibt es eine Übersicht der Rabatte und Preisregeln? Kann man hier eine Suche nutzen, z.B. nach “Rabattcode”? Gibt es eine Sortierung, z.B. nach den neuesten Regeln? Kann man Filter nutzen, um z.B. nur nach “aktiven” Regeln einzugrenzen?



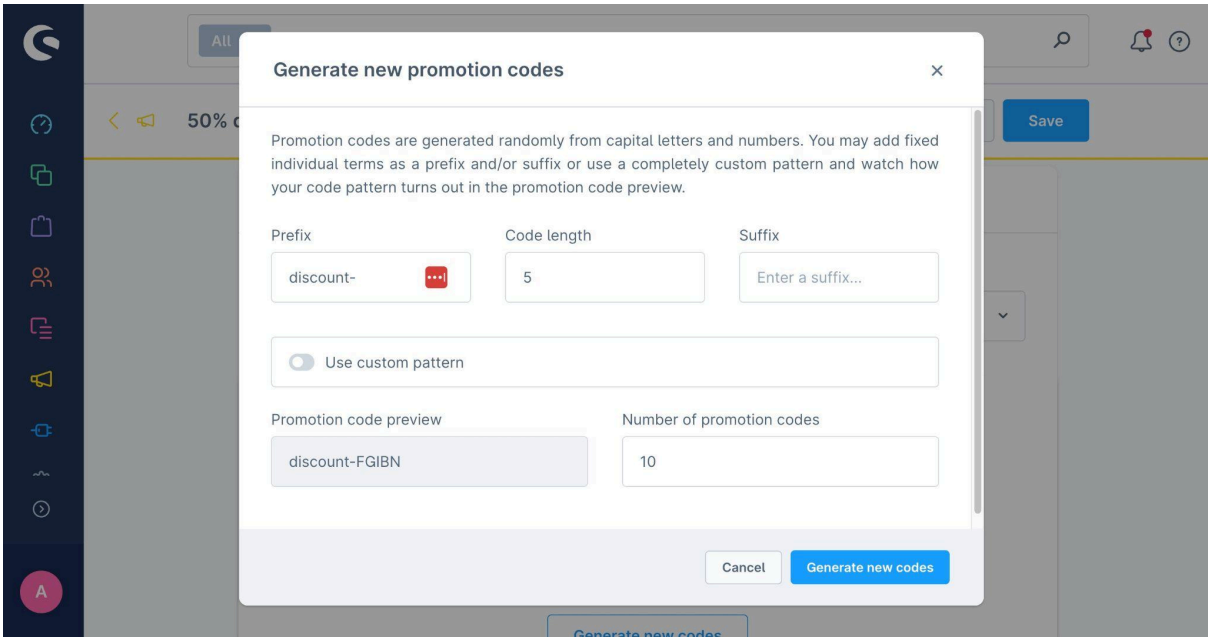
Verwaltung von Warenkorbpreisgeln in Magento 2

- Welche allgemeinen Einstellungen kann der Admin Nutzer bei der Erstellung einer Regel setzen? Kann der Admin definieren, in welchem Zeitraum die Regel aktiv ist? Kann die Regel auf eine max. Anzahl an Nutzungen (z.B. pro Kunde) beschränkt werden? Falls mehrere Regeln aktiv sind, kann der Admin die Priorität der einzelnen Regel steuern? Kann die Regel von der Kombination mit anderen Regeln ausgeschlossen werden?



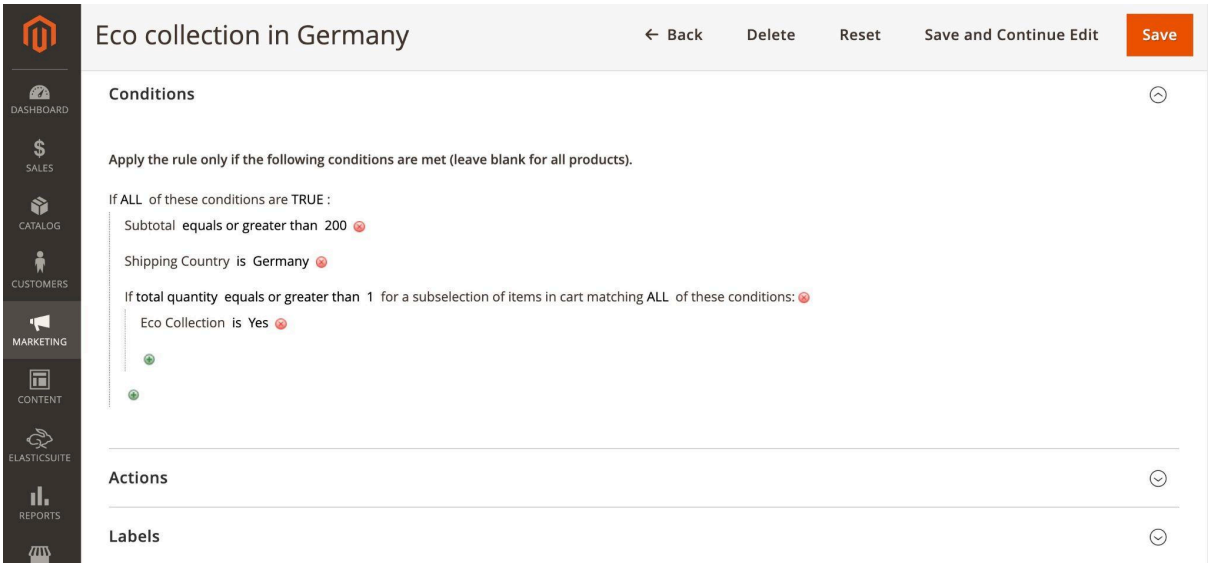
Erstellung von Warenkorbpreisgeln (Promotions) in Shopware 6

- Kann der Admin Nutzer definieren, ob eine Rabattcodeeingabe vom Nutzer im Storefront notwendig ist? Wenn ja, kann der Admin Nutzer einen spezifischen Code konfigurieren oder mehrere Codes vom System aus, etwa nach einem bestimmten Schema, generieren?



Generierung von Rabattcodes in Shopware 6

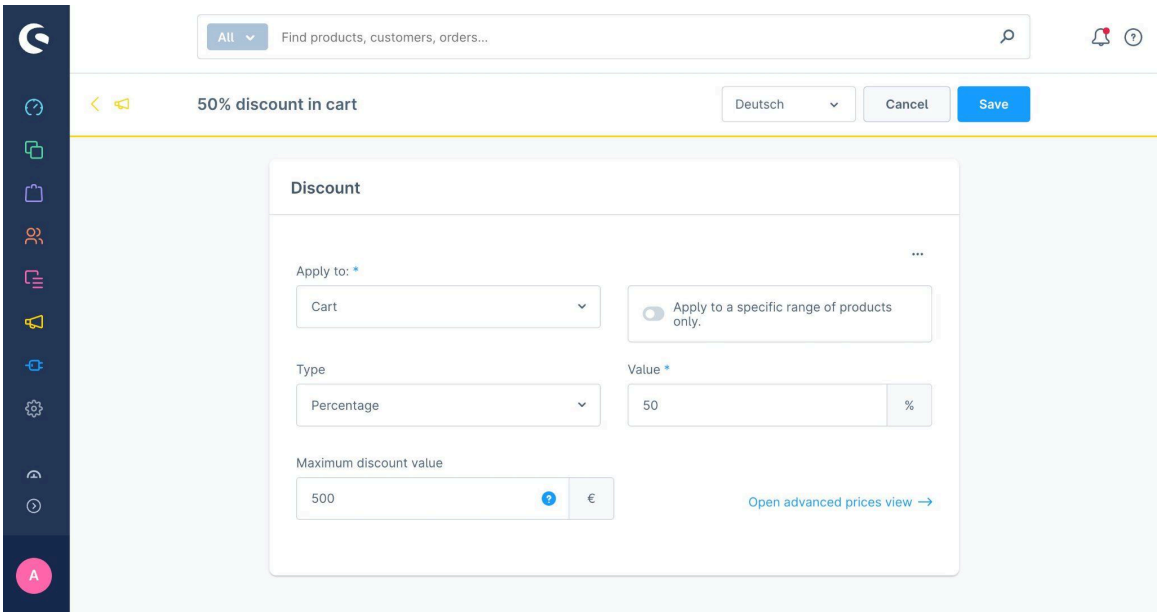
- Kann der Admin Nutzer die Bedingungen für das Auslösen der Regeln definieren? Kann die Regel für bestimmte Mandanten oder Kundengruppen ausgeschlossen werden? Kann der Admin Nutzer bestimmte Warenkorbeigenschaften, z.B. Warenkorbwert mindestens 200€, sowie Kundeneigenschaften, z.B. Versandland ist Deutschland, definieren? Können Produktattribute oder Kategorien für die Bedingungen genutzt werden?



Definition von Bedingungen für Warenkorbpreisregeln in Magento 2

- Welche Art an Preis- und Rabattregeln kann der Admin Nutzer definieren? Kann ein absoluter, z.B. 10€, oder prozentualen Rabatt, z.B. 10%, eingestellt werden? Gibt es eine Möglichkeit, Produkte kostenfrei als Geschenke, etwa ein Magazin

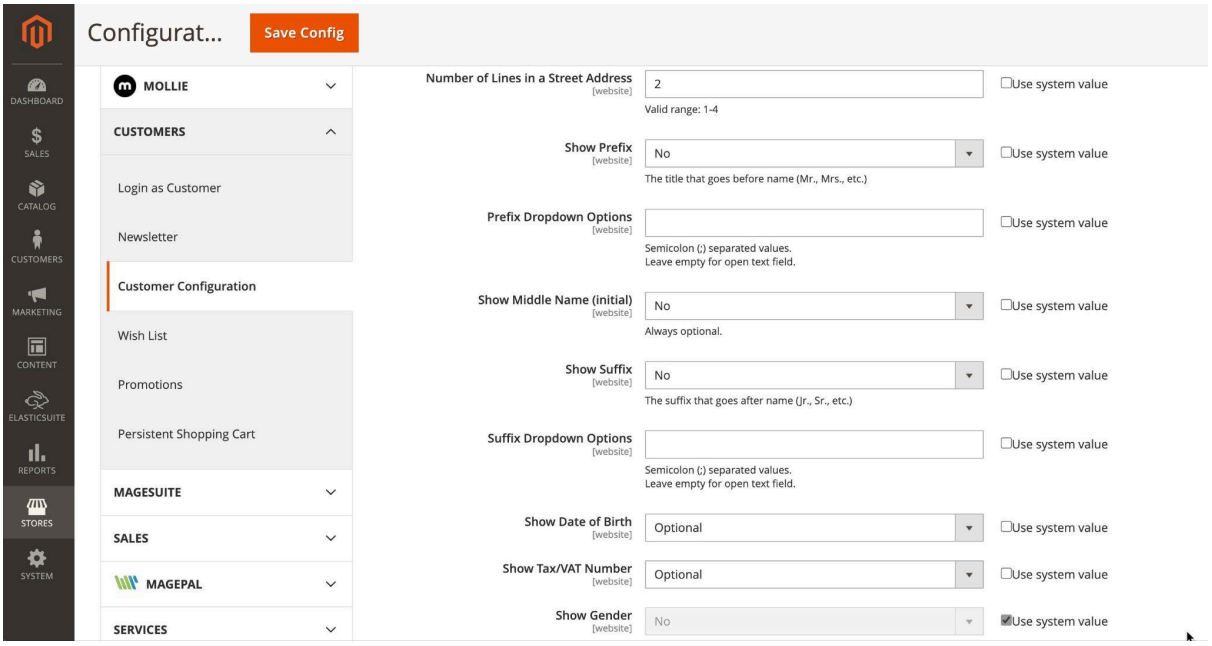
oder eine Produktprobe, dem Warenkorb hinzuzufügen? Gibt es eine “kaufe X erhalten Y” Rabattregel?



Definition der Rabattart und -höhe in Shopware 6

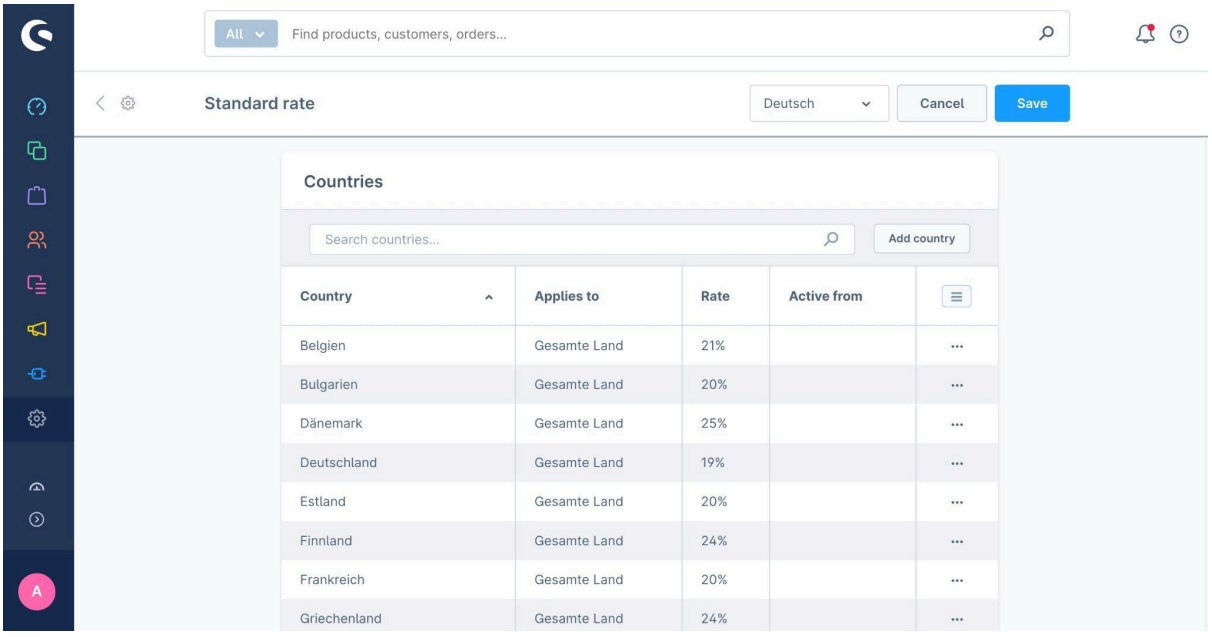
Das aktuelle E-Commerce System bietet ggf. auch Einstellungen im Warenkorb und Check-out. Dabei sollten die folgenden Funktionen beschrieben werden:

- Können Regeln für den Warenkorb definiert werden, unter denen der Kunde keinen Check-out abschließen darf? Können etwa eine maximale Anzahl an enthaltenen Produkten oder ein Mindestbestellwert definiert werden?
- Kann der Admin Nutzer die Felder im Check-out bzw. für die Kundenregistrierung und die Adressen definieren? Kann er lediglich vordefinierte Felder steuern, z.B. Bundesland oder Anrede einblenden, oder kann er eigens definierte Felder konfigurieren? Falls der Admin Nutzer eigene Felder definieren kann, sind Einstellungen zu Validierungen, z.B. eine minimale Anzahl und maximale Anzahl an Zeichen, möglich? Kann man Felder jeweils als optional oder erforderlich definieren?



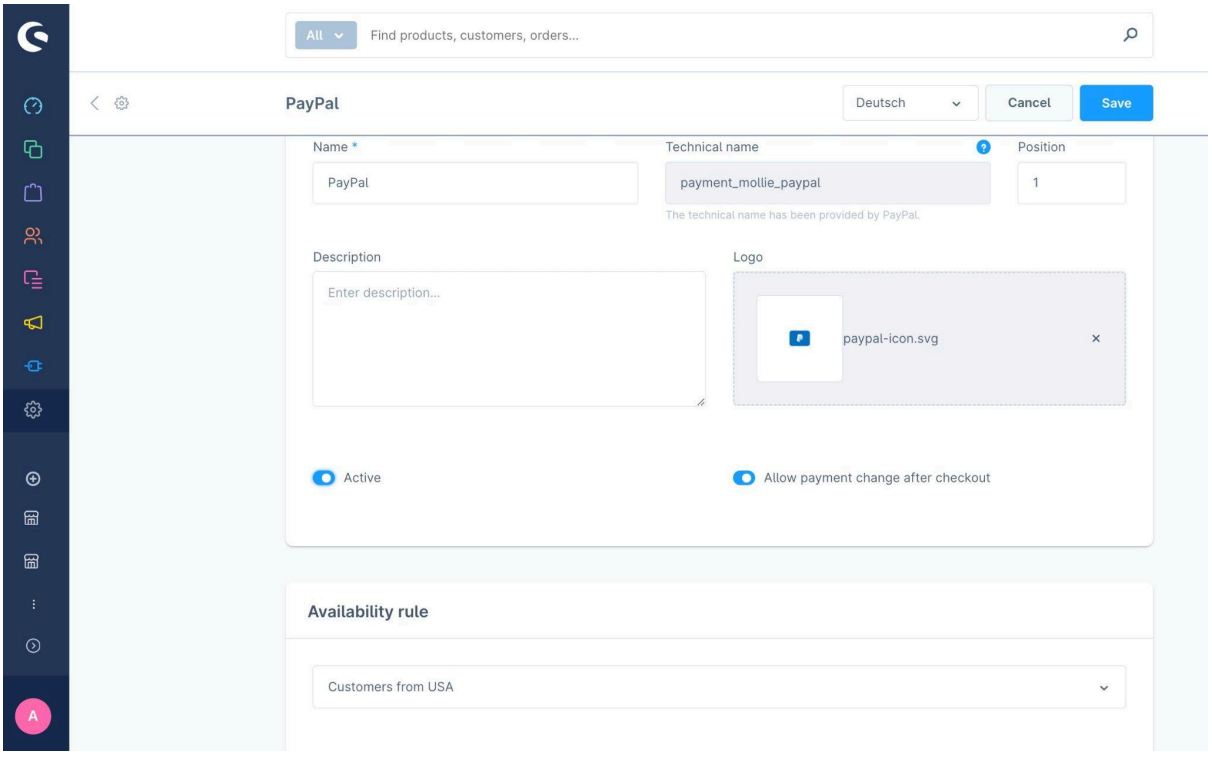
Konfiguration der Kunden- und Checkoutfelder in Magento 2

- Kann der Admin Nutzer die für den Check-out erlaubten Länder steuern? Kann jeweils eine Trennung zwischen “erlaubt für Versand” und “erlaubt für Rechnung” erfolgen? Können die definierten Adressfelder jeweils pro Land definiert werden?
- Können im Admin Bereich mehrere Steuersätze und Steuerregeln definiert werden? Können sich Produkte nach Steuerklasse unterscheiden, wie z.B. Bücher mit reduzierten Steuersatz? Können Kundenkonten nach Steuerklasse, z.B. falls der Kunde eine Firma mit VAT ID nutzt, gesteuert werden? Kann der Admin Nutzer unterschiedliche Steuersätze, jeweils mit Standard und reduziertem Steuersatz, pro Versandland definiert werden?



Konfiguration von Steuersätzen in Shopware 6

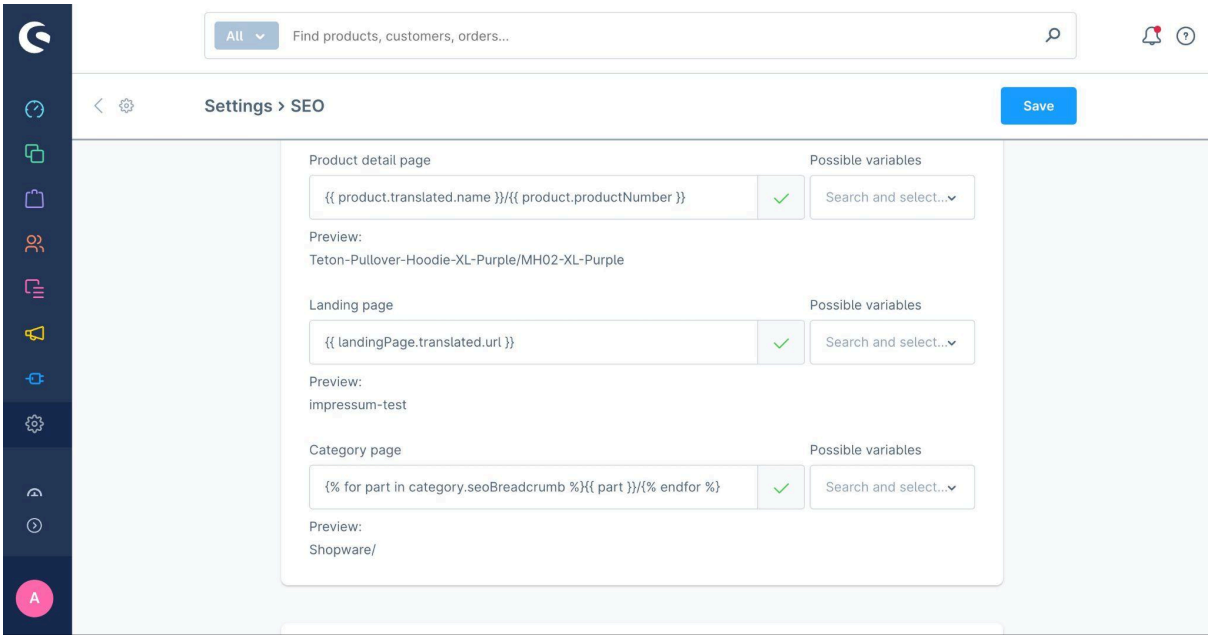
- Können für den Check-out unterschiedliche Versandmethoden, wie etwa “Standard” oder “Express” definiert werden? Können die Versandkosten jeweils pro Versandland gesteuert werden? Können mit Regeln gewisse Versandmethoden ausgeschlossen werden, z.B. kein Expressverwand unter einem Warenkorbwert von 50€? Kann der Admin Nutzer Versandkosten abhängig von einer Warenkorbeigenschaft, wie etwa “Gewicht” oder “Warenkorbwert”, berechnen lassen? Sind Versandmethoden wie “Abholung in der Filiale” oder “Packstation” abgebildet?
- Können für den Check-out unterschiedliche Zahlungsmethoden, wie etwa “Vorauskasse” oder “Rechnungskauf” definiert werden? Können mit bestimmten Regeln Zahlungsmethoden ausgeschlossen werden, z.B. kein Rechnungskauf unter einem Warenkorbwert von 150€? Ist es möglich, Kundeneigenschaften für das Ausschließen von Zahlungsmethoden zu nutzen, wie z.B. Kunden ohne bisherige Bestellung dürfen den Rechnungskauf nicht nutzen?
- Können Online Zahlungsmethoden, wie etwa Kreditkarte, Sofortüberweisung, Apple Pay oder Paypal, über eine Einstellung zum Payment Service Provider gesteuert werden? Kann man einzelnen Zahlungsmethoden weiter konfigurieren, und z.B. 3-D Secure bei Kreditkarten aktivieren? Kann der Admin Nutzer eine gesonderte Zahlungsgebühr, wie etwa zusätzlich 2% Gebühren für Kreditkarten, definieren? Ist eine Steuerung der Zahlungsmethoden pro Versandland möglich?



Konfiguration der Zahlungsmethode "Paypal" in Shopware 6

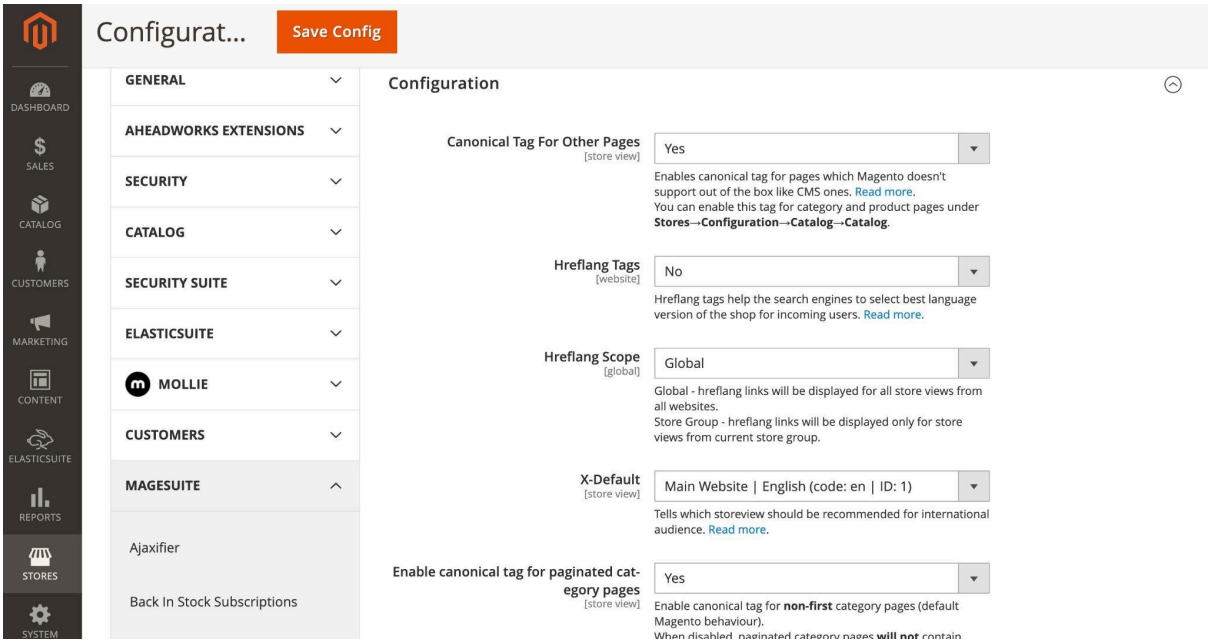
In der Regel will man als E-Commerce Owner die Storefront Inhalte, also Katalog- sowie Contentseiten, für Suchmaschinen indexieren, um eine gewisse Sichtbarkeit über die organischen Suchmaschinenergebnisse ("SEO") zu erhalten. Auch sind in der Regel einige Marketingwerkzeuge für das Marketing-Team, wie etwa ein Analytics Tool oder das Einspielen von Marketingpixeln, wichtig. Hierbei sollte der Adminnutzer etwa die folgenden Funktionen beschreiben:

- Gibt es eine Maske oder Regeln für die Generierung von URL Schlüsseln? [18] Kann man diese Regeln einzeln für Entitäten wie Produkte, Kategorien und Contentseiten definieren?



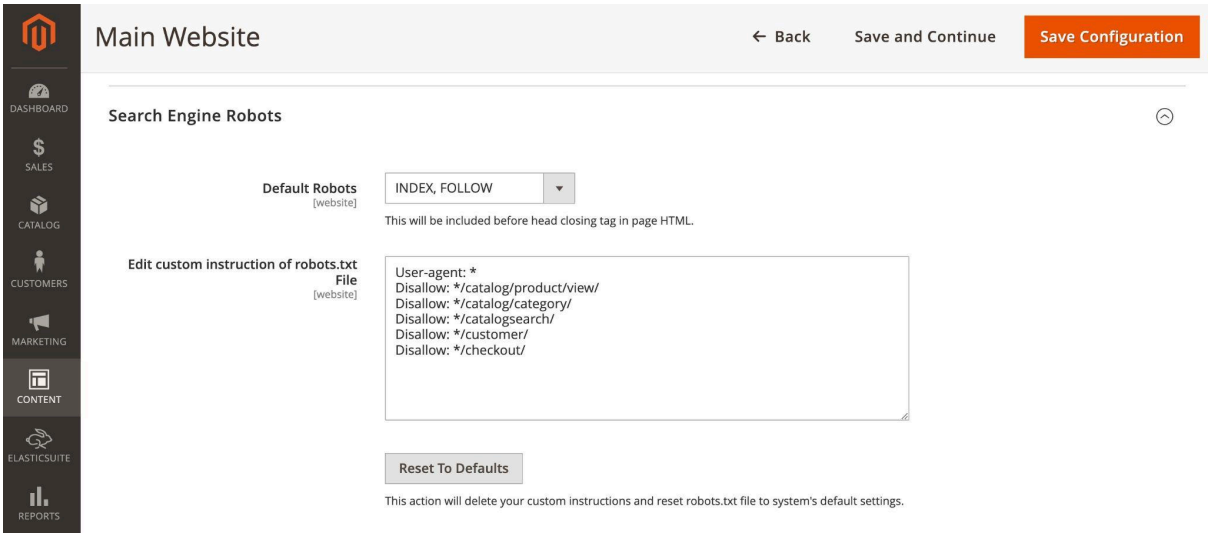
Konfiguration der URL Schlüssel in Shopware 6

- Kann der Admin Nutzer Einstellungen zu Canonical Tags oder Hreflang Tags tätigen? Können selbst referenzierende Canonical Tags als Default gesetzt werden, und diese ggf. pro Produkt-, Kategorie- oder Contentseite überschreiben? Können Hreflang Tags pro Mandant oder Sprache gesetzt werden?



Konfiguration der Meta Tags über das Plugin "MageSuite" in Magento 2

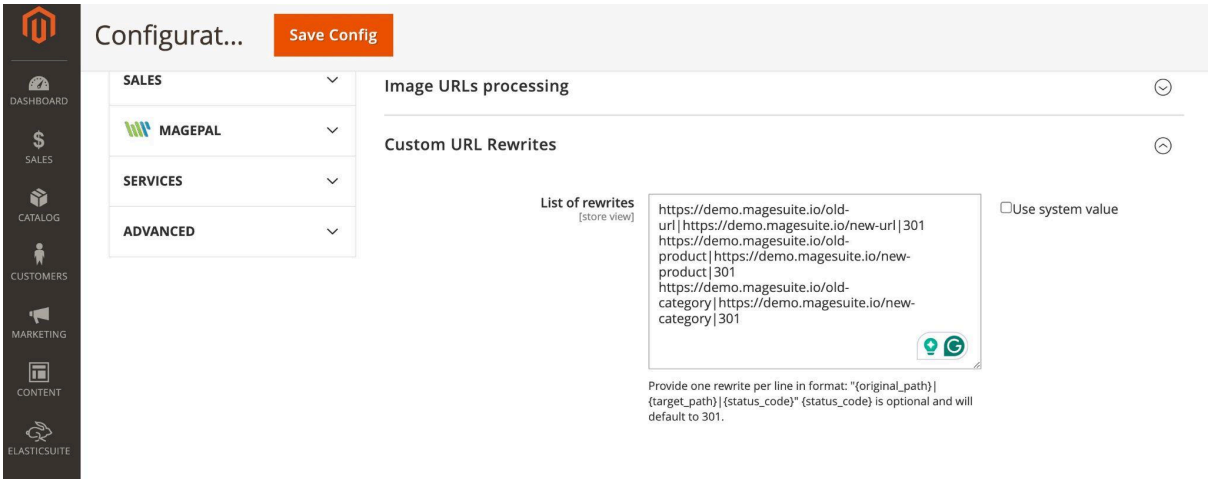
- Können Einstellungen zu Meta Robot Tags, wie etwa "nofollow" oder "noindex", gesetzt werden? Können diese Einstellungen pro Produkt, Kategorie oder Content-Seite getätigt werden?[19]
- Kann der Admin Nutzer die "robots.txt" Datei eigenständig editieren und somit z.B. bestimmte Bots aussperren?[20] Kann man Regeln zur Generierung der Sitemap tätigen und diese in der robots.txt Datei verlinken?[21]



Konfiguration der Meta Robot Tags und der Robots.txt Datei in Magento 2

- Können Einstellungen zu strukturierten Daten getätigt werden, wie z.B. Mappings zwischen Produktattributen und den Schema.org Tags? Können diese Einstellungen jeweils für Produkte, Kategorien und Contentseiten getätigt werden?[22]

- Können Einstellungen zu OpenGraph Meta Tags getätigt werden, wie z.B. Default Werte und Mappings zwischen dem OpenGraph Tag und einem Produktattribut? Können diese Einstellungen jeweils für Produkte, Kategorien und Contentseiten getätigt werden?[23]
- Kann der Admin Nutzer eigenständig URL Rewrites, also einen 301 oder 302 Redirect, einfügen?[24]

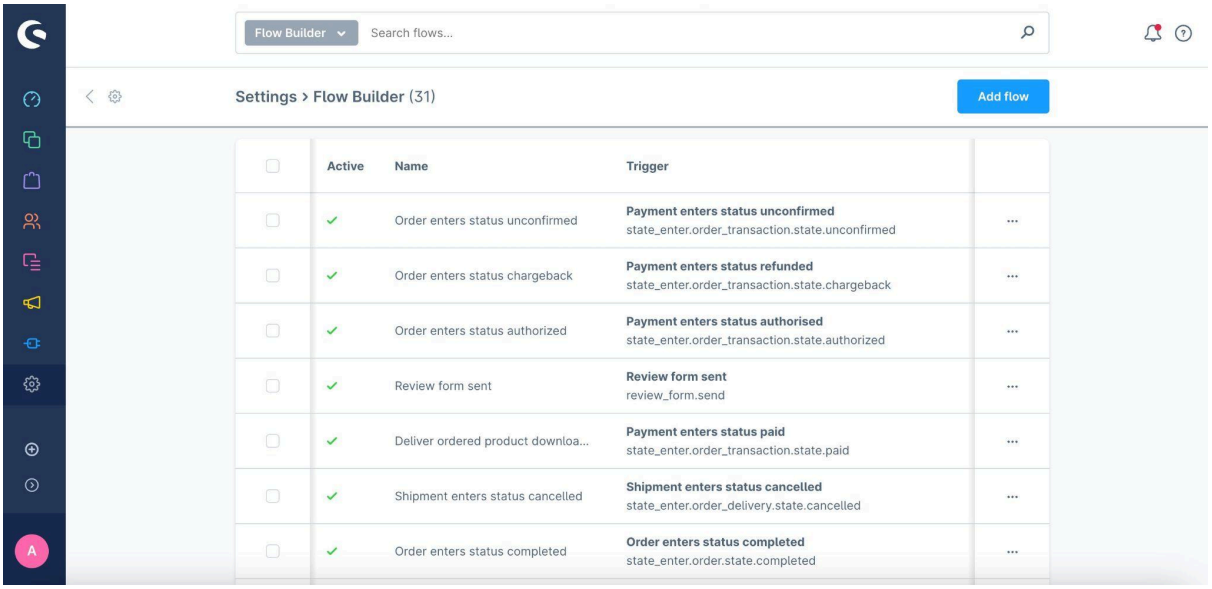


Konfiguration der URL Rewrites über das Plugin "MageSuite" in Magento 2

- Kann der Admin Nutzer eigenständig einen Google Analytics Container oder einen Google Tag Manager Container mit dem E-Commerce System verknüpfen?[25] Kann man pro Mandant bzw. Sub-shop einen Container hinterlegen?

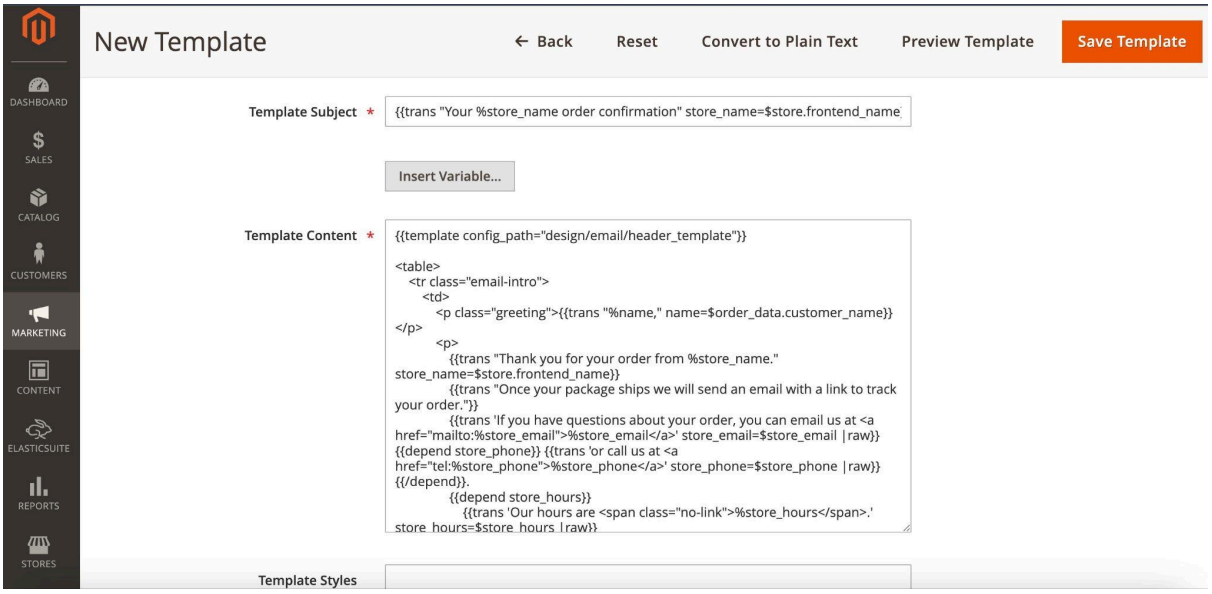
Ein weiterer wichtiger Bestandteil einer E-Commerce Plattform sind E-Mails, die vom System versandt werden. Hier sollte man etwa die folgenden Funktionen beschreiben:

- Kann der Admin Nutzer den E-Mail-Server eigenständig konfigurieren? Kann der Admin Nutzer ebenfalls die E-Mail Sender Adressen definieren?
- Kann der Admin Nutzer-Szenarien definieren (z.B. Bestellbestätigung, Passwort vergessen, neues Kundenkonto, ...), die einen E-Mail Versand auslösen?



Flowbuilder für E-Mail Benachrichtigungen in Shopware 6

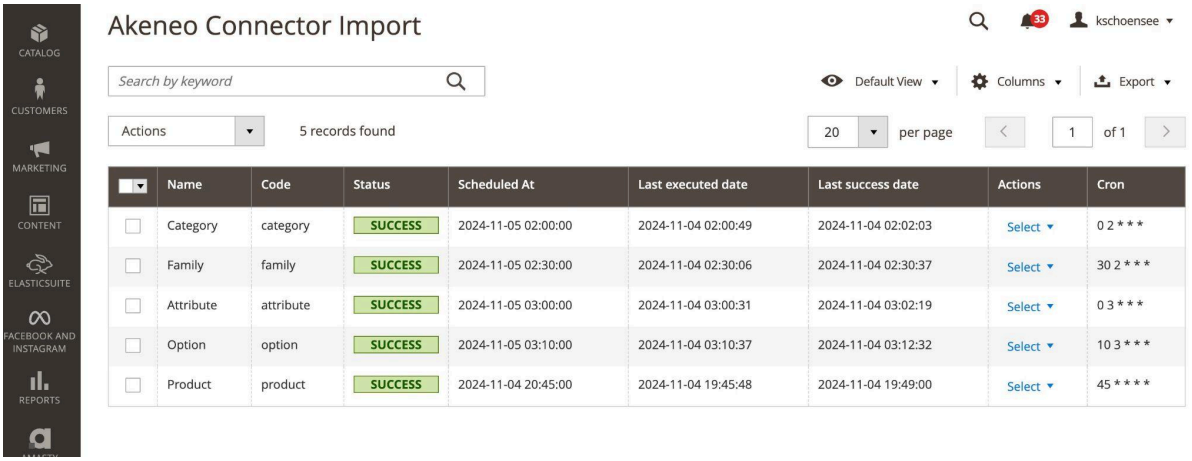
- Kann der Admin Nutzer E-Mail Templates gestalten und darin Texte, Styles usw. ändern? Kann man diese pro Sprache bzw. Sub-shop anpassen?



Erstellung von E-Mail Templates in Magento 2

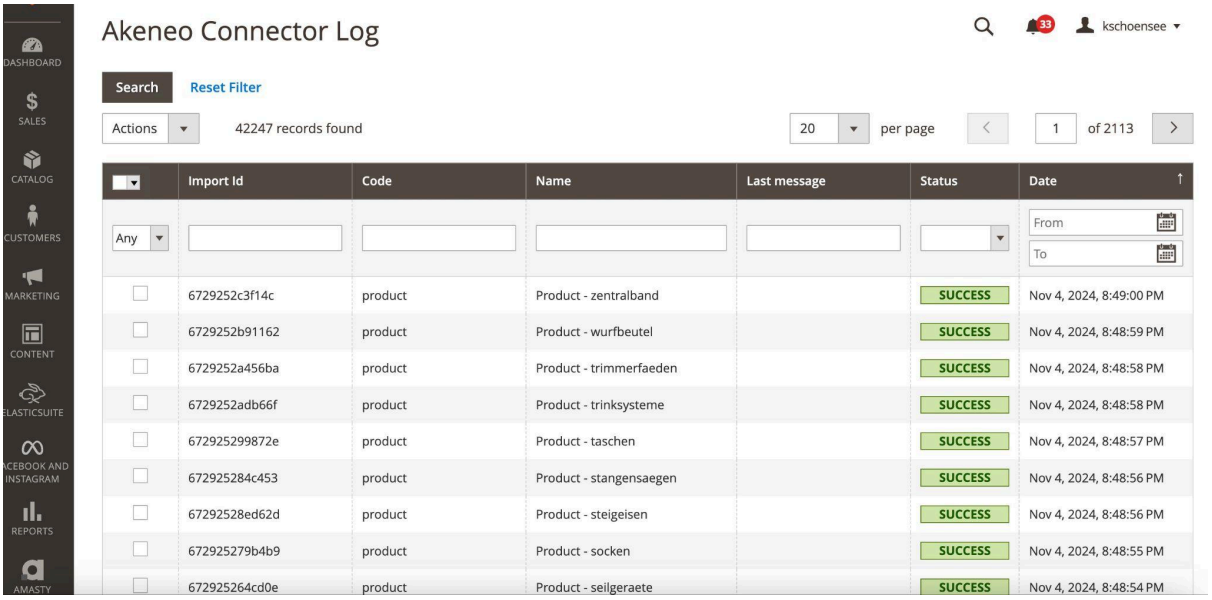
Oft gibt es in gängigen Shopsystemen einen Bereich, in dem reine Backend und Admin bezogene Einstellungen getätigt werden können. Dazu gehören etwa:

- Gibt es einen Bereich, in dem Cronjobs gesteuert werden können? Wird in der E-Commerce Plattform eine Indexierung und Caching von Inhalten genutzt, die der Admin Nutzer steuern kann? Können Jobs von einzelnen Integrationen, z.B. dem Produktimport, gesteuert werden?



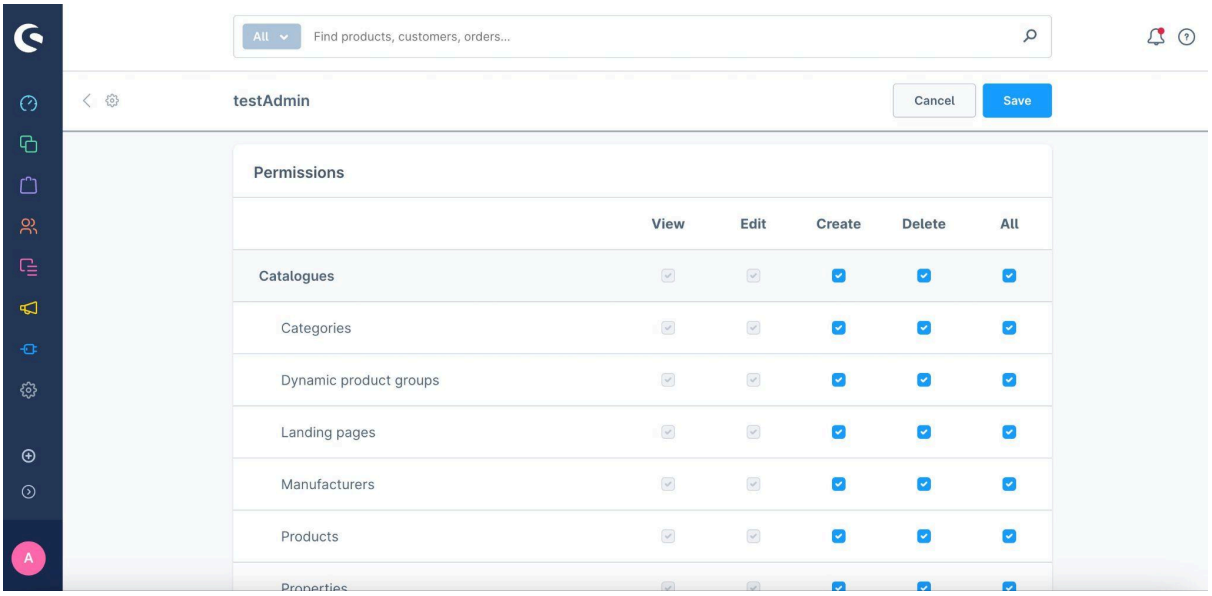
Konfiguration des Akeneo Connectors (PIM) in Magento 2

- Gibt es ein Logging von erfolgreicher und fehlerhafter Kommunikation mit Drittsystemen, wie z.B. fehlerhafte Importe vom Warenbestand aus dem ERP? Welche Informationen enthalten diese Logs? Werden Änderungen von Admin Nutzern ebenfalls geloggt und können diese sogar rückgängig gemacht werden?



Logging des Akeneo Connectors (PIM) in Magento 2

- Gibt es ein Rechte- und Rollensystem für Admin Nutzer? Können neben den Bereichen, z.B. Produkte, Content-Seiten, Rabattregeln, ebenfalls die Berechtigungen, wie Lesen, Erstellen, Ändern oder Löschen, angepasst werden? Können diese Berechtigungen auch für API User getätigt werden?



Definition von Rechten und Rollen in Shopware 6

Analyse der Systemlandschaft

E-Commerce-Systeme sind in der Regel eingebettet in eine komplexe Systemlandschaft, die aus verschiedenen, miteinander verknüpften Systemen besteht. Während das E-Commerce-System selbst vor allem den digitalen Verkauf von Produkten ermöglicht, werden andere Aufgaben von spezialisierten externen Systemen übernommen. Diese können folgende Funktionen abdecken:

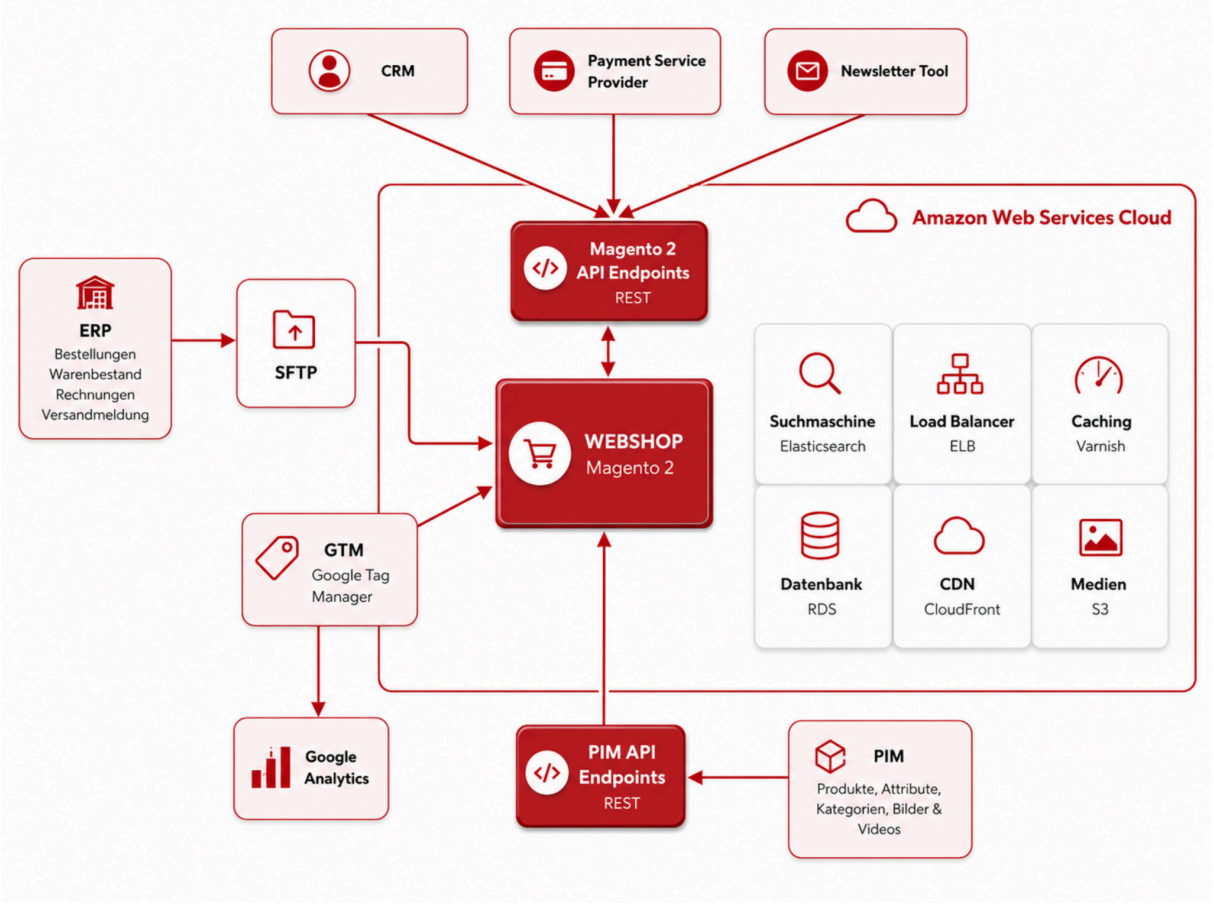
- Buchhaltung**:** Verarbeitung von Rechnungen, Zahlungen und Steuermeldungen.
- Fulfillment und Logistik: Abwicklung von Lagerbeständen, Versand und Retouren.
- Warenwirtschaftssysteme (ERP): Verwaltung von Beständen, Beschaffung und Produktionsplanung.
- Zahlungsabwicklung: Integration von Payment Service Providern (PSP) für sichere Transaktionen.
- Customer Relationship Management (CRM): Pflege und Analyse von Kundenbeziehungen.
- Produktdatenmanagement (PIM): Verwaltung und Anreicherung von Produktinformationen.
- Reporting und Business Intelligence (BI): Datenanalysen und Entscheidungsgrundlagen.
- Analytics und Tracking: Erfassung und Auswertung von Nutzerdaten.
- Searchandising: Lösungen für Suche, Produktempfehlungen und Personalisierung.
- Marketing Automation: Automatisierung von Kampagnen und Newsletter-Versand.

Um eine übersichtliche und effiziente Systemlandschaft zu schaffen, ist es empfehlenswert, die einzelnen Systeme und ihre Kommunikationswege präzise zu dokumentieren. Eine visuelle Darstellung als Schaubild kann dabei helfen, die Verbindungen und Datenflüsse zwischen den Systemen zu verdeutlichen.

Zusätzlich spielt die Infrastruktur der E-Commerce-Plattform eine entscheidende Rolle, insbesondere bei der Nutzung von on-premise Shopsystemen. Die Infrastruktur umfasst alle Ressourcen, die für den Betrieb der E-Commerce-Instanz erforderlich sind. Diese befinden sich entweder in einem lokalen Rechenzentrum oder in einer Cloud-Umgebung. Zu den typischen Komponenten gehören:

- Server: Physische oder virtuelle Maschinen, auf denen das E-Commerce Systemm installiert ist, inklusive Test-, Staging- und Produktionsumgebungen.

- Loadbalancer: Verteilt eingehenden Traffic auf mehrere Server, um eine gleichmäßige Auslastung zu gewährleisten.
- Datenbanken: Speicherung von Bestellungen, Kundendaten, Produktinformationen und anderen geschäftsrelevanten Daten.
- Medien-Storage: Speichert Bilder, Videos und andere Medieninhalte.
- Caching-Server: Optimiert die Ladezeiten durch Zwischenspeicherung häufig genutzter Daten.
- Content Delivery Network (CDN): Liefert statische Inhalte wie Bilder oder JavaScript von geografisch nahen Servern aus, um die Ladezeit zu verkürzen.
- Suchmaschinen-Server: Systeme wie ElasticSearch zur schnellen Verarbeitung von Suchanfragen und Filterfunktionen.



Im Folgenden werden einige Integrationen mit einem PIM, einem ERP, einem CRM sowie einem PSP als Beispiel beschrieben. Diese können als Anhaltspunkt bei der eigenen Analyse dienen.

PIM-Integration

Dieser Abschnitt bietet eine detaillierte Übersicht über die relevanten Aspekte, die bei der PIM-Integration berücksichtigt werden sollten, um eine reibungslose Migration der Schnittstelle zu gewährleisten.

Zunächst muss festgelegt werden, welche Entitäten zwischen dem PIM und dem E-Commerce System synchronisiert werden sollen. Typische Entitäten sind:

- Kategorien: Soll der gesamte Kategoriebaum, inklusive Unterkategorien, übertragen werden? Mit wie vielen Ebenen an Kategorien können das E-Commerce System und das PIM umgehen? Was passiert, wenn eine Kategorie im PIM verschoben oder gelöscht wird?
- Produkte: Erfolgt pro Import die Aktualisierung eines einzelnen Produktes (einzelne SKU) oder werden mehrere Produkte aktualisiert? Welche Produkttypen (einfache, konfigurierbare, digitale Produkte etc.) werden ausgetauscht? Wie viele Ebenen an konfigurierbaren Produkten gibt es (z.B. ist "Größe" das einzige konfigurierbare Attribut, oder gibt es "Größe" und "Farbe" als zwei konfigurierbare Attribute)?
- Medien: Welche Arten von Medien, z.B. Bilder, Videos, PDFs, werden synchronisiert? Gibt es im Import eine Reihenfolge, die beachtet werden muss, z.B. erstes Produktbild aus dem PIM immer auf der Produktkachel anzeigen, zweites Produktbild auf der Produktdetailseite, usw.?
- Attribute und Attributwerte: Werden Attribute zwischen dem PIM und E-Commerce System ausgetauscht?

Im nächsten Schritt wird der Abgleich von Datenstrukturen und Attributen, oder das "Mapping der Attribute" zwischen PIM und dem E-Commerce System, beschrieben:

- Attribut-Kompatibilität: Verfügen PIM und das E-Commerce System über dieselben Datentypen? Werden die Attributtypen zwischen dem E-Commerce System und PIM einheitlich erkannt und korrekt interpretiert, z.B. Texte mit HTML-Tags, numerische Werte, Singe-select, multi-select, usw.?
- Lokalisierung: Wie erfolgt die Übersetzung und Darstellung von Attributen, wie Produktnamen, Beschreibung, usw., in mehreren Sprachen? Wie werden unterschiedliche Währungen zwischen PIM und E-Commerce System gehandhabt?
- Mapping-Logik: Welches System ist für das Mapping der Attribute verantwortlich?[26] Im Rahmen der Mapping Beschreibung sollten ebenfalls diese Aspekte dokumentiert werden:
- Datenvalidierung: Festlegung von Validierungsregeln, wie z.B. maximale Zeichenanzahl, Pflichtfelder und zugelassene Zeichen, wie etwa Umlaute.

- Attribute-Mapping: Ein exaktes Mapping aller Attribute ist notwendig, um sicherzustellen, dass der Datenfluss zwischen PIM und dem E-Commerce System konsistent bleibt.

Anschließend sollte der Kommunikationsfluss und die technische Integration beschrieben werden. Darunter versteht man typischerweise:

- Datenfluss: Findet der Austausch unidirektional (nur vom PIM zum E-Commerce System) oder bidirektional (E-Commerce System meldet Daten zurück an das PIM) statt? Denkbar ist, dass etwa Produkte (und deren SKU) erstmalig im E-Commerce System entstehen, an das PIM gemeldet werden und angereichert werden, und anschließend die angereicherten Daten an das E-Commerce System zurückgespielt werden. Gibt es ein E-Commerce Testsystem, das mit einer PIM Sandbox integriert ist, oder wird das E-Commerce Testsystem mit dem Produktiv PIM integriert?
- Datenformat und Übertragungswege:
- Dateibasierter Austausch: Kommt ein Import von Dateien wie CSV, XML, oder JSON zum Einsatz? Wo werden die Dateien abgelegt und welches System bereinigt die Ablage?
- API-basierte Integration: Wenn eine API-Integration vorliegt, welche API-Endpunkte werden genutzt? Erfolgt ein Import der PIM Daten über die PIM Endpoints und "zieht" das E-Commerce System Daten (PULL Verfahren), oder werden die PIM Daten vom PIM an das E-Commerce System per Shop API Endpoints "gedrückt" (PUSH Verfahren)?[27]
- Operationen auf Entitäten: Welche Operationen tätigt das PIM mit den Entitäten im E-Commerce System? Kann das PIM Produkte sowohl erstellen als auch aktualisieren und löschen?
- Synchronisationsfrequenz: Wird die Synchronisation automatisch in festgelegten Intervallen, z.B. alle 60 Minuten, oder nach Bedarf ausgelöst?[28]

Um eine sichere Kommunikation zu gewährleisten, sind folgende Fragen zur Authentisierung zu beschreiben:

- Authentifizierungsmechanismen: Erfolgt der Zugriff durch Bearer Tokens, Basic Auth (Benutzername und Passwort im URL Header), Austausch eines SSH Keys oder zusätzliche Mechanismen?
- Zugriffsrestriktionen: Gibt es ein IP- oder Domain-Allowlisting, um die API-Anfragen oder den Zugriff auf einen SFTP Server zu kontrollieren?

ERP-Integration

Im folgenden Abschnitt werden relevanten Aspekte, die bei der ERP-Integration berücksichtigt werden sollten, beschrieben.

Zunächst muss festgelegt werden, welche Entitäten zwischen dem ERP und dem E-Commerce System synchronisiert werden sollen. Typische Entitäten sind in diesem Kontext:

- Bestellungen (Order Export): Werden Bestellungen vom E-Commerce System an das ERP automatisiert übermittelt? Findet eine manuelle Überprüfung der neuen Bestellungen im E-Commerce System statt, bevor diese an das ERP exportiert werden? Werden nur Bestellungen mit bestimmten Order Stati, wie z.B. “bezahlt” an das ERP übermittelt? Wird dieser Status vom PSP gesetzt? Was passiert mit Bestellungen der Zahlungsmethoden “Vorkasse” und “Rechnung”, deren Zahlungseingang später erfolgt? Können Bestellungen nachträglich im E-Commerce System geändert werden und wenn ja, wird ein weiterer Order Export angestoßen? Werden pro Export ein oder mehrere Bestellungen übermittelt?
- Warenbestand (Stock Import): Wird der Warenbestand automatisch vom ERP an das E-Commerce System gesendet? Wenn ja, gibt es einen Warenbestand pro Produkt, oder werden mehrere Warenlager, z.B. im Omnichannel Kontext, an das E-Commerce gemeldet? Welches System ist für die Reservierung der Bestände nach Erstellung einer neuen Bestellung zuständig? Wird das ERP den reservierten Bestand im nächsten Stock Import berücksichtigen?
- Order Stati und Dokumente (Versandmeldung, Rechnung und Retouren Import): Werden vom ERP einzelne Order Stati an das E-Commerce System gemeldet?
- In welchem System wird die Rechnung generiert? Wird die Rechnung im ERP generiert und an das E-Commerce System übermittelt? Soll diese im Storefront für den Kunden einsehbar sein oder per E-Mail verschickt werden?
- In welchem System wird die Versandmeldung erstellt? Wird die Versandmeldung inkl. einem Sendungslink an das E-Commerce System übermittelt? Soll diese im Storefront für den Kunden einsehbar sein oder per E-Mail verschickt werden? Wo wird das physische Printlabel für die Pakete erstellt?
- In welchem System werden Retouren angemeldet und erstellt? Gibt es im Storefront eine Möglichkeit, eine Retoure anzumelden? Wenn ja, wird diese an das ERP gemeldet? Wird alternativ das Retourenlabel beim Paket mitgeliefert und wird die Retoure erstmalig beim ERP angemeldet? Wenn diese Retoure an das

E-Commerce System gemeldet wird, welches System steuert die Rückerstattung beim PSP an? Welches System erstellt die Gutschrift und meldet die diese an den Kunden?

- Kunden: Im B2B Kontext ist es üblich, dass Kunden zwischen dem E-Commerce System und dem ERP oder einem CRM kommuniziert werden. Wie sieht der Kundenregistrierungsprozess aus? Können Nutzer eigenständig ein Konto im Storefront erstellen und anschließend Bestellungen tätigen, oder erfolgt zunächst eine Synchronisation mit dem ERP? Ist ein Freigabeprozess von Kunden zwischen dem E-Commerce System und ERP bzw. CRM eingeführt? Welches System ist für die Verwaltung von Kundendaten zuständig? Falls das ERP die Hoheit über Kundendaten verfügt, können Kunden in ihrem Kundenkonto im Storefront bestimmte Daten, wie z.B. Firmenname, Adresse, VAT ID, nicht ändern dürfen?
- Preise und Preislisten: Werden Preise im ERP oder im PIM verwaltet? Werden diese automatisiert an das E-Commerce System übermittelt? Im B2B Kontext ist denkbar, dass das ERP Preislisten pro Kundenkonto führt. Falls ein Kunde keine Preisliste hat, gilt eine bestimmte “Default” Preisliste bzw. Listenpreise?

Analog zur PIM Integration, sollte das Mapping der unterschiedlichen Datenfelder zwischen dem ERP und E-Commerce System beschrieben sein[29]:

- Lokalisierung: Falls das E-Commerce System mehrere Sprachen, Währungen und Mandanten bedient, wie werden diese im ERP entsprechend zugeordnet?
- Mapping-Logik: Welches System ist für das Mapping der Attribute verantwortlich? Im Rahmen der Mapping Beschreibung sollten ebenfalls diese Aspekte dokumentiert werden:
- Datenvalidierung: Festlegung von Validierungsregeln, wie z.B. maximale Zeichenanzahl, Pflichtfelder und zugelassene Zeichen.
- Attribute-Mapping: Ein exaktes Mapping aller Felder ist notwendig, um sicherzustellen, dass der Datenfluss zwischen ERP und E-Commerce System konsistent bleibt.

Im nächsten Schritt sollte der Kommunikationsfluss und die technische Integration beschrieben werden. Darunter versteht man typischerweise:

- Datenfluss: Findet der Austausch unidirektional (nur vom ERP zum E-Commerce System) oder bidirektional (E-Commerce System meldet Daten zurück an das ERP) statt? Falls Kundenkonten sowohl im ERP und dem Storefront geändert werden können, ist eine bidirektionale Kommunikation üblich. Gibt es ein E-Commerce Testsystem, dass mit einer ERP Sandbox integriert ist?

- Datenformat und Übertragungswege:
- Dateibasierter Austausch: Kommt ein Export und Import von Dateien wie CSV, XML, oder JSON zum Einsatz? Wo werden die Dateien abgelegt und welches System bereinigt die Ablage?
- API-basierte Integration: Wenn eine API-Integration vorliegt, welche API-Endpunkte werden genutzt? Erfolgt der Order Export über die ERP Endpoints und "drückt" das E-Commerce System Daten (PUSH Verfahren), oder werden die Bestellungen vom ERP per Shop API Endpoints "gezogen" (PULL Verfahren)?
- Operationen auf Entitäten: Welche Operationen tätigt das ERP mit den Entitäten im E-Commerce System? Kann das ERP z.B. Kunden oder Bestellungen erstellen, aktualisieren und löschen?
- Synchronisationsfrequenz: Wird die Synchronisation automatisch in festgelegten Intervallen (z.B. alle 60 Minuten) oder nach Bedarf ausgelöst?

Schließlich sind folgende Punkte zur Authentisierung zu beschreiben:

- Authentifizierungsmechanismen: Erfolgt der Zugriff durch Bearer Tokens, Basic Auth (Benutzername und Passwort), Austausch eines SSH Keys oder zusätzliche Mechanismen?
- Zugriffsrestriktionen: Gibt es ein IP- oder Domain-Allowlisting, um die API-Anfragen oder den Zugriff auf einen SFTP Server zu kontrollieren?

CRM-Integration

Dieser Abschnitt behandelt Aspekte, die bei einer CRM-Integration berücksichtigt werden sollten. Da sich viele Funktionen bei gängigen Marketing Automation- und E-Mail Marketing Anbietern ähneln und teilweise CRM Funktionen enthalten sind, kann dieser Abschnitt ebenso für die Integration von Marketing Automation- und E-Mail Marketing Tools angewandt werden.

Im ersten Schritt muss festgelegt werden, welche Daten zwischen dem CRM und dem E-Commerce System synchronisiert werden sollen. Typische Entitäten sind:

- Kunden: Werden Kunden zwischen dem E-Commerce System und CRM synchronisiert? Welches System ist für die Verwaltung von Kundendaten zuständig? Falls das CRM die Hoheit über Kundendaten verfügt, können Kunden in ihrem Kundenkonto im Storefront bestimmte Daten, wie z.B. Adresse, nicht ändern dürfen?

- Bestellungen: Werden Bestellungen vom E-Commerce System an das CRM übermittelt? Werden nur Bestellungen mit bestimmten Order Stati, wie z.B. “bezahlt”, an das CRM übermittelt?
- Interaktionen: Welche Interaktionen werden vom E-Commerce System an das CRM gemeldet? Sind es bspw. Warenkorbabbrüche, Newsletter Abos, Support Anfragen, Downloads, besuchte Seiten, Suchanfragen oder Ähnliche?
- Gutscheincodes, Preis- und Rabattregeln: In welchem System werden Gutscheine, Preis- und Rabattregeln verwaltet? Falls die Regeln im CRM verwaltet werden, werden diese an das E-Commerce System übermittelt? Werden im CRM Gutscheincodes erstellt, die im Storefront eingelöst werden können? Haben diese Gutscheincodes einen Restwert und können die Gutscheine wiederverwendet werden (quasi Gutscheinkarte), oder kann man die Codes nur einmalig ohne Restwert verwenden? Können zwischen dem E-Commerce System und CRM die gleichen Modelle an Warenkorbpreisregeln abgebildet werden?[30]
- Treuepunkte: Gibt es im Storefront des E-Commerce Systems ein Treuepunkteprogramm, welches einen Austausch zwischen CRM und E-Commerce System bedarf? Wo werden die gesammelten Treuepunkte, und die Regeln zum Sammeln und Einlösen von Treuepunkten verwaltet? Für welche Aktionen (Transaktionen oder Interaktionen) werden Treuepunkte gesammelt?
- Produktempfehlungen (Recommendation Engine): Werden im CRM Produktempfehlungen anhand der Kundendaten, Bestellungen und Interaktionen erstellt? Wenn ja, wo werden diese im Storefront ausgespielt?

Auch im Rahmen einer CRM-Integration sollte das Mapping der unterschiedlichen Datenfelder zwischen dem CRM und E-Commerce System beschrieben sein:

- Lokalisierung: Falls das E-Commerce System mehrere Sprachen, Währungen und Mandanten bedient, wie werden diese im CRM entsprechend zugeordnet?
- Mapping-Logik: Welches System ist für das Mapping der Attribute verantwortlich? Im Rahmen der Mapping Beschreibung sollten ebenfalls diese Aspekte dokumentiert werden:
- Datenvalidierung: Festlegung von Validierungsregeln, wie z.B. maximale Zeichenanzahl, Pflichtfelder und zugelassene Zeichen.
- Attribute-Mapping: Ein exaktes Mapping aller Felder ist notwendig, um sicherzustellen, dass der Datenfluss zwischen CRM und E-Commerce System konsistent bleibt.

Im nächsten Schritt sollte der Austausch aus technischer Sicht beschrieben werden. Darunter versteh man typischerweise:

- Datenfluss: Findet der Austausch unidirektional (nur vom CRM zum E-Commerce System) oder bidirektional (E-Commerce System meldet Daten zurück an das CRM) statt? Falls Kundenkonten sowohl im CRM und dem Storefront geändert werden können, ist eine bidirektionale Kommunikation üblich. Gibt es ein E-Commerce Testsystem, das mit einer CRM Sandbox integriert ist?
- Datenformat und Übertragungswege:
- Dateibasierter Austausch: Kommt ein Export und Import von Dateien wie CSV, XML, oder JSON zum Einsatz? Wo werden die Dateien abgelegt und welches System bereinigt die Ablage?
- API-basierte Integration: Wenn eine API-Integration vorliegt, welche API-Endpunkte werden genutzt? Erfolgt der Export eines neuen Shop Kunden über die CRM Endpoints und "drückt" das E-Commerce System Daten (PUSH Verfahren), oder werden neue Shop Kunden vom CRM per Shop API Endpoints "gezogen" (PULL Verfahren)?
- Operationen auf Entitäten: Welche Operationen tätigt das CRM mit den Entitäten im E-Commerce System? Kann das CRM z.B. Kunden oder Bestellungen erstellen, aktualisieren und löschen?
- Synchronisationsfrequenz: Wird die Synchronisation automatisch in festgelegten Intervallen (z.B. alle 60 Minuten) oder nach Bedarf ausgelöst?

Auch bei einer CRM-Integration sind einige Aspekte bzgl. der Authentisierung zu beschreiben:

- Authentifizierungsmechanismen: Erfolgt der Zugriff durch Bearer Tokens, Basic Auth (Benutzername und Passwort), Austausch eines SSH Keys oder zusätzliche Mechanismen?
- Zugriffsrestriktionen: Gibt es ein IP- oder Domain-Allowlisting, um die API-Anfragen oder den Zugriff auf einen SFTP Server zu kontrollieren?

Schließlich sollte im Rahmen einer CRM-Integration beschrieben sein, welche Kommunikation direkt vom CRM an den Kunden geht, und damit abseits vom E-Commerce System stattfindet. Beispiele hierfür können sein:

- E-Mails, wie etwa Newsletter oder Aktiverungsmails mit Gutscheinen
- SMS Benachrichtigungen zu neuen Rabattaktionen
- Push-Benachrichtigungen mit Gutscheinen oder neuen Produkten

PSP-Integration

Viele gängige PSP bieten fertige Extensions zur E-Commerce-Integration an.[31] In diesem Fall wird die technische Integration, meistens per API Schnittstelle, von der Extension geliefert. Der E-Commerce Owner definiert im Rahmen der Extension, welches PSP Konto (welche API User, welcher Mandant, usw.) im E-Commerce System genutzt werden soll. Auch kann der E-Commerce Owner die gewünschten Zahlungsmethoden, oft per Lieferland, definieren. Der Vorteil für den E-Commerce Owner im Vergleich zu einer eigenen Integration liegt darin, dass der Extension-Anbieter die Integration wartet und der E-Commerce Owner laufenden Kosten für Wartung und Instandhaltung reduziert sind. Der E-Commerce Owner spart sich zudem die initialen Aufwände für die Integration. Insgesamt sollten bei der PSP Integration die folgenden Punkte beschrieben sein:

- Wie werden Bestellungen zwischen dem E-Commerce System und dem PSP abgeglichen? Wird eine eigene Transaktion-ID vom PSP generiert und zwischen E-Commerce System und PSP synchronisiert? Falls nicht, wird alternativ die vom E-Commerce System vergebene Order ID genutzt?
- Welche Zahlungsmethoden (pro Land) sind aktiviert?
- Gibt es besondere Regeln für die Verfügbarkeit von Zahlungsmethoden, z.B. Rechnungskauf nur bei eingeloggten Kunden?
- Gibt es Aufschläge bei einigen Zahlungsmethoden, z.B. 2% bei Kreditkarten Bestellungen?
- Welche Order Stati werden im E-Commerce System über die PSP Integration gesteuert?
- Wie wird der PSP im Check-out im Storefront eingebunden? Wird der Nutzer auf eine externe Seite vom PSP geleitet, oder finden alle Eingaben zu Zahlungsmethoden im Storefront statt?
- Erlaubt die Integration eine Synchronisation von Rückerstattungen? Wenn ja, werden Erstattungen im E-Commerce System oder im Admin Bereich des PSP ausgelöst? Wird eine Erstattung in beide Richtungen synchronisiert? Sind Teilerstattungen, z.B. einzelne Produkte und deren Mengen, unterstützt?
- Gibt es ein Logging der Kommunikation zwischen E-Commerce System und PSP? Kann man hier etwa fehlgeschlagene Transaktionen und deren Fehlercode einsehen? Kann man auf Ebene der Bestellungen die bisherige PSP Kommunikation einsehen?

Analyse der Business-Aspekte

Um die Notwendigkeit und den Umfang eines Replatforming-Projekts fundiert zu bewerten, ist eine detaillierte Analyse aus betriebswirtschaftlicher Sicht unerlässlich. Im Folgenden sind die zentralen Business-Aspekte aufgeführt, die im Rahmen eines Replatformings analysiert und dokumentiert werden sollten:

- Kosten und Total Cost of Ownership (TCO): Hierunter sind alle Kosten zu verstehen, die im Rahmens des Betriebs und Instandhaltung der aktuellen E-Commerce Plattform anfallen. Dazu gehören etwa Lizenzkosten für die Software des E-Commerce Systems, sowie Lizenzen für genutzte Extensions, Connectoren oder angeschlossene Systeme (PIM, ERP, usw.). Auch fallen etwaige Transaktionsgebühren oder Provisionen, z.B. für den PSP, unter die TCO. Hostinggebühren sind ebenfalls relevante Kosten. Schließlich entstehen Personal- oder Agenturkosten für die technische Entwicklung und Wartung des aktuellen Systems.[32]
- Marketing und SEO: Hier sollten alle Aspekte und Daten bzgl. dem Online-Marketing und SEO beschrieben sein. Insbesondere sollten KPIs und Metriken in einem SEO Überwachungstool oder der Google Search Console genannt sein. Relevant sind etwa der Sichtbarkeitsindex, monatliche Impressionen, Anzahl der indexierten Seiten, Anzahl semantischer Fehler usw. Marketingausgaben oder CPC Gebühren sollten hier ebenfalls genannt sein. Schließlich sollte eine Bestandsaufnahme aus dem Analytics Tool mit KPIs wie Umsatz, durchschnittlicher Warenkorbwert, Conversion Rate und dem Traffic nicht fehlen. Sofern verfügbar, sollten auch der Customer Lifetime Value sowie die durchschnittlichen Akquisitionskosten pro Kunde beschrieben sein.
- Kundenbeschwerden und Nutzerzufriedenheit: Hier sollten die Beschwerden von Kunden und Nutzern beschrieben werden, die ggf. schon in den vorherigen Kapiteln als Signale für das Replatforming wahrgenommen wurden. Beschwerden von Nutzern können in Form von der Anzahl an Supportanfragen oder Nutzerfeedbacks wahrgenommen werden. Nutzerbeschwerden können als ausbleibende Conversions und damit entgangener Umsatz interpretiert werden. Mitarbeiter äußern Ihre Beschwerden eher auf informellem Weg, per E-Mail oder mündlich. Diese gilt es dennoch schriftlich festzuhalten. Mitarbeiterbeschwerden sind ein Indiz für ineffiziente und zeitintensive Prozesse, die wiederum unnötige Personalkosten bedeuten.

✓ CHECKLISTE IST-ANALYSE DER AKTUELLEN E-COMMERCE PLATTFORM

- Die aktuelle E-Commerce Plattform wurde unter technologischen Aspekten mithilfe relevanter KPIs beschrieben.
- Die aktuelle E-Commerce Plattform wurde unter funktionalen Aspekten bzgl. des Storefronts beschrieben, darunter:
  - Funktionen im Header-Bereich
  - Suche und Navigation
  - Sortierung und Filter auf der Kategorieseite
  - Funktionen auf Content-, Blog- und FAQ Seiten
  - Inhalte und Funktionen auf der Produktdetailseite
  - SEO-relevante Funktionen (Meta-Tags, Sitemap, ...)
  - Warenkorb und Checkout
  - Login und Kundenbereich
  - Kontaktmöglichkeiten
  - User Consent und DSGVO-relevante Funktionen
  - E-Mail Versand
- Die aktuelle E-Commerce Plattform wurde unter funktionalen Aspekten bzgl. des Admin Bereichs beschrieben, darunter:
  - Mandantenfähigkeit und Sub-Shops
  - Verwaltung vom Katalog (Attribute, Kategorien und Produkte)
  - Contentmanagement und Pagebuilder
  - Konfigurationen der Suche und Filter
  - Bestellverwaltung
  - Kundenverwaltung
  - Rabatt- und Preisregeln
  - Einstellungen für den Checkout (Liefer- und Zahlungsmethoden, Steuern, ...)
  - SEO-relevante Einstellungen (Sitemap, Meta-Tags, ...)
  - E-Mails und Benachrichtigungen
  - Integrationen (API-User, Produkt-Feeds, Import- und Exportprofile, ...)
- Die Systemlandschaft der aktuellen E-Commerce Plattform wurde mit einer Grafik sowie einer kurzen technischen Beschreibung der Schnittstellen dokumentiert.
- Falls ein PIM integriert wurde, sind die ausgetauschten Entitäten (z.B. Attribute, Preise, Produkte, ...) sowie die Schnittstelle aus technischer Sicht beschrieben.

- Falls ein ERP integriert wurde, sind die ausgetauschten Entitäten (z.B. Bestellungen, Warenbestand, Kunden ...) sowie die Schnittstelle aus technischer Sicht beschrieben.
- Falls ein CRM integriert wurde, sind die ausgetauschten Entitäten (z.B. Kunden, Bestellungen, ...) sowie die Schnittstelle aus technischer Sicht beschrieben.
- Weitere externe Systeme und deren Schnittstellen (Payment-Service-Provider, Analytics, Tracking, Produkt-Feeds, ...) wurden aus technischer Sicht beschrieben.
- Die aktuelle E-Commerce Plattform wurde unter Business Aspekten mit relevanten KPIs beschrieben

3.2 Stakeholder Analyse

Eine sorgfältige Stakeholder-Analyse ist ein wichtiger Bestandteil eines E-Commerce Replatforming Projekts, da sie zu den funktionalen Anforderungen beisteuert und eine reibungslose Koordination und Kommunikation des Projekts sicherstellt. Im ersten Schritt geht es darum, die relevanten Stakeholder zu identifizieren. Zu den **internen Stakeholdern** zählen etwa das Management, Mitarbeiter wie Produktdatenmanager, Content-Manager, das Marketing- und Vertriebsteam, die IT-Abteilung sowie der Kundenservice. **Externe Stakeholder** umfassen vor allem die Endkunden, Lieferanten (diese können im B2B Kontext besonders relevant sein) sowie Agenturen (Design, Content, Marketing, ...) oder externe Entwicklungspartner.

Im nächsten Schritt sollten die aktuellen **Herausforderungen (Pains)**, Bedürfnisse und **Interessen (Gains)** der Stakeholder analysiert werden. Das Management legt beispielsweise großen Wert auf Budgettreue und eine positive Rendite des Projekts. Kunden erwarten eine verbesserte Nutzererfahrung und schnellere Ladezeiten, während Mitarbeiter von verbesserten Prozessen und Zeitersparnissen profitieren wollen. Externe Partner wie Agenturen bringen Expertise ein, erwarten jedoch klare Briefings, reibungslose Deployment-Prozesse und eine angenehme Zusammenarbeit.

Ein weiterer wichtiger Aspekt ist der **Einfluss**, den die verschiedenen Stakeholder auf das Projekt haben. Das Management hat in der Regel einen hohen Einfluss, da es die strategische Richtung vorgibt und das Budget sponsert. Kunden spielen eine zentrale Rolle, da ihre Zufriedenheit direkten Einfluss auf den Umsatz hat. Auch Mitarbeiter wie Produktdaten- oder Content-Manager sind entscheidend, da ihre Akzeptanz und Mitarbeit den Erfolg der Umsetzung beeinflussen. Eine klare Kommunikation und enge Zusammenarbeit mit allen Stakeholdern sind deshalb essenziell, um deren Unterstützung und Vertrauen sicherzustellen. Die Stakeholder Analyse legt den Grundstein, um das Replatforming aus der Perspektive Change Management zu meistern.

✓ CHECKLISTE STAKEHOLDER ANALYSE

- Es wurden interne (Mitarbeiter verschiedener Teams) sowie externe Stakeholder (Kunden, Lieferanten, ...) beleuchtet. Der Einfluss der einzelnen auf den Projekterfolg wurde berücksichtigt.
- Die Pains und Gains der jeweiligen Stakeholder sind identifiziert.

3.3 Wettbewerber Analyse

Eine Wettbewerberanalyse hilft nicht nur, die eigene Marktposition besser zu verstehen, sondern liefert auch wertvolle Einblicke, um sich von der Konkurrenz abzuheben und gezielte Optimierungen für das Replatforming vorzunehmen.

Zunächst sollten die eigenen **Wettbewerber identifiziert** werden. Die Suche nach relevanten Wettbewerbern beginnt mit Tools und Plattformen, die auf Datenanalyse spezialisiert sind. KI-Tools (z.B. ChatGPT und Google Gemini) und Online-Marketing-Plattformen (z.B. Sistrix oder SEMrush)[33] bieten wertvolle Einblicke in Suchmaschinenrankings, Sichtbarkeit und Marktaktivitäten. Eine einfache Google-Suche nach den eigenen Keywords zeigt außerdem, welche Online-Shops in der Branche sichtbar sind. Darüber hinaus geben Marktplätze wie Amazon oder eBay sowie Social-Media-Kanäle Aufschluss darüber, welche Unternehmen ähnliche Produkte anbieten und wie sie diese vermarkten.

Im nächsten Schritt sollte man mehr Informationen zur **Positionierung der Wettbewerber** herausfinden. Um die Positionierung der Wettbewerber zu verstehen, sollte deren Produktsortiment, Zielgruppenansprache und Vertriebskanäle analysiert werden. Zu prüfen ist, ob sie ähnliche Produkte anbieten und wie stark sich das eigene Produktsortiment überschneidet. Auch die genutzten Vertriebskanäle, wie Online-Marktplätze, Direktvertrieb, Partnervertrieb bzw. Affiliates über die eigene Website. Die Präsenzen der Wettbewerber auf Social-Commerce-Plattformen geben Hinweise auf ihre strategische Ausrichtung und die angesprochene Zielgruppe.

Eine Wettbewerberanalyse sollte auf klar **definierten Kriterien** basieren. Dazu gehören Preise, um die Preisstrategie der Konkurrenz zu verstehen, sowie Marktanteile, um deren Reichweite und Relevanz einzuschätzen. Technische Kriterien wie Ladezeiten und SEO-Metriken (z. B. Sichtbarkeitsindex, Rankings) zeigen die Performance des E-Commerce Systems. Die Benutzererfahrung (UX) und eine funktionale und qualitative Analyse des Storefronts, z. B. Qualität der Assets, Navigation, Filteroptionen, Checkout-Prozess, geben Einblicke in die Qualität der Webseiten der Wettbewerber.

Eine **SWOT-Analyse** (Strengths, Weaknesses, Opportunities, Threats) kann dabei helfen, die Stärken und Schwächen der Wettbewerber systematisch mit den eigenen

zu vergleichen. Beispielsweise könnte ein Wettbewerber mit einer starken SEO-Performance glänzen, während der eigene Online-Shop durch bessere Produktdaten oder personalisierte Empfehlungen hervorsticht. Schwächen der Wettbewerber, wie fehlende mobile Optimierung oder langsame Ladezeiten, bieten dir Chancen, sich abzuheben. Gleichzeitig hilft die Analyse, mögliche Bedrohungen wie aggressive Preisstrategien der Konkurrenz frühzeitig zu erkennen.

Zusammenfassend lässt sich festhalten, dass die Wettbewerberanalyse nicht nur wertvolle Erkenntnisse über die Positionierung und Strategie deiner Konkurrenz liefert, sondern auch Ansatzpunkte, um die eigenen Anforderungen an die neue E-Commerce-Plattform zu definieren.

✓ CHECKLISTE WETTBEWERBER ANALYSE

- Es wurden relevante Wettbewerber inkl. der Positionierung im Markt identifiziert.
- Die Wettbewerber werden mithilfe von KPIs relativ zum eigenen Unternehmen eingeordnet.
- Die Stärken und Schwächen der Wettbewerber sind ebenfalls bekannt.

3.4 Zielgruppenanalyse

Die Zielgruppenanalyse hilft, das E-Commerce System gezielt auf die Bedürfnisse und Erwartungen der Kunden auszurichten. Sie liefert die Grundlage, um das Angebot, die Ansprache und die Plattform optimal auf die Zielgruppe abzustimmen. Die Analyse beginnt mit der **Sammlung von Daten aus verschiedenen Quellen**. Analytics Tools wie Google Analytics, Matomo[34] und Social Media Insights liefern wertvolle KPIs zum Verhalten und Präferenzen der Nutzer. Spezialisierte Tools zur Nutzeranalyse, wie Hotjar[35], können das Nutzerverhalten in Form von Videos, Click Heatmaps und Scrollcharts visualisieren. Im B2B-Kontext können zusätzlich Daten aus CRM- und ERP-Systemen genutzt werden, während externe Studien und statistische Datenbanken branchenspezifische Trends aufzeigen können.

Eine erfolgreiche Zielgruppenanalyse erfordert die **Definition relevanter Merkmale**. Dazu gehören demografische Faktoren wie Alter, Geschlecht, Region, Sprache und Einkommen. Technische Aspekte wie verwendete Browser und Endgeräte sind ebenfalls wichtig, um die technischen Anforderungen des neuen Projekts zu steuern. Außerdem sollte das Suchverhalten analysiert werden: Wie gelangen Kunden in das Storefront? Sind es organische Suchergebnisse, direkte Zugriffe, Social Media oder E-Mails?

Ein weiterer Punkt ist das **Kaufmotiv**. Werden Produkte für den Eigengebrauch erworben? Oder werden Produkte häufig als Geschenk für Familie oder Freunde, für eigene Kinder oder Haustiere, oder sogar im Auftrag von Unternehmen im B2B-Kontext, gekauft? Besonders die **Bedürfnisse und Interessen** der Zielgruppe sollten hier berücksichtigt werden – etwa ein Fokus auf Nachhaltigkeit, Preisbewusstsein, Produktqualität, Kundenservice oder schnelle Lieferzeiten.

Gerne kann an dieser Stelle die Unterscheidung zwischen **Neukunden** und **Bestandskunden** erfolgen. Neukunden legen oft Wert auf eine gute UX und Preise, während Bestandskunden von personalisierten Angeboten und Treueprogrammen profitieren können.

Die Ergebnisse der Analyse sollten genutzt werden, um die **Zielgruppe zu segmentieren und Personas** zu erstellen – fiktive Profile, die typische Kunden repräsentieren. Jede Persona sollte konkrete Merkmale enthalten, wie Alter, Beruf, Interessen, bevorzugte Kanäle und Kaufmotive. Wichtig ist, die Personas auf realen

Daten zu entwerfen und sie regelmäßig zu aktualisieren, um Veränderungen in der Zielgruppe zu berücksichtigen.

Durch eine umfassende Zielgruppenanalyse kann der E-Commerce Owner die Anforderungen an die neue E-Commerce Plattform schärfen und damit die Bedürfnisse der Kunden besser erfüllen.

✓ CHECKLISTE ZIELGRUPPENANALYSE

- Es wurden Tools für die Zielgruppenanalyse eingerichtet und diese sammeln Daten über relevante Merkmale der Nutzer.
- Die Nutzer werden in relevante Kundensegmente eingeordnet (bzw. es wurden Personas erstellt) und die Bedürfnisse sind identifiziert.

3.5 Business Model Canvas

Das Business Model Canvas (BMC) ist ein strategisches Werkzeug zur Strukturierung und Darstellung von Geschäftsmodellen. Im E-Commerce-Kontext kann das Business Model Canvas die Zusammenhänge zwischen den verschiedenen Bereichen eines E-Commerce Systems aufzeigen und Potenziale für Optimierungen sichtbar machen. Für ein Replatforming bietet es daher eine hervorragende Grundlage, um das bestehende Geschäftsmodell zu analysieren und die Anforderungen an das neue System zu definieren.

Das BMC besteht aus neun Bausteinen, die die zentralen Elemente eines Geschäftsmodells abdecken. Der erste Baustein sind die **Kundensegmente**. Diese definieren, für wen das E-Commerce System Werte schafft. Ein Online-Shop könnte unterschiedliche Kundengruppen bedienen: B2C-Kunden wie junge Erwachsene, die auf Nachhaltigkeit achten, oder B2B-Kunden, die wiederkehrende Bestellungen für ihr Unternehmen aufgeben. Das Verständnis der Kundensegmente ermöglicht es, die E-Commerce Plattform und die Marketingmaßnahmen gezielt auszurichten. Gerne können hier Informationen aus der Zielgruppenanalyse einfließen.

Auf Basis der Kundensegmente wird das **Wertangebot** (oder Alleinstellungsmerkmale) festgelegt. Im E-Commerce sind das beispielsweise schnelle Lieferzeiten, eine Abo-Funktion für automatisierten Bestellung oder exzellente Produktberatung. Besonders wichtig ist die Differenzierung: Warum sollten Kunden

bei diesem Online-Shop einkaufen und nicht bei der Konkurrenz? Ein Beispiel für ein starkes Wertangebot kann ein Drogeriemarkt sein, bei dem sich Nutzer per Live-Chat zu Produkten beraten lassen können (bspw. Windeln) und ein Abo abschließen können. Die Nutzer erhalten damit regelmäßig richtigen die Windeln und sparen sich das lästige Tragen.

Die **Kanäle**, über die Kunden erreicht werden, sind ein weiterer zentraler Aspekt. Dazu gehören nicht nur der eigene Online-Shop, Newsletter und mobile Apps, sondern auch indirekte Kanäle wie Amazon, eBay oder Social-Media-Plattformen. Durch die Analyse der Kanäle kann geprüft werden, welche Vertriebskanäle am effektivsten sind und ob Potenziale bestehen.

Im nächsten Baustein **Kundenbeziehungen** wird beschrieben, wie Kunden mit dem E-Commerce System interagieren. Diese können von automatisierten Prozessen wie Self-Service-Funktionen im Kundenbereich, einem Ticketing-Tool, bis hin zu persönlichem Support, per Live-Chat, reichen. Loyalty-Programme oder personalisierte Produktempfehlungen mittels E-Mails sind ebenfalls zentrale Elemente, um langfristige Kundenbindungen aufzubauen. An dieser Stelle können die Mitarbeiter aus dem Kundensupport wertvolle Einblicke liefern.

Ein E-Commerce-Geschäft ist ohne eine Analyse der **Einnahmequellen** unvollständig. Hier sollte beschrieben werden, ob die Einnahmen eher direkt aus dem E-Commerce System oder externen Quellen, wie z.B. Marktplätzen oder Preissuchmaschinen, stammen. Sofern zutreffend, kann hier die Verteilung zwischen B2C und B2B sowie zwischen online und stationären Käufen gemacht werden.

Für die Umsetzung des Geschäftsmodells sind **Schlüsselressourcen** entscheidend. Hier wird beschrieben, welche Ressourcen für ein erfolgreiches E-Commerce Projekt notwendig sind. Dazu gehören etwa das Hosting bzw. die Infrastruktur, der physische Lagerbestände und ein gut eingespieltes Team (Entwickler, Content-Manager, Marketer, Produktdatenmanager, E-Commerce Manager, Kundensupport, ...). Diese Ressourcen ermöglichen es, die Anforderungen der Kunden zu erfüllen und die Geschäftsziele zu erreichen.

Die **Schlüsselaktivitäten** umfassen die zentralen Aufgaben eines E-Commerce Projekts, um Kunden zu bekommen. Darunter zählen die Optimierung des Storefronts und der UX, die Pflege des Produktkatalogs und die Umsetzung von

Marketingmaßnahmen. Auch die Buchhaltung und die Logistik – vom Lagermanagement bis zur schnellen Lieferung – gehört zu diesen Kernaufgaben.

Die **Schlüsselpartnerschaften** unterstützen diese Aktivitäten und sorgen für das Alleinstellungsmerkmal des E-Commerce Systems. Dazu zählen Logistikdienstleister, Zahlungsanbieter oder Lieferanten, die für eine zuverlässige Warenverfügbarkeit sorgen. Ebenso sollten hier die involvierten externen Agenturen genannt sein.

Abschließend wird die **Kostenstruktur** analysiert. Sie setzt sich aus fixen Kosten, wie Lizenzen oder Hosting, und variablen Kosten, wie Werbung, PSP Transaktionskosten oder Versandkosten, zusammen. Die Kostenstruktur gibt Aufschluss darüber, wo Einsparpotenziale liegen und welche Investitionen notwendig sind.

Das Business Model Canvas ermöglicht es, das Geschäftsmodell eines E-Commerce-Systems in seiner Gesamtheit zu erfassen. Es zeigt nicht nur, welche Bereiche besonders gut funktionieren, sondern auch, wo Schwächen und Optimierungspotenziale liegen. Für ein Replatforming-Projekt schafft es damit eine fundierte Grundlage, um die Plattform nicht nur technisch, sondern auch strategisch zukunftssicher auszurichten.

✓ CHECKLISTE BUSINESS MODEL CANVAS

- Das Business Model Canvas wurde mit allen relevanten Stakeholdern erstellt. Dieses beinhaltet Informationen über:
  - Kundensegmente (Beschreibung der Kunden, Persona, Zielgruppe)
  - Wertangebot (Was ist der USP gegenüber Wettbewerbern?)
  - Kanäle (Über welche Kanäle werden Kunden auf das Angebot aufmerksam gemacht?)
  - Kundenbeziehungen (Wie oft und wie wird mit Kunden kommuniziert?)
  - Einnahmequellen (Wie kommen die E-Commerce relevanten Umsätze zu Stande?)
  - Schlüsselressourcen (Welche Ressourcen sind für einen erfolgreichen Betrieb notwendig?)
  - Schlüsselaktivitäten (Welche Aktivitäten sind für einen erfolgreichen Betrieb notwendig?)
  - Schlüsselpartnerschaften (Welche Partner sind für einen erfolgreichen Betrieb notwendig?)
  - Kostenstruktur (Wie kommen die E-Commerce relevanten Kosten zu Stande?)

3.6 Dos und Don'ts in der Analyse

Die Analysephase eines Replatforming-Projekts ist ein entscheidender Schritt, der bereits vor der Umsetzung über den Erfolg oder Misserfolg des Vorhabens bestimmen kann. Sie bietet die Grundlage, um den aktuellen Zustand des E-Commerce Systems zu verstehen und eine fundierte Entscheidungsbasis für die nächsten Schritte zu schaffen. Fehler in dieser Phase können schwerwiegende Auswirkungen für die weiteren Stadien des Projekts haben, weshalb es essenziell ist, sich an bewährten Praktiken zu orientieren.

Zu den wichtigsten Prinzipien gehört es, mit Daten und Fakten zu arbeiten und nicht auf Bauchgefühl zu vertrauen. Tools wie **Hotjar**, die Heatmaps und Sitzungsaufzeichnungen liefern, oder **Analytics** und Business-Intelligence-Systeme bieten wertvolle Einblicke in Nutzerverhalten, Conversion Rates und technische Schwachstellen. Diese objektiven Daten sind unverzichtbar, um die Zielgruppe sowie Stärken und Schwächen des aktuellen Systems realistisch einzuschätzen.

Ein weiterer zentraler Aspekt ist der **Umgang mit Stakeholdern**. Hier gilt es, die Bedeutung einzelner Gruppen weder zu über- noch zu unterschätzen. Während die Meinung von Führungskräften, wie CEOs, oft stark gewichtet wird, dürfen operative Stakeholder wie Mitarbeiter oder Lieferanten nicht vernachlässigt werden. Diese Gruppen bringen praxisnahe Perspektiven ein, die für die Analyse von Prozessen und Bedürfnissen essenziell sind. Vor allem sollten die Bedürfnisse der Kunden erfüllt werden, da diese für den Umsatz mit der E-Commerce Plattform sorgen. Die Erfüllung der Wünsche der Führungsebene zu Lasten der Kundenbedürfnisse wird zwar das Projekt auf kurze Sicht beruhigen, auf langer Sicht jedoch zu sub-optimaler Kundenzufriedenheit und damit ausbleibenden Umsätze führen.

Technische Anforderungen sollten in der Analysephase **keinesfalls unterschätzt werden**. Aspekte wie Pagespeed, Web Vitals und die technische Integration zwischen Systemen haben einen direkten Einfluss auf die Nutzererfahrung, Ladezeiten die SEO-Performance und letztlich auf den Umsatz. Optimierte Ladezeiten führen nicht nur zu einer besseren UX und damit zu einer höheren Conversion Rate. Sie beeinflussen ebenso die Core Web Vitals und können damit Sichtbarkeit des Online-Shops in Suchmaschinen verbessern, was wiederum zu höheren Besucherzahlen führt. In Summe führen also bessere Ladezeiten zu mehr Umsatz.

Die **Ist-Analyse** sollte zudem nicht nur oberflächlich betrachtet, sondern mit dem **nötigen Tiefgang erfolgen**. Besonders bei technischen Integrationen wie der Anbindung von PIM-, ERP- oder Logistiksystemen ist eine präzise Dokumentation notwendig. Wichtige Details, wie API-Spezifikationen, Felder in XML- oder CSV-Daten, Validierungsmechanismen, und die dahinterliegende Geschäftslogik dürfen nicht ausgelassen werden. Diese Informationen sind essenziell, um spätere Implementierungsprobleme zu vermeiden.

Ein weiteres Do besteht darin, das **bestehende System aus funktioneller Sicht kritisch zu hinterfragen**. Dabei sollte geprüft werden, ob alle vorhandenen Features und Integrationen tatsächlich genutzt werden und welchen Kosten-Nutzen-Effekt sie haben. Auch der Wartungsaufwand der existierenden Funktionen sowie mögliche Alternativen sollten betrachtet werden: gibt es für die bestehende Funktion eine bessere oder günstigere Option? Oft wächst ein E-Commerce System über Jahre hinweg, durchlebt ggf. mehrere Mitarbeiterwechsel, und damit werden einige Funktionen als “gegeben” gesehen. Ein Sparring mit externen Experten, wie Beratern oder anderen E-Commerce-Managern, kann hier neue Perspektiven eröffnen und Optimierungspotenziale aufzeigen.

KAPITEL 4

KONZEPTION UND UMSETZUNG

Die Konzeptions- und Umsetzungsphase ist das operative Herzstück eines E-Commerce-Replatforming-Projekts. Hier werden die zuvor definierten Ziele, Anforderungen und Analysen in konkrete Konzepte und technische Lösungen übersetzt. Dabei wird nicht nur der Projektmanagement-Ansatz gewählt, sondern auch die praktische Umsetzung mit klar definierten Verantwortlichkeiten und Prozessen geplant. Ein zentraler Bestandteil dieser Phase ist die Erstellung eines Product Backlogs, das alle benötigten Funktionen und Features priorisiert, sowie einer realistischen Projekt-Timeline, die mit den relevanten Akteuren abgestimmt ist.

Die Gestaltung des Storefronts spielt eine entscheidende Rolle, da sie das Kundenerlebnis maßgeblich beeinflusst. Usability und visuelle Klarheit stehen dabei im Fokus. Gleichzeitig wird das Integrationskonzept entwickelt, das sicherstellt, dass alle angebundenen Systeme – von ERP und PIM bis hin zu Payment Service Providern – nahtlos miteinander kommunizieren.

Ein weiterer Kernbereich ist das Migrationskonzept, das detailliert beschreibt, wie Daten sicher vom alten auf das neue System übertragen werden. Ergänzend dazu muss die Infrastruktur aufgesetzt werden, einschließlich Hosting, Deployment-Workflows und Testumgebungen.

Die Phase endet mit der Einrichtung eines QA-Prozesses, der sicherstellt, dass das neue System technisch stabil und benutzerfreundlich ist. Abschließend werden die wichtigsten Dos und Don'ts beleuchtet, um typische Fehler in der Konzeption und Umsetzung zu vermeiden. Diese Phase ist entscheidend, um sicherzustellen, dass die Vision des neuen E-Commerce-Systems nicht nur theoretisch, sondern auch praktisch erfolgreich umgesetzt wird.

4.1 Projektmanagement

In diesem Kapitel wird das Projektmanagement des E-Commerce-Replatformings beschrieben. Es werden verschiedene Aspekte beleuchtet – von der Wahl der richtigen Methodik, wie SCRUM oder Wasserfall, bis hin zur praktischen Anwendung agiler Ansätze.

Besonderes Augenmerk wird auf die Rolle des Product Owners, die Definition eines klaren Product Backlogs und der Erstellung einer realistischen Projekt-Timeline gelegt. Darüber hinaus wird Change Management als integraler Bestandteil betrachtet, um sicherzustellen, dass technische und organisatorische Veränderungen nachhaltig umgesetzt werden. Ziel ist es, dem E-Commerce Owner konkrete Werkzeuge und Strategien an die Hand zu geben, um das Projekt erfolgreich zu steuern.

SCRUM vs. Wasserfall

In diesem Abschnitt werden bei beiden gängigsten Projektmanagementansätze miteinander vergleichen: SCRUM und Wasserfall. **SCRUM** ist eine agile Projektmanagementmethode, die iterative und inkrementelle Ansätze verwendet. Projekte werden in sogenannte **Sprints** unterteilt, die typischerweise ein bis vier Wochen dauern. Am Ende jedes Sprints steht ein funktionsfähiges Produktinkrement. SCRUM zeichnet sich durch regelmäßige Meetings und eine enge Zusammenarbeit zwischen allen Beteiligten aus. Das Projekt wird durch ein klar priorisiertes **Product Backlog** gesteuert, in dem alle Aufgaben in Form von Epics und Stories enthalten sind. Die Rollen in SCRUM sind klar definiert: Es gibt einen **Product Owner**, der die Anforderungen priorisiert, ein **SCRUM-Team**, das die Umsetzung übernimmt, und einen **SCRUM Master**, der für die Prozessdisziplin sorgt. In SCRUM gibt es feste Zeremonien, die den Ablauf eines Projekts strukturieren. Diese helfen, eine klare Kommunikation zwischen Teammitgliedern und Stakeholdern sicherzustellen und kontinuierlich auf die Projektziele hinzuarbeiten:

- Sprint Planning: Zu Beginn jedes Sprints wird entschieden, welche Aufgaben aus dem Product Backlog bearbeitet werden. Der Product Owner nennt seine gewünschten Aufgaben, diese werden vom SCRUM Team eingeschätzt und geplant.
- Daily Scrum: Ein kurzes, tägliches Meeting, bei dem jedes Teammitglied über Fortschritte, Hindernisse und Pläne spricht. Im E-Commerce-Projekt kann dies

helfen, frühzeitig Blocker zu identifizieren, etwa Verzögerungen bei der PIM-Integration.

- Sprint Review: Am Ende eines Sprints werden die umgesetzten Aufgaben vorgestellt. Stakeholder und Team prüfen gemeinsam, ob die Ergebnisse den Anforderungen entsprechen.
- Sprint Retrospective: Nach jedem Sprint reflektiert das Team, was gut lief und wo Verbesserungen möglich sind, um zukünftige Sprints effizienter zu gestalten.

Was sind jeweils die Vor- und Nachteile von SCRUM?

SCRUM bietet eine gewisse Flexibilität und ermöglicht eine schnelle Anpassung an Änderungen, da in jedem Sprint neue Prioritäten gesetzt werden können. Die regelmäßigen Meetings und Reviews fördern die Kommunikation und schaffen Klarheit über den Projektstatus, was insgesamt die Transparenz verbessert. Dieser kontinuierliche Austausch mit Stakeholdern sorgt dafür, dass das Endprodukt besser auf deren Bedürfnisse zugeschnitten ist. Das Risiko wird im SCRUM minimiert, indem Probleme frühzeitig erkannt und innerhalb des nächsten Sprints adressiert werden können.

Die regelmäßigen Meetings können jedoch zeitintensiv sein, womit ein gewisser Kommunikationsaufwand einhergeht. Zudem wird bei jeder praktischen Anwendung von SCRUM ein gewisser Workflow im Tooling, z.B. JIRA, Trello, Asana, eingesetzt und damit können bei Stakeholdern und Teammitgliedern ohne SCRUM Erfahrung Missverständnisse entstehen. Da SCRUM auf Flexibilität setzt, können feste Zeitpläne teilweise schwer durchsetzbar sein. Unerwartete Change Requests oder Feedbacks können das ursprünglich geplante Product Backlog aufblähen, was zu Unsicherheiten in Deadlines führen kann.

Der **Wasserfallansatz** ist ein traditionelles Projektmanagementmodell, bei dem das Projekt in sequenzielle Phasen unterteilt wird, die nacheinander abgeschlossen werden. Die typischen Phasen sind **Anforderungsanalyse, Design, Implementierung, Test, Deployment und Wartung**. In der Regel muss jede Phase abgeschlossen sein, bevor die nächste beginnen kann. Es gibt wenig Spielraum für Anpassungen, sobald die Anforderungen definiert sind. Die Durchführung der Prozesse und die Implementierung der Anforderungen sind die Verantwortung des Projektleiters. Im Wasserfall übernimmt dieser also die Planung, Koordination, Kommunikation, Überwachung und Abnahme.

Was sind jeweils die Vor- und Nachteile von Wasserfall?

Die Struktur ist einfach zu verstehen, und der Fortschritt lässt sich leicht nachverfolgen. Es gibt in der Regel weniger Meetings oder Workflows, die vom ganzen Team zu beachten sind. Die Anforderungen sind zum Projektstart fix und klar definiert, sodass das ganze Team eine gute Orientierung für das Ziel hat. Aufgrund dessen sind die Deadlines und Budgets sind von Anfang an fixiert und überschaubar, und damit ist die Planbarkeit besser.

Auf der anderen Seite führt der fixe Projektplan zu einer gewissen Unflexibilität. Falls Änderungen nach Beginn der Implementierung auftreten, können diese schwer umzusetzen sein. Zudem gibt es gewisse Risiken, da ein anwendbares Produkt erst später im Projektverlauf geliefert wird. Probleme oder Missverständnisse können somit erst spät erkannt werden, was zu erheblichen Verzögerungen und Nachbudgetierungen führen kann. Da in der Regel Meetings nur bei Bedarf durchgeführt werden und Anpassungen erst nach Abschluss einer Phase vorgenommen werden, entstehen im Zweifel längere Feedbackzyklen.

Zusammenfassend eignet sich SCRUM besonders für dynamische Projekte, bei denen Flexibilität und kontinuierliches Feedback erforderlich sind. Wasserfall ist hingegen ideal für Projekte mit klar definierten Anforderungen, stabilen Rahmenbedingungen und einer festen Zielvorgabe. Die Implementierung einer E-Commerce Plattform ist in der Praxis aufgrund der folgenden Faktoren eher ein dynamisches Vorhaben:

- Änderungen in der Gesetzeslage, wie z.B. die DSGVO, Accessibility Richtlinie, Produktsicherheitsverordnung usw. führen zu neuen Anforderungen
- Einbindung von externen Systemen, wie z.B. einem PIM, PSP oder ERP, sowie Extensions, die jeweils durch Updates neue oder geänderte Funktionen bzw. Schnittstellen liefern. Es entstehen Change Requests
- Mehrere Teams arbeiten am Projekt mit, wie z.B. Design, Marketing, Produktdatenmanagement. Es kommt somit schnell zu Änderungswünschen oder Anfragen zu neuen Funktionen.

Im weiteren Verlauf des Buches wird daher vom SCRUM Ansatz beim Replatforming Projekt ausgegangen.

SCRUM in der Praxis

SCRUM basiert auf klar definierten Zeremonien, die Struktur und Transparenz in den Entwicklungsprozess bringen. Diese Zeremonien sind darauf ausgelegt, das Team zu koordinieren, den Fortschritt zu überprüfen und Hindernisse zu beseitigen. Es gibt fünf Hauptzeremonien: Sprint Planning, Daily Stand-up, Sprint-Review, Sprint Retrospective und das Backlog Refinement. Das SCRUM-Team, welches für die Umsetzung des Projektes zuständig ist, umfasst in der Regel 3 bis 9 Mitglieder. Im E-Commerce Kontext sind es bspw. Frontend- und Backendentwickler, Designer, QA-Ingenieure und Systemadministratoren bzw. DevOps.

Zu Beginn eines jeden Sprints treffen sich das SCRUM-Team, der Product Owner und der SCRUM Master im **Sprint Planning**, um die Arbeit für den kommenden Sprint zu planen. Der Product Owner präsentiert priorisierte Backlog-Items (Tickets), und das Team wählt die Aufgaben aus, die es basierend auf ihrer Kapazität und den Sprintzielen umsetzen kann. Im besten Fall sind die zu planenden Tickets bereits vor dem Planning vom Team gesichtet und geschätzt, entweder eigenständig von jedem Mitglied oder in einem Backlog Refinement (mehr dazu weiter unten). Das Ergebnis ist ein Sprint Backlog mit klaren Aufgaben, die während des Sprints abgeschlossen werden sollen. Vor dem ersten Sprint empfiehlt sich ein sogenannter Sprint 0. Hier wird noch keine Funktion geliefert, sondern das Projekt vorbereitet, wie etwa

- Einrichtung der Infrastruktur, Code Repository und Build & Deployment Prozessen
- Konzeption von Funktionen, Schnittstellen und des Storefronts
- Einrichtung des richtigen Toolings für Designer, Entwickler und Tester
- Erstellung Product Backlogs inkl. Priorisierung

Das **Daily Stand-up** ist ein tägliches Meeting und dauert max. 15 Minuten. Es dient dazu, den Fortschritt zu synchronisieren und mögliche Blockaden zu identifizieren oder notwendige Entscheidungen zu treffen. Als Orientierung für den Ablauf können die folgenden Fragen und Antworten eines jeden Teilnehmers (SCRUM-Team und SCRUM Master) dienen: *Was habe ich gestern erreicht? Was plane ich heute? Welche Hindernisse gibt es?* In der Theorie nimmt der Product Owner optional teil, meist als Zuhörer. In der Praxis bewährt es sich jedoch, den Product Owner als festen Bestandteil des Daily Meetings zu installieren, damit Entscheidungen schneller getroffen werden können. Falls im Daily Stand-up technische oder ungeklärte Fragen bestehen, können diese von den betroffenen Personen nach dem Meeting separat in kleiner Runde geklärt werden.

Am Ende des Sprints präsentiert das Team im **Sprint-Review,** dem Product Owner und anderen Stakeholdern, z.B. Content-Managern, ERP Dienstleister, Marketer, die abgeschlossenen Arbeiten. Ziel ist es, die umgesetzten Tickets vorzustellen und Feedback einzuholen, das für künftige Iterationen genutzt wird. Der E-Commerce Owner kann in diesem Meeting feststellen, ob die Umsetzung in die gewünschte Richtung geht und ob die Anforderungen an die neue E-Commerce Plattform erfüllt werden. Damit ist das Sprint-Review vermutlich die wichtigste SCRUM Zeremonie und dementsprechend sollte sich das Entwicklungsteam und der Product Owner vorbereiten.

Die **Sprint Retrospective** richtet sich auf die Prozesse und das Team selbst. Sie findet ebenfalls am Ende des Sprints statt und wird vom SCRUM Master moderiert. Gemeinsam werden Herausforderungen, Erfolge und Verbesserungspotenziale diskutiert. Ziel ist es, konkrete Maßnahmen zur Optimierung der Zusammenarbeit und Arbeitsweise zu definieren. Dies können etwa Verbesserungen in den Workflows der Projektmanagement-Software, Zusammenarbeit mit anderen Teams (Design, Marketing, ...) oder dem Build & Deployment-Prozess sein.

Optional und auf Anfrage findet das **Backlog Refinement** statt. Der Product Owner und das Entwicklungsteam arbeiten gemeinsam daran, das Product Backlog zu aktualisieren und zu priorisieren. Aufgaben werden detailliert beschrieben und geschätzt, sodass das Team während des Sprint Plannings gut vorbereitete Tickets hat.

Alle Zeremonien folgen einem strukturierten Ablauf und finden in regelmäßigen Abständen statt, womit der Aufwand für Koordination und Kommunikation reduziert wird. Der SCRUM Master sorgt dafür, dass die Meetings zielgerichtet bleiben und zu regelmäßigen Zeiten stattfinden, während der Product Owner den geschäftlichen Fokus einbringt. Das SCRUM-Team ist in jeder Zeremonie zentral, da es letztendlich die Verantwortung für die Umsetzung der Arbeit trägt. Das Ergebnis dieser Zeremonien ist ein kontinuierlich verbessertes Produkt, eine transparente Kommunikation und ein adaptiver Entwicklungsprozess.

Ein Sprint dauert in der Regel zwischen einer Woche und einem Monat. Diese "Timebox" des Sprints, bleibt jedoch immer identisch, um die Komplexität und den Koordinationsaufwand der einzelnen Teilnehmer zu reduzieren. Kürzere Sprints führen zu einem gewissen Overhead an Zeremonien, da sich das Team jede Woche auf das Review und Planning vorbereiten sollte. Längere Sprints erhöhen hingegen das Geschäftsrisiko, da die umgesetzten Tickets erst nach längerer Zeit präsentiert und angepasst werden können.

Product Owner bestimmen

Falls ein externer Partner für die Umsetzung des Projekts engagiert wird, sollte vorab entschieden werden, wer die Rolle des Product Owners (PO) übernimmt: der Auftraggeber (E-Commerce Owner) oder die Agentur.

Falls der Auftraggeber den PO stellt, arbeitet dieser eng mit dem Entwicklungsteam der Agentur zusammen, um die Anforderungen zu priorisieren und Entscheidungen zu treffen. Vorteilhaft in dieser Konstellation sind die direkten Geschäftskenntnisse. Der Auftraggeber hat ein tiefes Verständnis für sein Geschäft, die Zielgruppe und die Marktanforderungen. Entscheidungen können schnell getroffen werden und basieren daher auf fundierten Kenntnissen. Damit kann der Auftraggeber strategische Prioritäten direkt und klar definieren. Zudem ist der Auftraggeber mit der Vision bestens vertraut und er fühlt sich als Mitarbeiter des E-Commerce Owners stark verantwortlich.

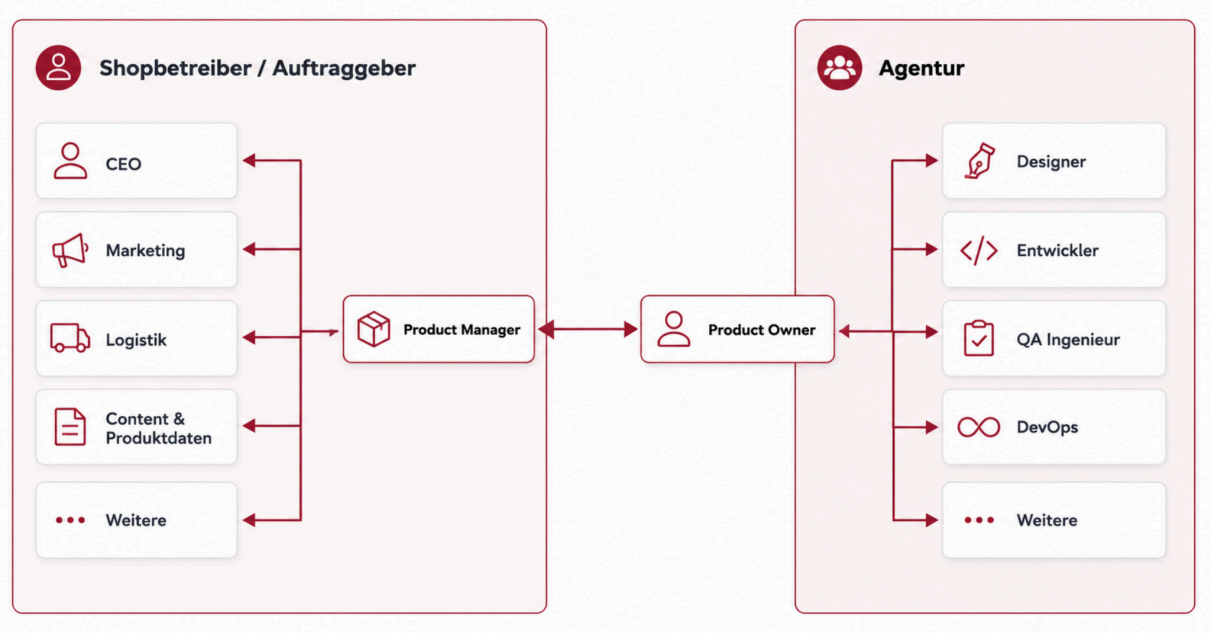
Allerdings haben die Auftraggeber oft weniger Erfahrung in der agilen Arbeitsweise oder der Projektmanagement-Software, was zu Verzögerungen oder Missverständnissen führen kann und damit das SCRUM-Team frustriert. Zudem muss der Auftraggeber zur Erfüllung der PO-Rolle Zeit und Ressourcen bereitstellen, was im Tagesgeschäft oft schwierig ist. Schließlich fehlt oft das technische Verständnis des

neuen E-Commerce Systems und daher können technische Zusammenhänge oft nicht ausreichend bewerten werden, was die Kommunikation mit dem Entwicklungsteam erschweren kann.

Falls die Agentur den PO stellt, nimmt dieser die Anforderungen des Auftraggebers auf und leitet das Projekt. Er arbeitet also direkt mit seinem eigenen Team zusammen. In dieser Konstellation bringt der Agentur-PO die gesamte Erfahrung und Expertise seiner Agentur und aller bisherigen Projekte ein. Er verfügt meistens umfangreiche Kenntnisse in agilen Projekten und im E-Commerce Umfeld, was die Effizienz und Qualität der Umsetzung steigert. Damit kann der Agentur-PO technische Anforderungen und Business-Ziele besser verbinden. Zu Anfang des Projekts wird er seine Kapazitäten sicherstellen und sorgt damit für eine kontinuierliche Verfügbarkeit. Es gibt es weniger Verzögerungen durch andere Verpflichtungen. Schließlich hat jede erfahrene Agentur aufgrund der vorherigen Projekte bereits standardisierte Prozesse und Tools im Einsatz, die die Projektsteuerung erleichtern. Im besten Fall hat der Agentur-PO in vorherigen Projekten mit den gleichen Entwicklern gearbeitet, sodass eine gewisse Chemie und Automatismen, ähnlich wie im Teamsport, entstehen.

Jedoch ist ein Agentur-PO oft weiter weg vom Geschäft der E-Commerce Owners, hat damit weniger tiefgehende Kenntnisse über die Unternehmensziele, die Marktstrategie und die spezifischen Prozesse außerhalb des E-Commerce Systems. Zudem entsteht eine gewisse Kommunikationslast, da der Agentur-PO eng mit dem E-Commerce Owner zur korrekten Umsetzung der Vision zusammenarbeiten muss. Schließlich gibt der E-Commerce Owner einen Teil der Verantwortung ab, was zu einem Gefühl des Kontrollverlusts führen kann.

Mit dem Konzept des **kundenseitigen Product Managers** vereint man die Vorteile beider Ansätze. Der Auftraggeber stellt in dieser Konstellation den Product Manager, der die strategische Planung sicherstellt und die Produktvision sowie Strategie kommuniziert. Der PO wird von der Agentur gestellt, der diese Vision in priorisierte Backlog-Items übersetzt und die Umsetzung leitet. Der E-Commerce Owner übernimmt als Product Manager weiterhin die volle Verantwortung, während die Agentur eher in operativen Aufgaben und der korrekten Umsetzung engagiert sind. Dies ist ein Mittelweg, der sich in der Praxis bewährt hat.



Product Backlog definieren

Das Product Backlog ist das zentrale Werkzeug im Scrum-Prozess. Es enthält alle Anforderungen, Funktionen und Aufgaben, die für das Projekt wichtig sind, und bildet die Grundlage für die Arbeit des Entwicklerteams. Um das Backlog effektiv zu gestalten, wird es typischerweise in drei Ebenen unterteilt: Epics, User Stories und Subtasks.

Epics sind große, übergeordnete Anforderungen, die oft in mehrere User Stories zerlegt werden. Ein Beispiel wäre der Checkout-Prozess. Diese Epic könnte in **User Stories** wie „Als Nutzer möchte ich eine einfache Gastbestellung durchführen können“ oder „Als Nutzer möchte ich verschiedene Zahlungsmethoden (Kreditkarte, Paypal, Sofortüberweisung) auswählen können“ zerlegt werden. **Subtasks** gehen noch tiefer und umfassen konkrete technischen Arbeitsschritte wie „Integration der API für Kreditkartenzahlungen“ oder „Anpassung des Frontends für das Kreditkarten-Formular“. In der Praxis werden Epics und User Stories vom Product Owner verfasst, während das Entwicklerteam die dazugehörigen Subtasks erstellt.

Ein weiterer wichtiger Aspekt ist die **Priorisierung der Backlog-Einträge**. Häufig wird hierfür der **Business Value** als Kriterium genutzt, um sicherzustellen, dass die wichtigsten und wertschöpfenden Anforderungen zuerst umgesetzt werden. Ein Beispiel: In einem Replatforming-Projekt könnte ein hoher Business Value für die Optimierung der Seitenladezeit, z.B. „Verbesserung der Core Web Vitals“, vergeben werden, da dies sowohl die Nutzererfahrung als auch die Conversion-Rate deutlich steigert. Weniger kritische Aufgaben wie „Anpassung der Newsletter-Templates“

könnten eine niedrigere Priorität erhalten. Aufgaben mit einem hohen Business Value zeichnen sich entweder durch eine signifikante Umsatzsteigerung (bessere Ladezeiten, bessere UX, bessere SEO Performance, ...) oder Kostenersparnis (Content-Seiten können eigenständig ohne Entwicklerkenntnisse responsive erstellt werden, Produktdaten werden aus dem PIM importiert, Bestellungen werden an das ERP exportiert, ...) aus. Eine Kostenersparnis beruht oft auf dem Wegfallen manueller Arbeitsschritte, die teilweise sogar zu menschlichen Fehlern führen können.

Bei der Priorisierung ist es extrem sinnvoll, die Tickets in Must-haves, Should-haves und Nice-to-haves zu kategorisieren. Man stellt damit sicher, dass alle Stakeholder die Prioritäten transparent nachvollziehen können. **Must-haves** sind essenzielle Funktionen, ohne die das Projektziel nicht erreicht werden kann. Sie stellen die minimale Basis dar, die das Produkt funktionsfähig machen und haben einen dementsprechend hohen Business Value. Dazu zählen etwa ein funktionierender Warenkorb, ein Checkout-Prozess und Zahlungsmethoden. Ohne diese Features wäre das E-Commerce System unbrauchbar. **Should-haves** sind sehr wichtige Funktionen, aber nicht unmittelbar notwendig für die Grundfunktionalität. Sie steigern die Nutzererfahrung oder Effizienz, und sollten aufgrund des hohen Business Values so bald wie möglich integriert werden. Dazu gehören etwa Filteroptionen im Produktkatalog oder eine Wunschliste. Das E-Commerce System funktioniert auch ohne sie, aber sie verbessern die Usability. **Nice-to-haves** sind optionale Funktionen, die keinen direkten Einfluss auf die Kernfunktionalität haben, aber zusätzlichen Mehrwert schaffen oder die Attraktivität erhöhen. Sie werden meist erst umgesetzt, wenn Ressourcen übrig sind, oder die Aufgaben mit geringem Aufwand umzusetzen sind. Dies sind bspw. animierte Produktpräsentationen oder personalisierte Produktempfehlungen.

Vor dem Projektstart sollte man mit allen Stakeholdern das **Minimum Viable Product (MVP)** definieren. Das MVP beschreibt alle Funktionen, für den ersten Livegang der neuen Plattform implementiert sein müssen. Dabei stehen E-Commerce Owner oft vor der Entscheidung, ob sie ein großes oder ein schlankes MVP wählen.

- Ein **großes MVP** enthält umfangreiche Features und bietet eine vollständigere Produktversion. Damit sind alle wichtige Funktionen enthalten und die Bedürfnisse aller Stakeholder sind eher erfüllt. Allerdings verlängert ein großes MVP die Entwicklungszeit und birgt die Gefahr, Ressourcen auf Funktionen zu verschwenden, die später kaum genutzt werden. Aufgrund der größeren Anzahl an

Features steigt die Komplexität rapide, sodass mehr Ressourcen für das Testing genutzt werden müssen. Das gesamte Projektrisiko steigt damit.

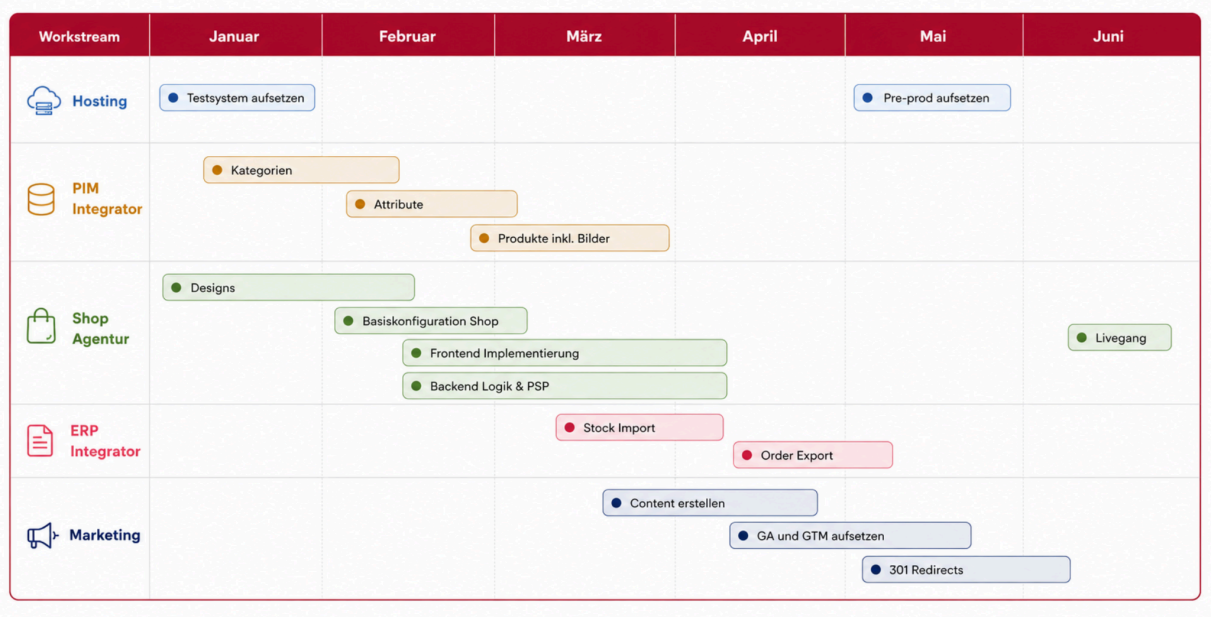
- Ein **schlankes MVP** konzentriert sich hingegen auf die minimal notwendigen Funktionen, die den Hauptnutzen des Produkts abdecken. Der Vorteil ist eine schnellere Markteinführung und die Möglichkeit, frühzeitig Feedback von echten Nutzern zu sammeln. Aufgrund des kleineren Funktionsumfangs ist das gesamte Projektrisiko und die Komplexität kleiner. Nachteile eines schlanken MVPs können jedoch eine eingeschränkte Benutzererfahrung und fehlende Funktionen sein, die für das Wachstum des Produkts wichtig wären.

Projekt Timeline

Neben dem Product Backlog hilft die Erstellung einer Projekt-Timeline dabei, dass alle beteiligten Akteure effizient zusammenarbeiten und das Projektziel termingerecht erreicht wird. Eine gut abgestimmte Timeline dient als zentrale Orientierungshilfe für alle Akteure des Projekts und schafft Klarheit über Aufgaben, Zuständigkeiten und Deadlines.

Bei der Planung der Timeline sollten zunächst die **Abhängigkeiten zwischen den Aufgaben der einzelnen Akteure** identifiziert werden. Beispielsweise müssen Produkte, Kategorien und Attribute im E-Commerce System sein, damit die technische Entwicklung, z.B. das Storefront, abgeschlossen wird. Diese wiederum sind notwendig, bevor Inhalte vom Content-Manager eingepflegt werden können. Ebenso sollten die Integrationen von ERP-, PIM- und Payment-Systemen frühzeitig erfolgen, da sie entscheidend für den Betrieb des E-Commerce Systems sind.

Die **Einbindung aller relevanten Akteure** ist essenziell, darunter Shop-Entwickler, ERP- und PIM-Integrator, Content-Manager, Online-Marketing-Teams und andere Stakeholder. Jeder Akteur sollte die geplanten Meilensteine und Deadlines kennen, um mögliche Konflikte oder Verzögerungen kommunizieren und zu vermeiden. Regelmäßige Synchronisationsmeetings, z.B. im Sprint-Review helfen, den Fortschritt zu überprüfen und die Timeline bei Bedarf anzupassen.



Ein weiterer wichtiger Punkt ist die Definition von **Pufferzeiten**. Komplexe IT Projekte mit Integrationen externer Systeme verlaufen selten ohne unvorhergesehene Herausforderungen, daher sollte die Timeline realistisch geplant und mögliche Verzögerungen eingeplant werden. Die Projekt-Timeline flexibel genug sein, um auf unvorhergesehene Änderungen zu reagieren, aber zugleich klare Struktur und Verbindlichkeit schaffen, damit das Replatforming erfolgreich umgesetzt wird.

Außerdem müssen **kritische Meilensteine** wie die Fertigstellung des MVP, der Beginn von Tests, und der Go-Live-Termin klar definiert und kommuniziert werden. Vor allem der Go-Live-Termin sollte so gelegt werden, dass genügend Zeit für Tests und Fehlerbehebungen bleibt.

Change Management

Ein E-Commerce-Replatforming ist nicht nur ein technisches Projekt, sondern sollte stets auch als ein **Change-Management-Projekt** betrachtet werden. Die Einführung einer neuen Plattform betrifft nicht nur einzelnes System, sondern mehrere Systeme und Prozesse, Teams und sogar die Unternehmenskultur. Ein solcher Wandel erfordert eine klare Strategie, um sowohl technische als auch menschliche Herausforderungen erfolgreich zu bewältigen.

In der Praxis lassen sich vier wesentliche Change Management Aspekte identifizieren, die bei einem Replatforming berücksichtigt werden sollten.

1. Beschreibung und Bewertung der geplanten Veränderung:

Ein zentraler Schritt ist die klare Beschreibung der geplanten Änderungen. Dabei sollten die geplanten Neuerungen, wie z. B. das neue CMS, die neue PIM- und ERP-Integration, ein neuer PSP, umfassend und verständlich beschrieben werden. Ebenso wichtig ist es, die damit verbundenen Risiken zu identifizieren und hinsichtlich des Schwierigkeitsgrads einzustufen. Beispiele dafür sind: technische Risiken (z. B. Produkte aus dem PIM werden im Storefront falsch dargestellt, die E-Commerce Bestellungen können vom ERP nicht eingelesen werden, der Warenbestand kommt im E-Commerce System nicht richtig an, Transaktionen werden im PSP nicht richtig abgebucht), organisatorische Risiken (z. B. Widerstand der Mitarbeiter, da das neue CMS anders funktioniert) oder projektbezogene Risiken (z. B. Verzögerungen beim PIM Integrator behindert die Implementierung des Storefronts). Für jedes Risiko sollten Strategien entwickelt werden, um sie zu mitigieren und zu kontrollieren.

2. Einführung planen:

Die Einführung der neuen Plattform und Prozesse erfordert einen durchdachten Plan. Dieser sollte sowohl den Product Backlog als auch die Projekt-Timeline umfassen, um sicherzustellen, dass alle notwendigen Aufgaben und Abhängigkeiten berücksichtigt werden. Allerdings kann es notwendig sein, diesen Plan weiter zu konkretisieren, beispielsweise durch detaillierte Rollout-Pläne oder Meilenstein-Dokumente. Ein gut strukturierter Plan gibt allen relevanten Stakeholdern Orientierung und hilft, den Fortschritt zu verfolgen und mögliche Hindernisse frühzeitig zu erkennen.

3. Stakeholder einbeziehen:

Ein weiteres Kernelement ist die aktive Einbindung aller relevanten Stakeholder. Hier kann die Stakeholder-Analyse aus der Analysephase wertvolle Einblicke bieten. Die Bedürfnisse und Erwartungen von Stakeholdern wie dem Management, den operativen Teams (Marketing, IT, Logistik, usw.) und externen Partnern sollten in den Change-Management-Prozess einfließen. Es ist wichtig sicherzustellen, dass alle Stakeholder rechtzeitig informiert werden und ihre Anforderungen berücksichtigt sind, um Widerstände zu vermeiden und die Akzeptanz zu fördern. Workshops, regelmäßige Meetings und transparente Kommunikation sind hier essenziell, damit die Stakeholder die geplanten Änderungen annehmen und umsetzen.

4. Überprüfung und Erfolgsmessung:

Der letzte Schritt ist die Überprüfung, ob die geplanten Änderungen die gewünschten Ergebnisse aus den vorherigen Kapiteln (2.2. Ziele und Erwartungen definieren) liefern. Hier kommen Business Intelligence oder Analytics ins Spiel. Es sollten klare und messbare Ziele definiert werden, wie z. B. die Verbesserung der Conversion-Rate, die Verkürzung von Ladezeiten oder die Optimierung interner Prozesse. Mithilfe von KPIs kann gemessen werden, ob die Veränderungen wie geplant funktionieren und welchen Einfluss diese auf das Geschäft haben. Regelmäßige Überprüfungen und Feedbackschleifen helfen dabei, Anpassungen vorzunehmen und den Erfolg sicherzustellen.

Diese vier Aspekte – Beschreibung und Bewertung der Änderungen, Einführung planen, Stakeholder einbeziehen sowie Überprüfung und Erfolgsmessung – bilden eine solide Grundlage, um ein Replatforming erfolgreich als Change-Management-Projekt zu gestalten. Sie gewährleisten, dass technische Neuerungen nicht isoliert betrachtet werden, sondern im Einklang mit den Bedürfnissen der Organisation und der Mitarbeiter umgesetzt werden. Da der Product Manager in der Regel die zentrale Schnittstelle zwischen der neuen E-Commerce Plattform und den relevanten Stakeholdern ist, eignet sich dieser am besten als Treiber des Change Management Prozesses.

✓ CHECKLISTE PROJEKTMANAGEMENT

- Es wurde zwischen Wasserfall- und agilen Projektmanagement Methoden abgewägt und eine fundierte Entscheidung getroffen.
- Falls SCRUM als Projektmanagement Methode gewählt wurde, wird vor der eigentlichen Implementierung ein vorbereitender Sprint 0 abgehalten. Dabei wurde sichergestellt, dass alle involvierten Parteien die richtigen Zugänge und Toolings haben, um das Projekt erfolgreich umzusetzen.
- Die Infrastruktur inkl. Build & Deployment Prozess funktioniert.
- Alle Spezifikationen inkl. Schnittstellen abgenommen sind, und damit das Product Backlog entsteht.
- Alle Rollen im Projekt klar sind, insbesondere der Product Owner und Product Manager.
- Die SCRUM Zeremonien sind definiert und für alle Teilnehmer verständlich.
- Es entsteht eine Projekt-Timeline, die Abhängigkeiten berücksichtigt und von allen Projektbeteiligten akzeptiert wurde.

- Der Product Backlog initial erstellt wurde und eine klare Priorisiserung enthält. Der Product Backlog wird während der Umsetzung als lebender Organismus gesehen, der durchgehend mit weiteren Ideen und Funktionen befüllt wird.
- Das Replatforming wird als Change Management Prozess gesehen, d.h. alle relevanten Stakeholder werden in den Prozess einbezogen und die Bedürfnisse und die Ängste berücksichtigt.

4.2 Storefront Konzeption

Die Konzeption der Storefront bildet Grundlage für das Nutzererlebnis und die Conversion-Rate eines Online-Shops. Zunächst muss der **Umfang der zu erstellenden Designs und Konzepte** geklärt werden. Hierzu gehören die Definition der notwendigen Ansichten, wie Homepage, Kategorieseiten (POP) und Produktdetailseiten (PDP), Warenkorb, Check-out, sowie die zu berücksichtigenden Viewports (typischerweise Desktop, Tablet und Mobil). Es sollte außerdem festgelegt werden, ob interaktive Clickdummies oder Wireframes benötigt werden, oder ob statische Designs ausreichen. Bei umfangreichen Frontend-Implementierungen kann es hilfreich sein, die Funktionsweise zunächst anhand von Clickdummies und Wireframes abzubilden und anschließend diese in einer visuellen Ausgestaltung zu überführen.

Das **richtige Tooling** ist entscheidend, um eine effiziente Zusammenarbeit zu gewährleisten. Tools wie Figma oder Adobe XD[36] bieten etwa eine kontextbezogene Kommentarfunktion und ermöglichen somit eine enge Kollaboration zwischen Designern, Entwicklern und dem Product Owner. Diese Tools unterstützen auch den Abnahmeprozess, bei dem Entwickler die technische Machbarkeit prüfen und Stakeholder wie Product Owner bzw. Product Manager sicherstellen, dass die funktionalen Anforderungen erfüllt werden. Eine Kollaboration in PDF Dokumenten oder Bilddateien sollte vermieden werden, da dies ineffizient ist und man schnell den Durchblick verliert.

Die Designs selbst sollten die **Grundanforderungen an eine gute Usability** abdecken. Dazu gehören etwa

- möglichst wenige Klicks zur Nutzung einer bestimmten Funktion oder eines Prozesses, z.B. Größenauswahl bei der PDP oder reduzierte Anzahl der Felder im Check-out,
- eine intuitive Navigation, z.B. klare Trennung zwischen Kategorie Ebenen,
- eine leicht zu findende Suche mit Fehlertoleranz,
- eine nicht überladene Darstellung von Content, z.B. sollte man auf Image Teasern CTA Buttons finden und Text lesen können,
- eine visuelle Hervorhebung von wichtigen Elementen wie dem „Add-to-Cart-Button,
- Nutzung von gängigen Icons, z.B. der Lupe für die Suche, und Formen z.B. Fortschrittsanzeigen im Checkout-Prozess.

Bei mobilen Nutzern, die oft einen Großteil der Zielgruppe ausmachen, ist eine **mobile-first-Strategie** sinnvoll. In bestimmten B2B-Projekten kann jedoch die Desktop-Version im Vordergrund stehen.

Darüber hinaus müssen natürlich die funktionalen Anforderungen des neuen E-Commerce-Systems, die Zielgruppenanalyse und die strategischen Ziele berücksichtigt werden. Wenn das Ziel beispielsweise Kostenersparnis lautet, könnte ein Design entworfen werden, das sich nah am nativen Frontend der Plattform orientiert. Falls Performance und Ladezeiten verbessert werden sollen, sind ein schlankes Frontend mit minimalem JavaScript-Einsatz eher passend.

Die Einbindung von Best Practices aus **Baymard-Studien oder Blogs**[37] helfen, bewährte UX-Leitsätze umzusetzen. Ebenso ist die **Auswertung von Nutzerdaten** aus Tools wie Hotjar oder Google Analytics essenziell, um Designentscheidungen auf Basis realer Nutzerbedürfnisse zu treffen.

Zukünftige Anforderungen wie die **Accessibility-Richtlinie**[38] sollten ebenfalls beachtet werden, um eine barrierefreie Nutzung für alle Nutzergruppen sicherzustellen.

Wie man anhand der obigen Punkte sieht, erfordert die Storefront Konzeption Ressourcen, Erfahrung und Expertise. Daher wird in der Praxis oftmals ein entsprechender externer Partner für die Erstellung der Designs des neuen E-Commerce Projekts engagiert.

✓ CHECKLISTE STOREFRONT KONZEPTION

- Es wurde definiert, wer für die Erstellung der Storefront Konzepte zuständig ist.
- Der Umfang der zu liefernden Konzepte ist bekannt.
- Alle involvierten Parteien haben Zugang zu den richtigen Toolings.
- Die Designs wurden vom Entwickler Team hinsichtlich der technischen Machbarkeit freigegeben.
- Der Product Owner hat die Konzepte hinsichtlich den funktionalen Anforderungen freigegeben.
- Die Storefront Konzepte berücksichtigen UX Best Practices, fundierte Studien (z.B. Baymard) sowie Accessibility Richtlinien

4.3 Integrationskonzept

Ein durchdachtes **Integrationskonzept** ist essenziell für den Erfolg eines Replatformings, da es die Ziele und Erwartungen an die neue Plattform und die dahinterliegende Systemlandschaft abbildet. Das Konzept sollte auf den Erkenntnissen aus der IST-Analyse und Systemlandschaft aufbauen, um vorhandene Schwachstellen zu adressieren und das volle Potenzial der neuen Plattform auszuschöpfen. Da viele Aspekte bei der Analyse der Systemlandschaft, wie z.B. das Mapping, Timing, usw., im vorherigen Abschnitt beschrieben wurden, werden diese Aspekte hier bewusst kurz gehalten.

Zunächst sollte geklärt werden, welche externen Systeme in die neue E-Commerce Plattform integriert werden müssen. Typischerweise zählen dazu

- **Backend-Integrationen**: Das PIM (Produktdatenmanagement), das CRM (Kundenkonten und Kontaktanfragen), das ERP (Bestellungen, Bestände, Rechnung und Versandstatus), Marketing-Automation Tools (Newsletter) und der PSP für Zahlungsabwicklungen.
- **Frontend-Integrationen**: Hierzu gehören Tools wie der Google Tag Manager (GTM), Systeme für Personalisierung, Produktempfehlungen (Recommendation Engine) und die Suche. Besonders bei einer Headless-Architektur sind klare Schnittstellen zwischen Frontend und Backend erforderlich.

Ein weiterer wichtiger Aspekt ist das Timing der Integrationen:

- **Echtzeit-Integrationen**: Ideal für zeitkritische Prozesse wie Lagerbestände, bei denen zeitnahe Updates erforderlich sind.
- **Periodische Synchronisation**: Geeignet für weniger zeitkritische Daten wie Produktupdates oder Marketing-Listen. Dies kann Lastspitzen auf den Systemen reduzieren.

Die Technologie der Schnittstelle hat erheblichen Einfluss auf die Effizienz und Skalierbarkeit:

- **API-basierte Integration**: APIs ermöglichen eine schnelle, flexible und direkte Kommunikation zwischen Systemen. Sie erlauben Echtzeit-Updates, geringere Fehleranfälligkeit, und damit eine höhere Automatisierung. Allerdings entstehen teilweise ein höherer Entwicklungsaufwand und eine Abhängigkeit von API-Versionen.

- **Dateibasierte Integration** (z. B. CSV oder XML Austausch per SFTP): Wird oft für Systeme verwendet, die keine API Schnittstellen unterstützen. Derartige Integrationen sind in der Praxis erprobt und einfach umzusetzen, kostengünstig und geeignet für große Datenmengen in Batch-Verarbeitung. Jedoch gibt es aufgrund der Ablage per SFTP einen Zwischenschritt, der zur verzögerten Synchronisation und Fehlern führen (jeder zusätzliche Schritt in einem Prozess erhöht das Risiko von technischen Fehlern).

Eine weitere Entscheidung betrifft die Wahl der API-Endpoints: sollen die Shop-API- oder externe API-Endpoints, z.B. vom PIM oder ERP, genutzt werden?

- **Externe API-Endpoints**: Moderne PIM, ERP oder CRM Systeme bieten oft eigene APIs, die direkt vom E-Commerce System genutzt werden. In diesem Fall wird die Schnittstelle auf Seiten der E-Commerce Plattform programmiert, was Shop-Entwickler Ressourcen bindet. Damit hat das eigene Entwicklerteam, die Kontrolle über den Datenfluss zwischen E-Commerce System und dem externen System. Auf der anderen Seite entsteht eine Abhängigkeit von externen Dienstleistern und deren API Schnittstellenupdates.
- **Shop-eigene APIs**: Das E-Commerce System stellt eine API Schnittstelle für externe Systeme zur Verfügung. Die Schnittstelle wird in dem Fall auf der Seite vom PIM, ERP oder CRM programmiert (oder über eine Extension bzw. Connector geliefert), womit die Shop-Entwickler lediglich unterstützend und beratend zur Seite stehen. Die Entwicklung des E-Commerce Systems sowie die Integration können damit zeitlich parallel ablaufen, da unterschiedliche Teams daran arbeiten. Die API Endpoints von gängigen E-Commerce Systemen, wie Magento 2, Shopware 6 oder Shopify, sind umfangreich dokumentiert, in der Praxis erprobt, und dementsprechend ausgereift. Jedoch besteht das Risiko, dass der externe PIM-, ERP- oder CRM-Dienstleister mit den Shop API Endpoints weniger Erfahrung hat und es kann zu Verzögerungen kommen.

Wie bereits in der Analyse der Systemlandschaft beschrieben, ist ein zentraler Bestandteil des Integrationskonzepts das **Mapping der Attribute** zwischen den Systemen. Beispielsweise muss das Attribut „Produkt-ID“ im PIM eindeutig dem entsprechenden Feld im E-Commerce System zugeordnet werden. Zusätzlich sollten **Validierungen** implementiert werden, um eine Datenkommunikation entsprechend den Anforderungen beider Systeme sicherzustellen. Fehlerhafte Daten können zu Inkonsistenzen und potenziellen Ausfällen führen.

Das Integrationskonzept sollte von allen beteiligten Parteien – einschließlich Shop-Entwicklern, ERP- und PIM-Integratoren, sowie anderen relevanten

Stakeholdern – geprüft und freigegeben werden. Entscheidungen sowie Change Requests sollten hier transparent nachvollziehbar sein. Diese Freigabe stellt sicher, dass keine Missverständnisse bzw. Nachbudgetierungen entstehen und alle Systeme reibungslos miteinander kommunizieren können.

✓ CHECKLISTE INTEGRATIONSKONZEPT

- Es wurde eine Grafik der Systemlandschaft inkl. aller integrierten Systeme erstellt.
- Die gewünschten Schnittstellen sind aus technischer Sicht spezifiziert, darunter
  - Wird ein Austausch von Dateien stattfinden, oder wird es eine API-basierte Integration geben? Welche API Endpunkte werden genutzt (Shop-eigen oder extern)?
  - Wie funktioniert die Authentifizierung zwischen den Systemen?
  - In welchem Timing werden Daten ausgetauscht?
  - Welche Validierungen sind notwendig?
  - Es besteht ein Mapping der Attribute bzw. Felder zwischen den Systemen.
- Das Konzept wurde von allen involvierten Parteien freigegeben.

4.4 Migrationskonzept

Beim Migrationskonzept wird sichergestellt, dass alle notwendigen Daten vom alten (Quell-) in das neue (Ziel-) System übertragen werden. Ein gut geplantes Konzept berücksichtigt den **Umfang der Migration** und definiert, welche Datensätze migriert werden müssen. In der Praxis werden oft die folgenden Datensätze migriert:

- Inhalte wie Landing-Pages, Content- oder Blog-Seiten, inkl. den URL Keys, Metadaten und Assets.
- Katalogdaten wie Produktattribute, Kategorien, Produkte inkl. den Assets. Dies ist oft der Fall, falls der Katalog im Quellsystem und nicht in einem externen PIM gepflegt wurde. Falls der Katalog in einem PIM gepflegt wird, sollten die Daten aus diesem importiert werden. Sofern im Quellsystem Produktbewertungen enthalten sind, sollten diese ebenfalls migriert werden.
- Kundeninformationen, darunter Logins, Kundengruppen, Adressen, Wunschlisten und ggf. Treuepunkte.
- Bestellungen und darunter enthaltene Produkte, angewandte Rabatte oder Coupon Codes, genutzte Zahlungs- und Versandmethoden, Rechnungsdaten und Versandlinks. Falls Kunden migriert wurden, sollten die Bestellungen den entsprechenden Kundenkonten zugeordnet werden können. Die Bestellungen sind in diesem Fall nach der Migration im Kundenbereich im Zielsystem zu finden, was die UX verbessert.
- Preis- und Rabattregeln, darunter Coupon-Codes, Gültigkeitszeiträume, Restguthaben

Ein weiterer Schritt ist das **Mapping der Felder und des Datenmodells** zwischen dem alten und dem neuen System. Unterschiedliche Datenstrukturen müssen abgeglichen werden, bspw. wenn Adressen im alten System in zwei Felder für Straße und Hausnummer aufgeteilt sind, während das neue System ein gemeinsames Feld verwendet. Ebenso können Coupon-Codes im neuen System zusätzliche oder abweichende Parameter haben, wie Gültigkeitsbedingungen oder Rabattmodelle. Dies sollte im vorherigen Kapitel Funktionale Analyse Admin Bereich definiert worden sein.

Bei der **Datenbereinigung** können Altlasten im Zuge des Replatformings über Bord geworfen werden. Dabei sollten fehlerhafte oder doppelte Datensätze – etwa Duplikate bei Kunden oder nicht mehr genutzte Preisregeln – vor der Übertragung bereinigt werden, um die Datenqualität im neuen System zu verbessern.

Je nach Komplexität des Projekts muss die passende **Technologie für die Migration** gewählt werden. Fertige Migrationstools oder Extensions sind oftmals ausreichend für Standard-Shop-Systeme, wie Shopify, Magento 2 und Shopware 6, sofern keine umfangreichen Individualanpassungen in der Geschäftslogik vorhanden sind. Bei Quellsystemen mit umfangreichen Individualentwicklungen entstehen vom Standard abweichende Datenbank-Schemata und daher ist eine maßgeschneiderte Lösung erforderlich. Dabei werden die Datenbanken individuell kopiert und deren Inhalte im neuen System entsprechend gemappt. Für Inhalte wie Blog- oder CMS-Seiten kann eine manuelle Migration oft sinnvoller sein, da sich im Regelfall die Pagebuilder und die CMS-Komponenten zwischen den Quell- und Zielsystemen stark unterscheiden. Ein programmatischer Ansatz wäre in diesem Szenario eher zu aufwendig.

In jedem Fall sollten während der Implementierung **Testmigrationen** geplant werden, um die Datenübertragung zu prüfen und Fehler frühzeitig zu erkennen. Diese sollten idealerweise auf einer separaten Umgebung stattfinden. Dabei ist darauf zu achten, dass sensible Daten wie Namen oder E-Mail-Adressen während der Tests pseudonymisiert werden, um Datenschutzvorgaben einzuhalten. Zudem sollte sichergestellt werden, dass während der Testmigration keine ungewollten Prozesse ausgelöst werden, wie etwa das Versenden von E-Mails oder die Kommunikation mit dem Zahlungsdienstleister. Hierfür sollten Funktionen wie die E-Mail-Kommunikation oder PSP-Integrationen deaktiviert werden.

Nach einer erfolgreichen Testmigration stellt sich die Frage nach dem Vorgehen beim **Go-Live**. Eine Option ist, die gesamten Daten erneut inkl. den aktuellen Quell-Daten zu migrieren. Alternativ können bereits migrierte Daten in der (Ziel) Pre-Production-Umgebung belassen werden, und zum Zeitpunkt des Go-Lives werden nur mit den seit der Testmigration neu erstellten („Delta-Daten“) ergänzt werden. Diese Entscheidung hängt vom Geschäftsmodell und der Datenmenge ab.

Ein durchdachtes Migrationskonzept minimiert Risiken und stellt sicher, dass der Übergang in das neue System reibungslos verläuft. Es bildet somit die Grundlage für einen erfolgreichen Livegang der neuen E-Commerce-Plattform.

✓ CHECKLISTE MIGRATIONSKONZEPT

- Der Migrationsumfang wurde geklärt, wie etwa
  - CMS Seiten, Blog, FAQ, ...
  - Attribute, Produkte und Kategorien
  - Bestellungen, Rechnungen und Liefermeldungen (Tracking Links)
  - Kunden inkl. Logins, Adressen und Bestellungen
  - Preis- und Rabattregeln inkl. Gutscheindcodes
  - Es wurde ein Mapping der Felder zwischen Quell- und Zielsystem erstellt.
- Eine Testmigration wurde erfolgreich durchgeführt.
- Optional: Die Daten wurden bereinigt.

4.5 Infrastruktur aufsetzen

Das Aufsetzen der Infrastruktur ist ein kritischer Schritt bei der Umsetzung eines E-Commerce-Projekts, insbesondere wenn eine externe Agentur involviert ist und das neue System on-premise betrieben wird. In solchen Fällen ist es notwendig, eine durchdachte Infrastruktur und einen zuverlässigen Deployment-Prozess zu etablieren.

Ein strategisch wichtiger Aspekt ist die Entscheidung darüber, **wo das Hosting stattfinden soll**. Es gibt drei gängige Optionen, die jeweils Vor- und Nachteile mit sich bringen:

- **Entwickler-Agentur**: Die Entwickler-Agentur übernimmt das Hosting, was den Vorteil hat, dass sie die Umgebung perfekt an ihre eigenen Entwicklungsprozesse anpassen kann. Zudem gibt es nur einen Ansprechpartner für das E-Commerce System bei jeglichen Fragen oder Wünschen, womit weniger Kommunikations- und Koordinationsaufwand entsteht. Notfälle, wie z.B. Downtimes, werden damit in der Regel eher schneller behoben. Jedoch hat der E-Commerce Owner weniger Kontrolle über die Infrastruktur und bei einem späteren Agenturwechsel wird der Transfer aufwendiger sein.
- **Eigene IT des E-Commerce Owners**: Wenn der E-Commerce Owner über eine erfahrene IT-Abteilung verfügt, kann das Hosting intern durchgeführt werden. Dies bietet maximale Kontrolle und Flexibilität, erfordert aber auch ausreichend personelle und technische Ressourcen sowie Expertise im Betrieb und in der Wartung von Servern. Durch die Zusammenarbeit mit externen Entwicklern entstehen Kommunikations- und Koordinationsaufwände.
- **Spezialisierter E-Commerce-Hoster**: Anbieter wie Maxcluster[39] oder andere spezialisierte E-Commerce-Hoster bieten bspw. Managed-Hosting-Lösungen an, die auf Anforderungen gängiger Shop-Systeme optimiert sind. Diese Option kombiniert professionelle Betreuung mit einer Entlastung der internen IT, ist jedoch oft mit höheren laufenden Kosten verbunden. Zudem ist eine dritte Partei involviert, was zu Kommunikations- und Koordinationsaufwänden führt.

Ein weiterer wichtiger Punkt ist die **Einrichtung der notwendigen Umgebungen**: In der Praxis sind drei Umgebungen gängig: **Test** bzw. **Dev** (im weiteren nur “Dev”), **Stage** und **Prod** (Produktion). Die Stage-Umgebung dient als Zwischenstufe, in der Änderungen vor dem Deployment auf Prod getestet werden können. Daher sollte Stage hinsichtlich dem Content und dem Katalog synchron mit der Produktivumgebung sein. Der Vorteil ist, dass gravierende Bugs frühzeitig erkannt und

dadurch das Risiko von Fehlern in der Produktivumgebung minimiert werden. Der Nachteil ist jedoch, dass sie zusätzliche Entwickler- und Contentmanager-Ressourcen für die Integrationen und Pflege erfordert. Auch führt das Hosting einer Stage Umgebung natürlich zu Hostinggebühren. Falls der Content und die Katalogdaten auf der Dev Umgebung den Inhalten auf Produktiv ähneln, kann die Stage Umgebung wegfallen.

Für einen effizienten Entwicklungsprozess muss ein klar definierter **Deployment-Prozess** etabliert werden. Typischerweise werden Änderungen zunächst lokal entwickelt, dann in die Dev-Umgebung hochgeladen und dort getestet, bevor sie über die Stage-Umgebung in die Produktivumgebung (Prod) gelangen. Bevor Code auf eine Umgebung hochgeladen wird, finden in der Regel zunächst **automatisierte Tests** statt. Diese helfen dabei, die Qualität des Codes kontinuierlich sicherzustellen. Unit-Tests prüfen dabei einzelne Funktionen oder Module isoliert auf korrektes Verhalten, während Integrationstests das Zusammenspiel mehrerer Komponenten – etwa zwischen Frontend, Backend und Schnittstellen – validieren. Insbesondere bei größeren Projekten mit vielen Releases oder externen Partnern sorgen automatisierte Tests dafür, dass neue Änderungen keine unerwünschten Seiteneffekte verursachen. Sie können frühzeitig in den CI/CD-Prozess integriert werden und reduzieren manuelle Testaufwände erheblich. Richtig eingesetzt, steigern sie die Stabilität, Geschwindigkeit und Verlässlichkeit des gesamten Deployment-Prozesses.

Nach den automatisierten Tests, werden die einzelnen Tickets bzw. Funktionen werden oft auf Dev von QA Ingenieuren getestet (**manuelle Tests**). Sofern die Tests erfolgreich waren, wird das Ticket auf Stage eingespielt. Auf Stage finden in der Regel ganzheitliche Smoketests statt, die alle relevanten E-Commerce Funktionen, wie z.B. Suche, Warenkorb, Check-out, ganzheitlich und auf mehreren Endgeräten und Browsern prüfen. Mehr dazu im nächsten Abschnitt QA Prozess aufsetzen.

Auch die Frage, **wo der Code gehostet wird**, ist von Bedeutung. Ein Code-Repository beim E-Commerce Owner bietet maximale Kontrolle, während ein Repository bei der Agentur eine engere Integration in deren Arbeitsabläufe ermöglicht. In der Regel ist das Code Repository der Agentur mit weiteren Tools integriert, womit die Qualität und Produktivität im Projekt verbessert wird. Einige Beispiele dazu sind:

- Integration mit dem Projektmanagement Tool, wie JIRA oder Asana
- Integration mit dem Chat, wie Slack oder Microsoft Teams

- Prozesse hinsichtlich Code Reviews und Merge Requests

Ein weiterer Punkt sind **Sandbox-Integrationen** von externen Systeme wie PSP, PIM, CRM oder ERP. Idealerweise sollten diese mit den Dev- und Stage-Umgebungen integriert werden, um unbeabsichtigte Änderungen oder Datenverluste auf Prod zu vermeiden. Falls das E-Commerce System Inhalte vom PIM lediglich konsumiert, können die Dev und Stage Umgebungen mit dem Produktiv PIM verbunden sein. Die Prod Umgebung sollte nach dem Go-Live jeweils mit den Produktiv Umgebungen vom PSP, PIM, CRM und ERP integriert sein.

✓ CHECKLISTE DEPLOYMENT PROZESS

- Es wurde eine Hosting-Strategie definiert (wer hostet, und welche Verantwortlichkeiten bestehen).
- Die definierten Umgebungen wurden mit einem Build & Deployment Prozess aufgesetzt, die Entwickler können zuverlässig arbeiten.
- Es wurde definiert, mit welchen externen Systeme die jeweiligen Umgebungen integriert sind.
- Der Deployment-Prozess bildet automatisierte Tests und den QA Prozess ab (es ist klar, wer welche Funktionen auf welcher Umgebung testet).
- Es wurde definiert, wo das Code Repository gehostet wird und ob dies mit weiteren Tools (z.B. JIRA, Slack, usw.) integriert ist.

4.6 QA Prozess aufsetzen

Ein gut definierter Testing-Prozess stellt sicher, dass die neue E-Commerce-Plattform stabil und gemäß den Anforderungen funktionsfähig implementiert wurde. Der Prozess sollte eng mit dem Entwicklerteam abgestimmt werden, um klare Verantwortlichkeiten und Abläufe zu etablieren. Eine wichtige Entscheidung ist, auf welchem System die einzelnen Features bzw. Tickets getestet werden. In der Regel erfolgt das Testing zunächst auf der Dev-Umgebung, bevor die Änderungen in die Stage- oder Produktivumgebung überführt werden.

Die Auswahl der **Endgeräte und Browser** ist ebenfalls entscheidend. Sie sollte auf Basis der Zielgruppenanalyse und Analytics-Daten erfolgen, damit die Tests die wirklich relevanten Nutzer abdecken. Gängige Kombinationen wie mobile Geräte mit Chrome oder Safari sowie Desktop-Browser mit Firefox und ggf. Edge sollten evaluiert werden. Zudem sollte definiert werden, welche Versionen der jeweiligen Betriebssysteme und Browser im Projekt unterstützt werden sollten.[40]

Ein fester Bestandteil des Testing-Prozesses ist die Erstellung eines **Bug-Reports** bzw. Bug-Tickets. Dieser sollte alle relevanten Informationen enthalten, darunter:

- das verwendete Gerät und den Browser inkl. Version,
- den Reproduktionspfad,
- das erwartete und das tatsächliche Ergebnis
- sowie unterstützende Materialien wie Screenshots, Videos, Order-IDs oder Links zu den betroffenen Seiten.

Die Projektmanagement-Software sollte Workflows für zwei Fälle abbilden: Erstellung eines neuen Bug-Tickets sowie das Testen einer bestehenden Story, die vom Entwicklerteam implementiert wurde. Damit wird die Nachverfolgung erleichtert und der Projektfortschritt dokumentiert. Ein klarer Unterschied sollte zwischen **Bugs** und **Verbesserungen** gemacht werden. Ein Bug bezeichnet eine Regression oder einen Fehler, bei dem etwas nicht wie in den definierten Akzeptanzkriterien der Story bzw. der Task funktioniert. Im Gegensatz dazu bezieht sich eine Verbesserung auf die Erweiterung eines Features oder die Berücksichtigung von speziellen Edge Cases, wie exotischen Browsern oder komplexen Kombinationen von Nutzerzuständen, Versand- und Zahlungsoptionen, usw.

Die Frage, **wer die Tests durchführt**, ist ebenfalls von Bedeutung. Entwickler haben den Vorteil, dass sie den Code und die Logik genau kennen, jedoch könnten sie "betriebsblind" für Fehler sein und damit die eigenen Fehler übersehen. Zudem wollen sich Entwickler in der Regel auf das Programmieren konzentrieren. Dedizierte QA-Ingenieure bieten hier eine unabhängige Perspektive und können systematisch vorgehen. QA-Ingenieure in erfahrenen Agenturen haben in der Regel ein ganzes Arsenal an gängigen Endgeräten und aktuellen Browsern, womit sie das Testing schneller durchführen können. Die jeweiligen Testszenarien werden bestenfalls in einem entsprechenden Tool, wie z.B. TestRail[41], dokumentiert. Damit können alle Teammitglieder die relevanten Szenarien einsehen und nachvollziehen, ob die Tests erfolgreich waren.

Der Testing-Prozess kann neben manuellen Tests auch **automatisierte funktionale Tests** umfassen. Manuelle Tests, die von Menschen durchgeführt werden, sind besonders bei der Prüfung von Benutzeroberflächen oder ungewöhnlichen Nutzungsszenarien wertvoll. Automatisierte Tests, beispielsweise mit Tools wie Playwright[42], bieten hingegen Effizienz und Konsistenz bei der Prüfung wiederholbarer Aufgaben. Diese beinhalten typischerweise

- Erstellung des Kundenkontos
- das Aufrufen einer Kategorieseite inkl. Anwendung von Filtern oder der Sortierung
- das Hinzufügen von Produkten in den Warenkorb
- das Anwenden eines Coupon-Codes
- das Ausfüllen der Felder im Check-outs sowie Abschluss der Bestellung.

Zusätzlich sollte **Integrationstests** in den QA-Prozess integriert werden, insbesondere für wichtige Systeme wie PSP, ERP, PIM oder CRM. Diese Tests gewährleisten, dass alle angebundenen Systeme wie konzipiert zusammenarbeiten und keine Daten verloren gehen oder falsch übertragen werden. Da Integrationstests von externen Systemen abhängen, sollten hier die Teams der integrierten Systeme unterstützen oder ggf. die Tests eigenständig durchführen. Mehr dazu im späteren Abschnitt Integrations- und **End-to-End-Tests (E2E)** Testing.

Der Fokus auf Smoke-Tests und E2E-Tests sollte im späteren Projektverlauf liegen, sobald die wichtigsten Features und Integrationen umgesetzt sind. Diese Tests prüfen, ob das gesamte System als Einheit funktioniert. Weitere Details zu diesen Tests werden im nächsten Kapitel behandelt.

✓ CHECKLISTE QA PROZESS

- Die relevanten Endgeräte und Browser für manuelle Tests wurden definiert.
- Es wurde ein QA-Prozess mit Hinblick auf die Umgebungen in der Infrastruktur aufgesetzt.
- Es wurde eine Vorlage für Bug-Reports erstellt.
- Die Personen für das Durchführen der Tests wurden bestimmt, und diese haben das richtige Tooling.
- Optional: es wurden automatisierte funktionale Tests implementiert.

4.7 Dos und Don'ts in der Konzeption und Umsetzung

Ein erfolgreiches E-Commerce-Replatforming erfordert klare Verantwortlichkeiten und Prozesse in der Konzeption und Umsetzung. Hier sind die wichtigsten Dos und Don'ts, die bei der Konzeptions- und Umsetzungsphase berücksichtigt werden sollten.

Zunächst müssen die **Verantwortlichkeiten zwischen dem E-Commerce Owner und dem externen Partner**, sowie die Rollen von Product Owner und Product Manager, klar definiert sein. Wer liefert die Anforderungen an das Entwicklerteam? Wer schreibt den Product Backlog, User Stories und koordiniert externe Stakeholder wie PIM- oder ERP-Integratoren? Falls die Verantwortlichkeiten nicht klar sind, kann es schnell zu Missverständnissen in der Timeline sowie den Anforderungen kommen. Auch wird das Entwicklerteam schnell frustriert sein, wenn unterschiedliche Anforderungen von verschiedenen Seiten kommuniziert werden.

Des Weiteren sollten die **Testing Prozesse definier**t sein. Hier sollte geklärt werden, wann der Product Owner, Product Manager oder andere Stakeholder testen sollen. Der E-Commerce Owner möchte natürlich möglichst früh mit dem Testen anfangen. Allerdings sollten interne Tests durch QA-Ingenieure vorab abgeschlossen sein, bevor der E-Commerce Owner mit den Tests beginnt, um unnötige Rückfragen zu vermeiden. In der Praxis kennt der Product Owner der entwickelnden Agentur sowohl das System inkl. Abhängigkeiten der einzelnen Features sowie das Product Backlog besser, womit der E-Commerce Owner auf sein Signal warten sollte.

Bei der Storefront Konzeption sollten **langwierige Diskussionen über Design-Details vermieden** werden. Diese beruhen oft auf Geschmacksfragen zur Ästhetik der einzelnen Stakeholder, und tragen nicht unbedingt einen Beitrag zur Usability oder Funktionsweise einzelner Features bei. Im Zweifel sollte man auf native, oder die einfachere bzw. günstigere, Storefront Implementierung setzen und A/B-Tests nach dem MVP Go-Live planen. Durch das Sammeln echter Daten können Entscheidungen daten- und faktenbasiert getroffen werden.

Bei der **Priorisierung der Features sollte nach Business Value** priorisiert werden. Man sollte sich also auf Funktionen konzentrieren, die den Umsatz steigern, Kosten senken oder die Effizienz bzw. Usability für Mitarbeiter und Nutzer verbessern. Man sollte sich nicht von persönlichen Meinungen oder den lautesten Stakeholdern leiten lassen.

Die Migration der **umgezogenen Daten sollte umfassend geprüft** werden, insbesondere bei Bestellungen, Kunden und Produktbewertungen. Man sollte sicherstellen, dass alle Relationen korrekt sind, z. B. dass Bewertungen den richtigen Produkten sind oder Bestellungen in Berichten auftauchen. Oftmals gibt es in gängigen Shop-Systemen eine Report- oder Export-Funktion von Datensätzen, sodass man mit Leichtigkeit die Summen der migrierten Daten einsehen und mit dem Quellsystem abgleichen kann.

Wichtig ist auch, **alle Parteien in die Timeline einzubeziehen.** Interne Teams und externe Partner sollten bei Ihren Arbeitsschritten, wie z. B. für Produktimporte, Storefront-Umsetzungen, ERP- und CRM-Integrationen sowie Marketing, koordiniert werden. Damit können die involvierten Akteure ihre Ressourcen rechtzeitig einplanen. Auch sollte die Timeline realistisch, mit einem gewissen Puffer geplant werden, d.h. Bedenken von einzelnen Akteuren sollten ernst genommen werden. Unrealistische Deadlines sehen für das Management bei der Planung attraktiv aus. Tatsächlich werden die involvierten Akteure schnell frustriert sein, wenn Deadlines nicht einzuhalten sind.

Bei den **Verantwortlichkeiten der Infrastruktur sollte zukunftsorientiert** geplant werden. Die eigene IT-Abteilung sowie ein spezialisierter Hoster können zwar oft eine erste funktionierende Dev-Umgebung im Rahmen des Zeitplans aufsetzen. Nach dem Livegang kommt es jedoch unter Produktivbedingungen oft zu unvorhersehbaren Ereignissen und Downtimes. Es entsteht Aufwand für zusätzliche Abstimmung und Fehlerfindung zwischen dem Hoster (Infrastruktur) und den E-Commerce Entwicklern, während die Gründe der Downtimes erst spät gefunden werden. Die Kommunikationsaufwände sowie längere Downtimes lassen sich meistens vermeiden, wenn sowohl die Entwicklung wie Hosting aus einer Hand kommen.

Schließlich sollte jedes **Replatforming als Change Management Projekt** gesehen werden**.** Dabei sollte man auf Ängste und Bedürfnisse der Beteiligten eingehen und die Vorteile der neuen Plattform erklären. Der E-Commerce Owner sollte Vertrauen aufbauen, um Motivation und Verständnis sicherzustellen. Andernfalls kann es zu Demotivation und, im schlimmsten Fall, Sabotage im Team kommen.

KAPITEL 5

TESTING UND GO-LIVE

Das Kapitel Testing und Go-Live bildet den Abschluss des Replatforming-Projekts und ist von entscheidender Bedeutung für den langfristigen Erfolg der neuen Plattform. Bevor das neue E-Commerce System live geht, müssen alle Elemente gründlich geprüft werden – von der technischen Stabilität über die Datenmigration bis hin zur Benutzerfreundlichkeit. Ein fehlerfreier Livegang ist entscheidend, da Probleme in dieser Phase direkt die Kundenzufriedenheit und den Geschäftserfolg beeinträchtigen können.

Zu Beginn dieses Kapitels wird beleuchtet, wie wichtige SEO Grundlagen berücksichtigt werden, um nach dem Relaunch negativen Auswirkungen auf die Sichtbarkeit in Suchmaschinen zu vermeiden. Außerdem wird der Umzug von Analytics und Tag Manager thematisiert, damit alle Tracking-Mechanismen auch nach dem Go-Live korrekt funktionieren.

Die Schulung des Teams spielt ebenfalls eine zentrale Rolle, damit alle Beteiligten sicher und effizient mit dem neuen System arbeiten können. Ergänzend dazu wird die Projektdokumentation beleuchtet, die als Wissensspeicher für zukünftige Optimierungen und das Debugging dient.

Das Kapitel widmet sich zudem den verschiedenen Testarten: Vom schnellen und effizienten Smoketesting, das die Grundfunktionalitäten überprüft, bis hin zu umfassenden Integrationstests und End-to-End-Tests, die sicherstellen, dass alle Geschäftsprozesse und Integrationen reibungslos funktionieren.

Schließlich wird in der Go-Live-Agenda beschrieben, wie der finale Übergang strukturiert und reibungslos gestaltet wird. Das Kapitel wird mit einer Zusammenstellung von Dos und Don'ts abgerundet. Dieses Kapitel liefert einen praxisorientierten Leitfaden, um den Livegang eines neuen E-Commerce-Systems erfolgreich und ohne größere Komplikationen zu meistern.

5.1 SEO Basics

Ein erfolgreiches E-Commerce-Replatforming erfordert eine sorgfältige Berücksichtigung von SEO-Grundlagen, um sicherzustellen, dass die Sichtbarkeit der Website in Suchmaschinen nicht beeinträchtigt wird. Im ersten Schritt sollte auf dem Pre-prod System ein **Check auf Best Practices** erfolgen. Dazu gehört das korrekte Setzen von Meta-Tags wie Title, H1 und Description, die Implementierung von Canonical-Tags, Hreflang-Tags bei Mehrsprachigkeit, OpenGraph-Tags für Social Media und Google Structured Data mithilfe von Schema.org. Ein Test über Tools wie den Schema.org Markup Validator[43] kann hier hilfreich sein, um Fehler frühzeitig zu erkennen.

Crawling und Fehlerprüfung mittels entsprechender SEO Tools[44] des neuen Systems sind essenziell, um mögliche Probleme zu identifizieren. Es sollten sowohl semantische Fehler, z. B. Duplicate Content, unklare Headline-Hierarchien oder doppelte Meta-Tags, als auch technische Fehler, z. B. 404-Seiten, langsame Ladezeiten, lange oder unlogische URL-Strukturen, fehlerhafte robots.txt oder falsche Canonical-Tags, überprüft werden. Links, die durch Filter- und Sortierfunktionen entstehen, sollten maskiert und für Suchmaschinen gesperrt werden, um unnötige Indexierung zu vermeiden. Damit wird das Crawling Budget von Suchmaschinen effizient genutzt, womit sich das SEO Ranking verbessert.[45]

Auf der Pre-Prod- oder Staging-Umgebung sollte ein **kontrolliertes Crawling** stattfinden, jedoch ohne Indexierung durch Suchmaschinen. Die Umgebungen sollten also durch ein IP-Allowlisting oder eine Basic Auth Beschränkung vor öffentlichen Zugriffen geschützt sein. Damit SEO Crawler dennoch auf die nicht-öffentlichen URLs zugreifen können, sollte die IP des Crawlers im Allowlisting hinzugefügt, oder der Basic Auth User im URL Request Header hinterlegt werden. Gleichzeitig ist eine **Content-Inventur des alten Systems** durchzuführen. Hierbei gilt es zu prüfen, welche Inhalte SEO-relevant sind, wie etwa Produkte, Kategorien und Content-Seiten (z. B. „Über uns“). Anschließend kann man leicht nachvollziehen, ob diese Inhalte bereits in das neue System übertragen wurden.

Das Crawling des alten und des neuen Systems liefert eine Übersicht der URLs zwischen altem und neuen System, was die Basis zur Erstellung der **301-Redirects** bildet. Diese sollen sicherstellen, dass die Rankings durch SEO-relevante Inhalte nicht verloren gehen.[46] Hierfür sollten alte URLs aus einem Crawl des bisherigen Systems mit den neuen URLs auf Pre-Prod abgeglichen werden. Externe Backlinks sollten

ebenfalls geprüft und bei Bedarf angepasst werden. Die notwendigen 301-Redirects können zunächst in einer Excel-Tabelle dokumentiert und anschließend in das neue E-Commerce System übertragen werden. Die eingeführten 301 Redirects sollten jedoch erst nach dem Go-Live aktiv geschaltet werden. Auch sollten nach dem Go-Live URLs in Marketingkampagnen angepasst werden. Alternativ sollten die in Marketingkampagnen bereits hinterlegten (alten URLs) mit den URL-Parametern aus den Marketingkampagnen in 301 Redirects berücksichtigt sein.[47] Eine Nichtberücksichtigung der Marketingkampagnen URLs wäre aus Marketing-Sicht doppelt ärgerlich, da man dennoch CPC Kosten zahlt und die entsprechenden Ad-Links auf falsche Seiten führen.

Die **robots.txt** Datei sollte so konfiguriert werden, dass sensible Bereiche wie der Check-out oder die Benutzerkontoverwaltung nicht indexiert werden. Gleichzeitig sollte mit dem Go-Live eine aktualisierte XML-Sitemap erstellt und in der Google Search Console eingereicht werden.

Nach dem Go-Live ist ein **SEO-Drop** nicht ungewöhnlich und sollte die Verantwortlichen nicht beunruhigen. Ein leichter Einbruch in Rankings und Traffic ist normal, da Google Zeit benötigt, um die neue Seite vollständig zu crawlen und zu bewerten. In der Regel verbessert sich die Performance nach einigen Wochen, wenn alle SEO-Maßnahmen sauber umgesetzt wurden.

✓ CHECKLISTE SEO BASICS

- Nach Implementierung des Storefronts und Erstellung des Contents, wurde die Pre-prod Umgebung auf einige SEO Best Practices geprüft, z.B.
- Meta-Tags korrekt gesetzt (Title, H1, Description)
- Canonical-Tags vorhanden und korrekt implementiert
- Hreflang-Tags für Mehrsprachigkeit vorhanden
- OpenGraph-Tags für Social Media gepflegt
- Schema.org Markup integriert und validiert
- Es wurde ein SEO Crawling von den Pre-prod und Prod Umgebungen durchgeführt. Dieses dient als Basis für
- Prüfung von Duplicate Content, falschen Meta-Tags, unlogischer Headline Struktur, defekten Links, ...
- Erstellung eines 301 Redirect Konzepts
- Es wurden eine robots.txt inkl. Sitemap generiert.

5.2 Analytics und Tag Manager Umzug

Der Umzug von Analytics und Tagging-Tools wie Google Analytics 4 (GA4), Matomo oder dem Google Tag Manager (GTM) auf ein neues E-Commerce System erfordert eine systematische Vorgehensweise. Da sich die Funktionsweise der Tools sehr ähnlich ist, kann das unten beschriebene Vorgehen weitestgehend für die meisten Analytics- und Tag-Manager-Tools übernommen werden.

Bevor der Umzug beginnt, sollte eine **vollständige Inventur** des bestehenden Tag-Managers durchgeführt werden:

- Tracking-Setup prüfen: Alle bestehenden Tags, Trigger und Variablen im Tag Manager sollten dokumentiert werden. Unnötige oder veraltete Trackings, die keine wertvollen Daten liefern, können entfernt werden
- Analytics-Inventur: Man sollte eine Übersicht der bestehenden Reports, Ziele (Conversions), benutzerdefinierten Dimensionen und Dashboards, die in der neuen Umgebung benötigt werden, erstellen
- DataLayer prüfen: Die im bisherigen System verwendeten DataLayer sollten hinsichtlich Struktur und Variablen detailliert erfasst werden

Im Anschluss kann im Tag Manager eine Sandbox Umgebung aufgesetzt und mit dem Stage oder Pre-prod integriert werden. Der Google Tag Manager wird bei gängigen Shopsystemen wie Shopware oder Magento oft durch das Hinterlegen der **GTM-Container-ID** in einer entsprechenden Extension integriert. Alternativ können Tags direkt über **gtag.js** im E-Commerce System implementiert werden. Falls eine Integration mit dem GTM genutzt wird, können beliebige (externe) Tags im GTM ohne Code Deployment konfiguriert werden. Falls eine gtag.js Integration genutzt wird, werden Tags direkt im gtag.js Code getätigt. Zudem können in diesem Szenario die erfassten Daten lediglich an Google Produkte ausgeleitet werden.[48]

Sofern mit dem Replatforming kein neues Tracking Konzept implementiert wird, sollte der **DataLayer** im neuen System gemäß dem alten System nachgebaut werden. In diesem Fall bleibt die Struktur möglichst analog zum alten System und es wird vermieden, dass bestehende Tags und Trigger angepasst werden müssen. Zum einen sollte man darauf achten, dass Produkt-IDs, Preise, Kategorien und Checkout-Ereignisse korrekt in den DataLayer übertragen werden. Zum Anderen sollten Events wie „addToCart“, „purchase“ oder andere E-Commerce-Aktionen fehlerfrei getriggert werden.

Bevor das Tracking live geschaltet wird, sollte es in der Staging-Umgebung getestet werden. Hierfür sollte man einen neuen Sandbox-GTM-Container nutzen, um das Tracking zunächst isoliert zu prüfen. Im GTM-Debug-Modus kann getestet werden, ob Tags wie Pageviews, Events und E-Commerce-Events korrekt feuern. Auch kann man prüfen, ob Variablen wie Produktdaten oder Benutzerinteraktionen sauber aus dem DataLayer ausgelesen werden.[49]

Nachdem die GTM Integration abgeschlossen wurde, kann die **Analytics Integration** erfolgen. Die Integration von Analytics kann entweder über den Tag Manager oder direkt über das E-Commerce System erfolgen. Bei einer Analytics **Integration über GTM** haben Marketer mehr Flexibilität bei Anpassungen, Erstellung von eigenen Analytics Reports und neuen Tags ohne Eingriff in den Code. Zudem gibt es eine bessere Übersicht und Kontrolle aller Tracking-Tags an einem zentralen Ort. Bei einer direkten Integration zwischen dem E-Commerce System und Analytics gibt es weniger Abhängigkeit von GTM-Konfigurationen. In jedem Fall sollte zunächst eine Sandbox Umgebung im Analytics Tool genutzt werden, und mit Pre-prod bzw. Stage integriert werden.

Da sich die URL-Struktur im Check-out-Prozess des neuen E-Commerce Systems ändern kann, müssen **Conversion-Events** und der **Checkout-Funnel** in Analytics angepasst werden. Zunächst sollte geprüft werden, dass Conversion-Daten wie **Transaktionen** oder **Umsätze** korrekt erfasst werden. Anschließend können die neuen Schritte und Ziele im Check-out definiert werden.

In der Analytics-Sandbox muss getestet werden, ob die Daten aus dem neuen E-Commerce System korrekt ankommen. Dazu gehören Pageviews, Events für wichtige Seiten und Aktionen sowie E-Commerce-Daten wie Warenkorb-Additionen, Check-out-Aktionen und natürlich Conversions.

Kurz vor dem Go-Live kann die E-Commerce **Pre-Prod** Umgebung direkt mit dem produktiven GA- und GTM-Setup integriert werden. Da in dieser Phase keine weiteren Änderungen mehr vorgenommen werden, bleibt das Tracking stabil, und es entstehen keine Tracking-Lücken beim Livegang.

Mit diesem strukturierten Vorgehen wird sichergestellt, dass der Umzug von Analytics und dem Tag Manager reibungslos verläuft. Das neue System erfasst alle relevanten Daten präzise, und die Performance des E-Commerce Systems kann ohne Unterbrechung weiter analysiert und optimiert werden. Ein korrektes Tracking stellt

sicher, dass im Zuge des Replatformings Marketing Kampagnen funktionieren. Ein konsistentes Analytics hingegen ist essenziell, um den Erfolg und die definierten Ziele des Replatformings zu messen.

✓ CHECKLISTE GA UND GTM UMZUG

- Das bestehende GTM Setup inkl. Analytics und DataLayer ist dokumentiert.
- Im der neuen E-Commerce Platform ist der DataLayer analog zum bestehenden nachgebildet.
- Die neue E-Commerce Platform (STAGE oder Pre-PROD) ist mit einem GTM Sandbox Container integriert (inkl. Analytics) und die Integration wurde geprüft.
- Eventuell fehlende DataLayer und Events wurden nachgezogen.
- Nach dem Livegang wurde die Integration nochmals überprüft.

5.3 Schulung neues E-Commerce System

Die Einführung eines neuen E-Commerce Systems erfordert eine sorgfältige Schulung der beteiligten Stakeholder, damit alle Prozesse und Workflows weitergeführt werden können. Ein wichtiger erster Schritt ist die Dokumentation der Stakeholder, Prozesse und Workflows im alten System, wie sie im Abschnitt Change Management beschrieben wurden. Dazu gehören Rollen wie der Content-Manager, Marketer, der Produktdatenmanager und der Kundenservice. Anschließend muss festgestellt werden, welche Workflows und Prozesse im neuen System nach dem Replatforming weiterhin relevant sind.

Zu den typischen Bereichen gehören das **Produktdatenmanagement**, z. B. die Pflege von Kategorien und Attributen oder der Import aus einem PIM, die **Content-Erstellung** mithilfe eines Pagebuilders oder dem CMS sowie das **Order-Management**, das Aufgaben wie den Export von Bestellungen, das Erstellen von Rechnungen und die Versandabwicklung umfasst. Auch spezielle Themen wie **Promotions**, **Preisregeln**, **Searchandising** (Suche und Facettennavigation, Produkt-Boosting) sowie **SEO-Einstellungen** (Meta-Daten, OpenGraph-Tags) sollten berücksichtigt werden. Idealerweise entsteht während der Entwicklung der neuen E-Commerce Platform eine Dokumentation der relevanten nativen Funktionen sowie der projektspezifischen Eigenentwicklungen. Mehr dazu um nächsten Abschnitt Projektdokumentation.

Ein sinnvoller und zeitsparender Ansatz besteht darin, zunächst auf öffentlich verfügbare Dokumentationen, User Guides und Trainings zurückzugreifen. Für gängige Systeme wie Shopify, Shopware oder Magento 2 bieten Plattformen wie die **Shopware Academy** oder der **Adobe Commerce User Guide** wertvolle Ressourcen. In diesem Fall erhalten die Anwender eine gute Basis zur Nutzung der nativen Funktionen des neuen E-Commerce Systems. Ergänzend dazu sollten spezifische Schulungen oder Workshops durch die E-Commerce-Agentur organisiert werden, die sich auf projektspezifische Anpassungen und neue Funktionen konzentrieren. Diese Schulungen sollten vor allem darauf abzielen, Unterschiede und Verbesserungen im Vergleich zum alten System verständlich zu machen.

Die Workshops müssen frühzeitig in der Projekt-Timeline eingeplant werden, um Abhängigkeiten zu berücksichtigen – etwa bei Produktimporten, ERP-Integrationen oder der Content-Migration. Es ist wichtig, zwischen der Schulung und dem Go-Live

genügend Pufferzeit einzuplanen. Diese Zeit kann genutzt werden, um das Gelernte in der Praxis zu üben, Testläufe durchzuführen, Rückfragen zu klären und eventuell auftretende Change Requests zu bearbeiten. Sie ist auch notwendig, um den Content vor dem Go-Live vollständig zu pflegen.

Schließlich sollte ein klarer Plan für den Support nach der Schulung vorhanden sein. Es muss festgelegt werden, wer im Alltag bei Fragen oder Problemen Unterstützung bietet – sei es die E-Commerce-Agentur oder der Softwareanbieter, wobei Letzterer oft nur bei kommerziellen Support-Plänen Unterstützung leistet. Mit einer gut geplanten und durchgeführten Schulung wird sichergestellt, dass das Team mit dem neuen System effizient arbeiten kann und damit dir Erwartungen an das neue E-Commerce System erfüllt sind.

✓ CHECKLISTE SCHULUNG

- Es gibt eine Übersicht, welche Personen mit dem aktuellen System (und Umsystemen) arbeiten.
- Es gibt eine Übersicht, welche Prozesse sich im Zuge der Replatformings ändern und ob es einer Schulung bedarf, bspw.
  - Produktdatenmanagenent (inkl. Kategorien und Attribute) und PIM Integration
  - Pagebuilder und CMS
  - Order Management und ERP Integration
  - Promotions, Preis- und Rabattregeln
  - Searchendising (Filter, Suche, usw.)
  - SEO Einstellungen
- Je nach neuem E-Commerce System haben sich die Schulungsteilnehmer bereits mit öffentlichen Ressourcen bekannt gemacht (User Guides, Dokumentationen, Youtube, ...).
- Die Schulungen sind auf den Personenkreis und auf die spezifischen Prozesse zugeschnitten.
- Die Schulungen sind gemäß der Timeline mit entsprechendem Puffer zum Go-Live geplant.
- Der Support nach Schulungen ist geregelt.

5.4 Projektdokumentation

Die Projektdokumentation dient als Nachschlagewerk für verschiedene Zielgruppen und gewährleistet, dass wichtige Informationen leicht zugänglich sind, auch nach Abschluss des Projekts. Zunächst sollten die Zielgruppen der Projektdokumentation geklärt werden, und der Inhalt sollte entsprechend angepasst werden:

- Anwender, z. B. CMS, Marketing, Produktdatenmanager: Fokus auf Bedienung, Workflows und häufig genutzte Funktionen.
- Entwickler und Integratoren: Details zu Eigenentwicklungen, eingesetzten Modulen, Integrationen, Daten-Mappings und technischen Spezifikationen wie API-Payloads.
- E-Commerce-Manager: Übersicht über Projektziele, Zugänge, Gesprächsprotokolle und wichtige Prozessschritte.

Bei größeren Projekten und Teams kann ein professionelles Tool wie Confluence sinnvoll sein.[50] Moderne Dokumentationstools bieten Kommentarfunktionen, die das Feedback und die Zusammenarbeit erleichtern. Mit einer Versionierung verfolgt man Änderungen in der Dokumentation nach und kann Entwürfe oder Konzepte freigeben. Mit einem Rechte- und Rollensystem wird sichergestellt, dass der gewünschte Nutzerkreis bestimmte Inhalte sehen und editieren kann. Eine Exportfunktion von Inhalten erlaubt das Teilen mit externen Stakeholder ohne eigene Zugänge.

Die folgende Tabelle zeigt eine mögliche Gliederung der Projektdokumentation:

GLIEDERUNGSPUNKT	KOMMENTAR UND BESCHREIBUNG
1.1 Ziele, IST und KPIs	Dokumentation der Projektziele, aktueller Zustand und Zielwerte (KPIs)
1.2 Umfang des Replatformings	Beschreibung des Projektumfangs und der Anforderungen vor Projektstart, z. B. neues Storefront-Design, ERP-Integration, PIM-Anbindung usw.
1.3 Stakeholder & Kontakte	Liste der Projekt-Stakeholder, inklusive Rollen und Kontaktdaten
1.4 Zeitplan	Übersicht über die Projektphasen und wichtige Meilensteine, die im Zeitplan erarbeitet wurden

GLIEDERUNGSPUNKT	KOMMENTAR UND BESCHREIBUNG
1.5 Erarbeitete Materialien	Erarbeitete Dokumente oder Analysen, z. B. Business Model Canvas, Wettbewerbsanalysen oder Zielgruppenanalysen
1.6 Links zu Designs	Links zu externe Tools wie Figma mit dem ausgearbeiteten Storefront-Design und Prototypen
2. Technische Dokumentation (allgemein)	Übersicht über die Infrastruktur, darunter Umgebungen, URLs und Zugänge sowie ggf. in der Infrastruktur genutzt Ressourcen, z.B. Datenbank, CDN, Cache Server, ...
2.1 Mandanten bzw. Sub-Shops	Details zu Mandanten/Sub-Shops inkl. URLs, Sprachen, Währungen, genutzten Themes, Verknüpfungen zu externen Systemen, z.B. ERP, PSP oder PIM
2.2 Konfigurationen	Beschreibung von nativen Konfigurationen, bspw. Steuer-, Versand- und Zahlungs-Einstellungen sowie SEO-Konfigurationen
2.3 Schnittstellen und Systemlandschaft	Dokumentation der gesamten Systemlandschaft sowie der final angebundenen Systeme inkl. SFTP Serverdaten oder API Endpoints, Mappings, Validierungen und Beispielen, wie Payloads oder Dateien zum Order Export oder Stock Import
2.4 Migration	Beschreibung der migrierten Datensätze und des genutzten Migrationskonzeptes
2.5 Custom Features (Eigenentwicklungen)	Beschreibung individueller Anpassungen inkl. den möglichen Konfigurationen im Adminbereich, Ticket Referenz und Links zu Code-Repositories und Modulen
3. Test-Szenarien	Dokumentation von getesteten Endgeräten, Browsern und Testprotokollen für z.B. Smoketests
4. Extensions und API-Keys	Liste der verwendeten Extensions, API-Keys und Zugänge zu Admin Bereichen externer Dienste, z. B. PIM, PSP, Newsletter, usw.

GLIEDERUNGSPUNKT	KOMMENTAR UND BESCHREIBUNG
5. Gesprächsprotokolle	Protokolle aus relevanten Meetings wie Sprint-Reviews, Retrospektiven inklusive Entscheidungen und den resultieren Action Items
6. Change Requests	Übersicht von Änderungswünschen und deren Freigaben, Ticket Referenz und Links zu Code-Repositories und Modulen
7. Schulungsmaterialien	Links zu offiziellen User-Guides und selbst erstellten Schulungsunterlagen
8. Known Issues	Liste bekannter Probleme und Troubleshooting-Tipps
9. Release Notes	Zusammenfassung von Deployments nach dem Go-Live, inklusive Verweise auf Tickets

Mit einer gut strukturierten und vollständigen Projektdokumentation können zukünftige Herausforderungen einfacher bewältigt und Projekterfolge nachhaltig gesichert werden. Zudem wird sichergestellt, dass bei personellen Änderungen und einem Wechsel der Agentur alle relevanten Informationen erhalten bleiben.

✓ CHECKLISTE PROJEKTDOKUMENTATION

- Der Anwenderkreis der Projektdokumentation identifiziert.
- Basierend auf dem Anwenderkreis sind die wesentlichen Inhalte sowie die notwendige technische Tiefe ist bekannt.
- Es wird eine Dokumentationssoftware eingesetzt, die ein gewisses Maß an Kollaboration ermöglicht.
- Die Projektdokumentation steht allen Projektbeteiligten zur Verfügung (ggf. nur mit Leserechten) und sie wird regelmäßig geprüft und aktualisiert.

5.5 Smoketesting

Smoketesting ist ein wichtiger Bestandteil des Testprozesses in einem E-Commerce-Projekt, insbesondere vor dem Go-Live. Während einzelne Features oft isoliert manuell getestet werden, umfasst Smoketesting eine ganzheitliche Prüfung der wichtigsten E-Commerce Funktionen und deren Abhängigkeiten, um eine Stabilität des Gesamtsystems zu gewährleisten.

Bei einem Deployment werden oft mehrere Features, Updates und Änderungen gleichzeitig eingespielt. Diese können unerwartete Wechselwirkungen verursachen, die nicht durch isolierte Einzeltests abgedeckt werden. Smoketests prüfen daher systemübergreifend, ob die kritischen Funktionen eines Online-Shops weiterhin wie erwartet arbeiten. Da der Go-Live als ein erstes, "großes" Deployment angesehen werden kann, sollte das Smoketesting im Projekt Zeitplan und der Projektdokumentation abgebildet sein. Smoketesting dient in erster Linie zum

- Sicherstellen, dass der Kaufprozess reibungslos funktioniert
- Überprüfung von Integrationen wie PSP, ERP-Systemen oder PIM-Lösungen
- Vermeidung von „Showstoppern", die das Kundenerlebnis massiv beeinträchtigen könnten, z. B. fehlerhafte Checkout-Prozesse oder Probleme mit der Produktsuche.

Smoketests sollten daher auf die wichtigsten E-Commerce-Funktionen fokussiert sein. Dazu gehören unter anderem:

- **Suche:**
  - Funktioniert die Suche für wichtige Keywords und werden die erwarteten Ergebnisse angezeigt?
  - Wird das Autosuggest korrekt, z.B. Produkte mit Bildern und Preisen, angezeigt?
  - Sind die Filter und die Sortierung auf der Suchergebnisseite funktionsfähig?
- **Navigation und Kategorien:**
  - Können bestimmte Kategorien über die Navigation aufgerufen werden?
  - Werden Produktkacheln korrekt angezeigt, z.B. mit Preis, Bild, Name, Verfügbarkeit?

- **Produkt- und Warenkorb-Funktionen:**
  - Werden auf der Produktdetailseite (PDP) relevante Informationen wie etwa Preis, Beschreibung und Assets (z.B. Produktbilder und Downloads) angezeigt?
  - Lassen sich Produkte von der Kategorie- oder Produktdetailseite in den Warenkorb legen?
  - Funktioniert das Wechseln von Varianten, z. B. Größe bei Schuhen?
  - Kann die Menge im Warenkorb geändert oder ein Produkt entfernt werden?
  - Funktionieren Rabatt- und Coupon-Codes korrekt?
- **Check-out-Prozess:**
  - Kann der Check-out als Gast abgeschlossen werden?
  - Werden korrekte Steuern, Versandkosten und Zahlungsmethoden für verschiedene Länder berechnet?
  - Funktionieren Test-Zahlungsmittel wie Test-Kreditkarten?[51]
  - Läuft der Check-out-Prozess auch für eingeloggte Kunden und unterschiedliche Kundengruppen, etwa B2C vs. B2B?
- **Kundenkonto:**
  - Ist die Registrierung eines neuen Kundenkontos mit dem Ausfüllen der Formularfelder möglich?
  - Funktioniert der Login für bestehende Konten?
  - Kann der Kundenbereich genutzt werden, z. B. Passwort ändern, oder Bestellhistorie einsehen?
- **E-Mail-Benachrichtigungen:**
- Werden Bestellbestätigungen, Versandbenachrichtigungen und andere relevante E-Mails korrekt versandt?
- **Integrationen:**
- Funktionieren angebundene Systeme wie ERP, PIM und PSP wie erwartet? Werden Bestellungen und Daten korrekt weitergeleitet?

Damit alle relevanten Nutzergruppen abgedeckt werden, sollten die Tests auf unterschiedlichen Kombinationen von **Sprachen, Endgeräten (Betriebssystem) und Browsern** durchgeführt werden. Beispielsweise:

- **Sprachen:** Testen der wichtigsten Sprachversionen des Storefronts, wie Deutsch oder Englisch.
- **Endgeräte:** Desktop, Tablet, Smartphone.

- **Browser:** Chrome, Safari, Firefox, Edge – jeweils in aktuellen Versionen.

Die relevanten Nutzergruppen wurden bereits in der Zielgruppenanalyse ermittelt.

Automatisierte Tests mit Tools wie **Playwright** gewinnen im E-Commerce zunehmend an Bedeutung. Bei einem automatisierten Test wird zunächst ein Testpfad programmiert, der von einem Bot oder einem Script ausgeführt wird. Gänge Tools für automatisierte Tests erlauben eine Echtzeit Nachverfolgung und nehmen ein Video auf. Auch erhält man ein Protokoll mit den positiv und negativ getesteten Schritten. Durch automatisierte kann der E-Commerce Owner „Showstopper“[52] frühzeitig erkennen, womit das Bug-Fixing früher gestartet werden kann. Zudem werden die manuellen Ressourcen effizient genutzt, insbesondere bei einer hohen Anzahl an Test-Szenarien und Endgeräte-/Browser-Kombinationen. Da Smoketesting für QA-Ingenieure zeitintensiv und wiederholend sind, können sich schnell menschliche Fehler einschleichen, die bei einem programmatischen Vorgehen eher nicht vorkommen. In Summe ermöglichen automatisierte Tests kürzere Release-Zyklen bei besserer Qualität der Tests.

Die Einführung von automatisierten Tests ist mit einem gewissen Initial-Aufwand verbunden, da die Tests zunächst programmiert werden müssen. Zudem sollten die automatisierten Tests laufend aktualisiert werden, da neue Funktionen und Software-Updates in das E-Commerce System eingespielt werden. Automatisierte Tests können also als Investition betrachtet werden, die sich in den folgenden Fällen rentieren:

- Bei komplexen Projekten mit vielen relevanten Test-Szenarien und Endgeräte-/Browser-Kombinationen.
- Bei Projekten mit häufigen Deployments, da die Initialkosten für das Aufsetzen automatisierter Tests über Zeit amortisiert werden.
- Wenn wiederholbare und standardisierte Test-Szenarien notwendig sind, z.B. der Check-out oder das Warenkorbverhalten.

Nach einem Deployment sollte zusätzlich ein kurzer Sanity Check auf Produktiv durchgeführt werden. Dieser umfasst eine schlanke Version des Smoketestings und überprüft nochmals die wichtigsten Funktionen. So wird sichergestellt, dass das Deployment keine unvorhergesehenen Probleme verursacht hat.

Smoketests sind unverzichtbar, um kritische Funktionen eines E-Commerce-Systems zu sichern, und sollten dementsprechend sorgfältig vor dem Go-Live eingeplant

werden. Sie bieten eine grundlegende Absicherung vor dem Deployment und minimieren das Risiko schwerwiegender Fehler, die die Nutzererfahrung beeinträchtigen könnten.

✓ CHECKLISTE SMOKETESTING

- Die Tester sowie die Testumgebung (z.B. STAGE) wurden für das Smoketesting definiert.
- Es besteht eine Übersicht an Szenarien (Dokumentation) der zu prüfenden Funktionen im Rahmen der Smoketests. Diese können bspw. die folgenden Bereiche betreffen:
  - Suche und Navigation
  - Kategorieseite inkl. Filter und Sortierung
  - Produktdetailseite
  - Warenkorb
  - Checkout
  - Kundenbereich
  - Kommunikation inkl. E-Mail Versand
- Der Zeitrahmen für Smoketests ist in der Projekt Timeline abgebildet.
- Die relevanten Endgeräte und Browser sind definiert.
- Nach dem eigentlichen Deployment bzw. Go-Live findet ein Sanity Check statt, der auf Produktiv die wesentlichen Funktionen in Kürze prüft. Dieser ist ebenfalls in Form einer Checkliste dokumentiert.
- Optional: es wurden automatisierte funktionale Tests eingeführt.

5.6 Integrationstests und End-to-End Tests

Integrationstests und End-to-End (E2E) Tests spielen eine zentrale Rolle in der Qualitätssicherung bei der Kommunikationen mit externen Systemen. Sie haben unterschiedliche Ziele, ergänzen sich jedoch, um sowohl einzelne Integrationen als auch komplette Geschäftsprozesse hinsichtlich der konzipierten Funktionsweise zu prüfen.

Integrationstests prüfen, ob die Kommunikation zwischen zwei Systemen, wie im Integrationskonzept geplant, funktioniert. Dabei liegt der Fokus auf den Schnittstellen und der korrekten Datenübertragung, einschließlich Mappings (die Felder zwischen den Systemen werden korrekt übersetzt werden, z. B. Preis oder Produktbeschreibung vom PIM zum E-Commerce System) und Validierungen (Dateneingaben entsprechen den Anforderungen entsprechen, z.B. auf maximal Anzahl der Ziffern oder ungültige Zeichen). Im Gegensatz dazu testen E2E-Tests komplette Geschäftsprozesse vom Anfang bis zum Ende. Dabei wird nicht nur die Kommunikation zwischen Systemen geprüft, sondern auch die Benutzererfahrung und die Funktionalität über mehrere Schritte hinweg. Im E-Commerce-Kontext umfasst das z. B. den gesamten Check-out-Prozess oder die Veröffentlichung eines Produkts ausgehend PIM bis ins Storefront. Zeitlich gesehen sollten Integrationstest so früh wie möglich durchgeführt werden, sobald eine Integration implementiert ist. So können grundlegende Probleme in der Kommunikation frühzeitig erkannt und behoben werden. E2E-Tests sollten erst dann durchgeführt werden, wenn die wesentlichen Konfigurationen, Features und Integrationen abgeschlossen sind. Sie dienen als finale Absicherung, bevor das System live geht.

Im Folgenden sind einige Praxisbeispiele für E2E-Tests beschrieben:

Produktimport aus dem PIM:

- Upload von Produktstammdaten, z. B. aus einer Excel-Datei, ins PIM
- Anreicherung der Produktdaten im PIM, z. B. Bilder, Texte, Attribute
- Freigabe der Produkte für den Export an das E-Commerce System durch einen PIM Nutzer mit entsprechenden Rechten
- Import der Produkte in das E-Commerce System
- Veröffentlichung der Produkte im Storefront, inkl. Überprüfung, ob sie in der Suche und der Kategorien korrekt erscheinen

- Alle relevanten Informationen, wie Produktbilder, Artikelnummer, Preise, Attribute, Beschreibungen, erscheinen an der erwarteten Stelle auf der PDP mit den korrekten Werten aus dem PIM
- Änderungen im PIM, z. B. andere Bilder, und Überprüfung, ob diese im Storefront korrekt angezeigt werden

Kundenanlageprozess:

- Registrierung eines neuen Kunden über das Formular im Storefront entsprechend der definierten Felder und Validierungen, z.B. Prüfung auf eine gültige VAT-ID
- Versand einer automatisierten E-Mail mit CSV-Daten des Formulars an den Kundendienst
- Verarbeitung der Daten im CRM, inkl. Prüfung auf Duplikate und Anreicherung, z. B. Kundengruppe oder Adresse
- Export des Kundenkontos zurück an das E-Commerce System
- Erstellung eines Kontos im E-Commerce System und Versand einer Begrüßung-E-Mail an den Kunden, inkl. “Passwort setzen” Link

Check-out und Fulfillment:

- Abschluss des Check-outs mit den erwarteten Feldern, Versand- und Zahlungsmethoden
- Versand der Bestätigungs-E-Mail mit korrekten Informationen, z.B. Warenkorb, Adresse, Versand- und Zahlungsmethode
- Überprüfung der Transaktion beim Payment Service Provider
- Export der Bestellung an das ERP-System
- Erstellung und Kommunikation der Rechnung im ERP, inkl. Meldung an das E-Commerce System
- Erstellung und Kommunikation der Versandbenachrichtigung, inkl. Meldung an das E-Commerce System
- Anpassung des Lagerbestands im ERP
- Import des aktualisierten Lagerbestands zurück in das E-Commerce System

Da jedes E-Commerce Projekt von einem reibungslosen Check-out-Prozess und den dahinterliegenden Integrationen lebt, sollte auf diesen Bereich der User Journey der Schwerpunkt liegen. Zudem sollte die Vielfalt der Test-Szenarien im Check-out beachtet werden, darunter:

- Sprachversionen des Storefronts, z. B. Deutsch, Englisch

- Unterschiedliche Nutzerstati, z.B. Gast, eingeloggter B2C-Kunde, B2B-Kunde mit gültiger VAT-ID
- Verschiedene Versandmethoden, z. B. Abholung, Expressversand
- Zahlungsoptionen, z. B. Kreditkarte, PayPal, Rechnungskauf
- Verschiedene Warenkorbkombinationen und Mengen, z.B. einfache Produkte, konfigurierbare Produkte, virtuelle Produkte

Wie man anhand der obigen Beschreibung erahnen kann, erfordern sorgfältig durchgeführte und dokumentierte Integrations- und E2E-Tests erhebliche personelle und zeitliche Ressourcen. Daher wird sich die Frage stellen, wer die Tests durchführen sollte. Integrationstests können in der Regel von den Entwicklern der jeweiligen Schnittstellen durchgeführt werden, z. B. den E-Commerce-Entwicklern oder ERP-/PIM-Entwicklern. Der Fokus liegt eher auf einer technischen Überprüfung, ob Daten korrekt übermittelt werden. E2E-Tests hingegen sollten von Personen durchgeführt werden, die den gesamten Geschäftsprozess kennen, z. B. E-Commerce-Managern, Content-Managern oder Product Managern. Der Schwerpunkt liegt beim Sicherstellen, dass der Prozess aus Nutzersicht reibungslos funktioniert.

Da sowohl Integrationstests als auch E2E-Tests umfangreich und iterativ sind, sollten sie fest im Projektzeitplan verankert werden. Rückfragen, Change Requests und erneute Testläufe können zusätzlichen Aufwände erfordern. Wichtig ist es, Pufferzeiten einzuplanen, um mögliche Verzögerungen aufzufangen.

Gerade E2E-Tests sind unverzichtbare Bestandteile eines E-Commerce-Replatforming-Projekts. Sie prüfen komplette Geschäftsprozesse aus einer ganzheitlichen Perspektive und stellen damit sicher, dass die Ziele und Erwartungen des Projekts erfüllt werden.

✓ CHECKLISTE E2E UND INTEGRATIONSTESTS

- Analog zum Smoketesting wurde eine Übersicht an Szenarien für Integrationstests und End-to-end Test erstellt. Diese basieren auf dem Integrationskonzept.
- Im Rahmen der Integrationstests werden relevante Szenarien berücksichtigt, wie bspw.:
  - Welche Systeme und welche Schnittstellen getestet werden
  - Ob die Mappings der Datenfelder richtig sind
  - Ob die Kommunikation im festgelegten Timing stattfindet
  - Ob Validierungen richtig implementiert sind
- Im Rahmen der End-to-end Tests werden Prozesse geprüft, wie bspw.
  - Die Produktanlage in einem PIM bis hin zur Veröffentlichen im Storefront
  - Kundenanlage vom Registierungsformular im Storefront bis hin zum E-Mail Versand über das CRM
  - Check-out und Fulfillment vom Bestellabschluss im Storefront über den Import im PSP und ERP bis zum E-Mail Versand von Rechnungen und Versandmeldungen aus dem ERP
- Die End-to-end Tests berücksichtigen unterschiedliche Kombinationen der Szenarien, bspw.
  - den Nutzerzustand (Gast oder eingeloggt)
  - Sprachversionen des Storefront
  - Versand- und Zahlungsmethoden im Check-out
  - Warenkorbkombinationen (unterschiedliche Produkttypen)
- Der Zeitrahmen für Integrations- und End-to-end Tests ist in der Projekt Timeline abgebildet.
- Der Abnahmeprozess ist dokumentiert.

5.7 Go-Live Agenda

Der Go-Live eines neuen E-Commerce-Systems ist der entscheidende Moment. Eine klar definierte und abgestimmte Go-Live-Agenda sorgt dafür, dass alle beteiligten Akteure ihre Aufgaben kennen und die Schritte in der richtigen Reihenfolge durchgeführt werden. Ziel ist es, mögliche Fehler zu minimieren und einen reibungslosen Übergang sicherzustellen. Daher sollte die Go-Live Agenda mit entsprechendem Vorlauf erstellt und mit allen beteiligten Akteuren besprochen werden. Auch sollten alle Akteure ihre Schritte vorab prüfen und bestätigen, dass alle Anweisungen verständlich und machbar sind. Bspw. sollte das Domainhosting prüfen, ob die notwendigen DNS Einträge gesetzt werden können.

Bevor die eigentliche Go-Live Agenda entworfen wird, sollte eine Liste aller am Go-Live beteiligten Akteure erstellt werden. Hier sollte neben den Kontaktdaten auch der Verantwortungsbereich in Bezug auf das Replatforming Projekt genannt sein. Im Folgenden wird eine beispielhafte Liste gezeigt, die in der Praxis üblich ist:

AKTEUR UND KONTAKT	ROLLE UND VERANTWORTUNGSBEREICH
Neue E-Commerce Agentur / Ops	Server Infrastruktur, Hosting des neuen E-Commerce Systems
Neue E-Commerce Agentur / Devs	E-Commerce System Entwickler, Implementierung neues E-Commerce System
Neue E-Commerce Agentur / PO	Product Owner neues E-Commerce System, Product Backlog Management
Shop GmbH	E-Commerce Owner, Product Manager
ABC ERP Dienstleister	ERP Dienstleister, Integration ERP und E-Commerce System
Alte E-Commerce Agentur	E-Commerce Entwickler, Administration des alten Produktivsystems
Maxcluster	Altes E-Commerce System Hosting
GoDaddy	Domainhosting der relevanten E-Commerce Domains

Mithilfe der Kontaktdaten der beteiligten Akteure können die beteiligten Personen im Notfall erreicht werden. Zusätzlich kann es sich als hilfreich erweisen, einen Echtzeitchat in Microsoft Teams, Slack oder einem anderen Chat-Tool einzurichten. Im besten Fall sind in der obigen Tabelle ebenfalls die IP-Adressen vermerkt, sodass die Akteure auf das Storefront vom neuen E-Commerce System vor der Veröffentlichung zugreifen können (ansonsten wird für öffentliche Nutzer eine Wartungsseite angezeigt).

Sofern die beteiligten Akteure identifiziert sind, kann die Agenda entworfen werden. Die untere Tabelle zeigt einen in der Praxis üblichen Go-Live Ablauf inkl. Kommentare:

#	TIMING	AUFGABE	AKTEUR	BESCHREIBUNG	KOMMENTAR
0	Im Projektverlauf	Pre-prod Konfiguration	Neue E-Commerce Agentur	Konfiguration vom neuen E-Commerce System, darin Extensions und Integrationen, z.B. PIM, PSP, ERP Integration, externe Systeme	Alle integrierten Systeme sollten vorkonfiguriert sein, vorzugsweise im Sandbox Modus
0	Ca. T-1 Woche	Anpassung der Infrastruktur	Neue E-Commerce Agentur / Ops	Vorbereitung der Infrastruktur auf produktive Bedingungen hinsichtlich Traffic und Transaktionen	Im besten Fall sollte vor dem Go-Live ein Lastentest durchgeführt werden Die nötigen Informationen kommen aus Analytics oder dem vorherigen E-Commerce Hosting
0	Ca. T-1 Tag	Wartungsseite im Pre-prod Storefront aktivieren	Neue E-Commerce Agentur / Ops	Im Storefront wird eine Wartungsseite vom neuen E-Commerce System ausgespielt	Es sollten lediglich bestimmte Nutzer per IP Allowlisting auf das Pre-prod Storefront zugreifen können, um die Funktionalität nach dem Domain Umzug zu testen
1.1	T 08:00	Zahlungs- und Versandmethoden deaktivieren	Alte Agentur	Versand- und Zahlungsmethoden im alten Produktiv deaktivieren	Öffentliche Nutzer können keine neuen Bestellungen tätigen

1.2	T 08:00	Wartungsseite im alten produktiv Shop aktivieren	Alte Agentur / Hosting	Öffentliche Nutzer sehen eine Wartungsseite und können keine Interaktionen, z.B. Änderungen am Kundenkonto, vornehmen	Es wird sichergestellt, dass in alten Shop keine Daten geändert werden
1.3	T 08:15	Bestellungen exportieren und Order Status aktualisieren	ERP Agentur	Aktuelle Bestellungen werden an das ERP exportiert und der Order Status wird angepasst	Es wird sichergestellt, dass das ERP die aktuellsten Bestelldaten erhält
2.1	T 08:30	Integrationen und Kommunikation deaktivieren	Alte E-Commerce Agentur	Alte Shop-Bestellungen können nicht mehr an das ERP exportiert werden. Der alte Shop importiert keine Bestände und Sendungen mehr. Marketing-Tools sind nicht mit dem vorherigen Projekt des alten Shops integriert Der Zahlungsverkehr im alten Shop ist deaktiviert. Der alte Shop versendet keine E-Mails mehr.	Der alte Shop kommuniziert nicht mehr mit externen Systemen oder Kunden, und daher werden keine Daten in angeschlossenen Systemen geändert
2.2	T 09:00	Datenbank Dump vom alten Shop zur Verfügung stellen	Altes Hosting	Die aktuellen Bestellungen, Kunden, Produktbewertungen, Coupon Codes werden im Datenbank Dump zur Verfügung gestellt	Dieser wird benötigt, um die Datenmigration in den neuen Shop vorzubereiten
2.3	T 09:30	Datenbank Dump migrieren	Neue E-Commerce Agentur / Dev	Der Datenbank Dump wird in den neuen Shop migriert	Alle bis zur Aktivierung der Wartungsseite geänderten Daten werden gemäß dem Migrationskonzept in das neuen neue E-Commerce System migriert

3.1	T 10:30	DNS Einträge setzen	Domain-hosting	Im Domainhosting der Shop relevanten Domains sollten DNS Einträge (A Records) gesetzt werden, damit die finale Shop URL öffentliche Nutzer auf das neue E-Commerce Storefront leitet	Hinweis: dieser Schritt ist notwendig, sofern eine bereits existierende Shop URL für den neuen Shop übernommen wird. Falls für den neuen Shop eine neue URL genutzt wird (im Internet noch nicht bekannt), kann dieser Schritt bereits vor Schritt 1 getätigt werden
3.2	T 10:45	Domain bzw. URLs im neuen Shop ändern	Neue E-Commerce Agentur / Ops	Im neuen E-Commerce System bzw. der Infrastruktur sollten die finalen Domains und URLs eingestellt werden	Im Anschluss werden öffentliche Nutzer die Wartungsseite vom neuen E-Commerce System einsehen können. Nur bestimmte Nutzer (z.B. die E-Commerce Manager und die Entwickler) können per IP Allowlisting auf das neue Storefront zugreifen.
3.3	T 11:00	Externe Systeme und Extensions in Produktiv Modus schalten sowie URLs anpassen	Neue E-Commerce Agentur	Im neuen E-Commerce System werden z.B. PSP, PIM, Marketing Automation, usw. in den Produktiv Modus konfiguriert. Falls externe Systeme eine Shop-URL benötigen, sollte diese ebenfalls konfiguriert werden[53]	Der neue Shop kommuniziert mit angeschlossenen Systemen im Produktivmodus
3.4	T 11:15	ERP im Produktiv modus anschließen	ERP Agentur	Da sich die URL vom neuen E-Commerce System von der pre-prod URL in die finale URL geändert hat, muss der API Endpoint der Shop API Endpoints im ERP angepasst werden (und ggf. der API	Der neue Shop kommuniziert mit dem ERP im Produktivmodus und erhält die aktuellen Bestände, tauscht Bestellungen, Rechnung sowie Versandmeldungen aus

				User). Das ERP sollte die aktuellen Bestände an den Shop melden	
4	T 11:30	Smoketesting vor der Veröffentlichung	Neue E-Commerce Agentur, Product Manager sowie relevante Stakeholder	Prüfung der kritischen Geschäftsprozesse, darunter Check-out und PSP Integration ERP Integration PIM Integration E-Mail Versand ...	Es sollte geprüft werden, ob alle kritischen Prozesse und Integrationen gemäß den definierten Szenarien aus dem Smoketesting und E2E Testing funktionieren. Da externe Systeme mit einer neuen URL bzw. Domain kommunizieren, sollten vor allem die Integrationen überprüft werden.
5.1	T 13:00	Veröffentlichung	Neue E-Commerce Agentur / Ops	Die Wartungsseite wird deaktiviert, womit alle öffentlichen Nutzer auf den neuen Shop zugreifen können	Der Go-Live hat stattgefunden und öffentliche Nutzer sehen das neue E-Commerce Storefront
5.2	T 13:15	Sitemap & robots.txt regenerieren	Neue E-Commerce Agentur / PO	Die Sitemap und robots.txt werden im neuen Shop mit den finalen URLs regeneriert. Die Sitemap kann in der Google Search Console hinterlegt werden	Vor diesem Schritt wurden die Pre-prod URLs in der Sitemap genutzt
5.3	T 13:30	301 Redirects setzen	Neue E-Commerce Agentur / Devs	Alte Shop URLs führen auf die URLs vom neuen E-Commerce System	Die 301 Redirects werden gemäß den Vorgaben vom SEO Konzept gesetzt Die SEO Performance von indexierten Seiten wird in den neuen Shop übernommen
6	T 15:00	Monitoring aufsetzen	Neue E-Commerce	Für das neue E-Commerce System	Im Falle von Downtimes oder

	Agentur / Ops	sollte ein Monitoring von relevanten technischen KPIs, z.B. CPU oder RAM Auslastung, sowie der Uptime konfiguriert werden	Performance-Probleme sollten bestimmte Nutzer informiert werden, sodass die Probleme zeitnah behoben werden können

Mit dem E-Commerce Owner sollte man zudem klären, ob das alte E-Commerce System weiterhin für bestimmte Nutzer zur Verfügung stehen sollte. Damit kann man im Nachgang alte Features einsehen, sowie die Datenmigration prüfen. Wichtig dabei ist, das alte E-Commerce System vor dem Zugang öffentlicher Nutzer, z.B. mittels Basic Auth oder IP Allowlisting, zu schützen.

✓ CHECKLISTE GO-LIVE AGENDA

- Alle für den Go-Live relevanten Akteure sind bekannt. Die Kontaktdaten und die Rollen sind festgehalten.
- Es wurde eine Go-Live Agenda erstellt und von allen beteiligten Akteuren akzeptiert.
- Für den Go-Live wird ein Echtzeit Chat eingerichtet, damit die Akteure durchgängig informiert sind und kommunizieren können.
- Die Wartungsseite sowie der Zugriff auf das Storefront wurde vor dem Go-Live getestet.
- Die Go-Live Agenda enthält alle für den Go-Live kritischen Schritte und den chronologischen Ablauf, darunter etwa
  - Aktivierung einer Wartungsseite
  - Datenmigration
  - DNS Records (Domain Wechsel auf die produktive URL) und deren Timing
  - Anpassung der externen Schnittstellen an die finale URL (z.B. API Endpoints beim PSP, ERP, PIM oder CRM)
  - Abschalten des alten Systems
  - Smoketesting vor der Veröffentlichung

5.8 Nach dem Go-Live

Nach dem erfolgreichen Go-Live beginnt eine kritische Phase, in der Stabilität, Performance und die korrekte Funktionsweise des neuen E-Commerce-Systems überwacht werden müssen. Zunächst ist sicherzustellen, dass das technische Monitoring korrekt eingerichtet ist – hierzu zählen die Überwachung der allgemeinen Verfügbarkeit (Uptime), der System-Performance (z. B. RAM- und CPU-Auslastung) sowie der relevanten API-Endpunkte.

Parallel dazu müssen alle Marketing- und SEO-relevanten Tools und Prozesse verlässlich arbeiten. Die Google Search Console sollte neue Inhalte indexieren können, während Google Ads, Merchant Center und weitere Marketingtools mit aktuellen Daten versorgt werden. Auch das Tracking via Google Tag Manager und Analytics muss fehlerfrei funktionieren. Ein laufendes SEO-Crawling hilft dabei, etwaige Probleme wie fehlerhafte Redirects oder nicht indexierte Seiten frühzeitig zu erkennen.

Für eine strukturierte Weiterarbeit sind klare Verantwortlichkeiten entscheidend: Wer priorisiert Bugs? Welche Workflows gelten für das Projektmanagement? In welchen Formaten werden Abstimmungen oder Stand-ups durchgeführt? Ebenso muss geklärt sein, wer den operativen Support übernimmt – ein dediziertes Hypercare-Team, das bestehende Umsetzungsteam oder ein neu aufgestelltes Wartungsteam.

Wichtig ist zudem, einen verbindlichen Rahmen für künftige Releases zu definieren – etwa, ob Deployments kontinuierlich (Continuous Delivery) oder on Demand erfolgen. Damit die Weiterentwicklung nicht ins Stocken gerät, sollte bereits jetzt ein Budgetrahmen definiert sein, auf dessen Basis eine Roadmap mit realistischen Zeitplänen und messbaren Zielen (z. B. KPIs) entstehen kann.

5.9 Dos und Don'ts im Testing und Go-Live

Die Testing- und Go-Live-Phase ist entscheidend für einen erfolgreichen Start der neuen E-Commerce-Plattform. Es gibt jedoch typische Stolpersteine, die vermieden werden können. Hier sind wichtige Dos und Don'ts, um häufige Fehler zu umgehen.

Vor dem Go-Live sollte ein umfassender SEO Crawl durchgeführt werden, da die **SEO auf den Gesamterfolg nicht unterschätzt** werden kann. Man sollte das neue System auf die Grundlagen, also das 301-Mapping, externe Marketing-Tools mit URL-Verweisen und Meta-Daten, prüfen. Ein SEO Ranking-Verlust ist schwer zu reparieren und kann Monate dauern.

Die **Projektdokumentation sollte stets aktuell gehalten** werden, und es sollten möglichst viele Details wie etwa Mappings, Workarounds, Konfiguration usw. festgehalten sein. Eine detaillierte Projektdokumentation mit Grafiken und Diagrammen, insbesondere bei Integrationen und der Systemlandschaft, erleichtern das Debugging und helfen bei der Zusammenarbeit in größeren Teams. Eine oberflächliche oder nicht aktuelle Projektdokumentation führt zu Missverständnissen, vor allem bei komplexen Integrationen. Fehlt der Überblick, dauert die Fehlerbehebung länger, was den Projekterfolg gefährden kann.

Des Weiteren ist es essenziell, **Rollen und Verantwortlichkeiten beim Testing klären.** Man sollte genauestens definieren, wer für welche Tests verantwortlich ist. Gerade für Integrations- und E2E-Tests sollten ausreichend personelle Ressourcen geplant werden. In jedem komplexeren Projekt werden dabei Bugs und Change Requests entstehen, bei denen gilt: Corner-Cases haben eine niedrigere Priorität als Fehler, die einen größeren Nutzerkreis betreffen.

Die **Go-Live Agenda sollte klar** sein und mit allen Beteiligten im Voraus besprochen werden. Dabei muss sichergestellt werden, dass Action Items realistisch und machbar sind. Themen wie der Zugang zum Domainhosting oder Berechtigungen für den DNS-Wechsel sollten rechtzeitig geklärt werden. Falls in der Agenda Unklarheiten oder Rückfragen entstehen, muss man diese zwingend erklären. Missverständnisse während des Go-Lives können zu kritischen Fehlern führen, die leicht im Vorfeld vermieden werden können.

Nach dem Go-Live gilt: **Nicht in Panik geraten**. Es ist normal, dass nach dem Go-Live kleinere Bugs auftreten oder Features (noch) nicht wie gewünscht funktionieren. Statt

sich von jedem Problem verrückt machen zu lassen, sollten klare Prozesse für Bugfixing und Priorisierung etabliert werden. Daher sollte nach dem Go-Live eine sogenannte „Hypercare“-Phase geplant werden, in der Support-Ressourcen bereitstehen, um dringende Bugs oder Anpassungen zu beheben. Die Support Kontingente und Verantwortlichkeiten sollten ebenfalls im Vorfeld geklärt werden.

KAPITEL 6

SCHLUSSWORT

In diesem Buch wurden die zentralen Phasen eines E-Commerce-Replatforming erörtert:

- In der Entscheidungsphase wurde beleuchtet, wie man Signale für ein Replatforming erkennt, Ziele und Erwartungen definiert und die richtige Plattform sowie den passenden Partner auswählt.
- Die Analysephase zeigte die Wichtigkeit, den Ist-Zustand technologisch, funktional und aus Business-Perspektive zu verstehen, Stakeholder zu analysieren und Wettbewerber sowie Zielgruppen genauer zu betrachten.
- In der Konzeptions- und Umsetzungsphase standen Projektmanagement, die Storefront-Konzeption, die Datenmigration, Integrationen und die QA im Fokus – all dies als Basis für eine erfolgreiche Implementierung.
- Abschließend wurde im Kapitel Testing und Go-Live darüber gesprochen, wie die letzten Schritte strukturiert und sorgfältig geplant werden, um das neue System stabil und ohne größere Risiken live zu bringen.

Ein Replatforming sollte nicht als notwendiges Übel, sondern als Chance gesehen werden: eine Gelegenheit, den Umsatz zu steigern, Prozesse effizienter zu gestalten, die User Experience zu verbessern, Mitarbeiter zufriedener zu machen und langfristig Kosten zu senken. Gerade in dynamischen Marktbedingungen und wirtschaftlich herausfordernden Zeiten werden Mitbewerber zurückhaltend agieren. Daher können kluge Investitionen die Möglichkeit bieten, Marktanteile zu gewinnen und diese durch smarte Features und eine exzellente Kundenbindung zu sichern.

Dennoch bringt ein Replatforming Herausforderungen mit sich: Fehler in jeder Projektphase können kostspielig sein. Dieses Vorhaben erfordert umfassende Expertise, Erfahrung und personelle Ressourcen. Wenn Kapazitäten oder spezifisches Know-how fehlen, sollte die Unterstützung durch Experten in Betracht gezogen werden. Oftmals sind die Kosten für externe Partner niedriger als die Folgen grober Fehleinschätzungen.

ÜBER DEN AUTOR

Karol Schoensee ist E-Commerce Consultant bei der creativestyle GmbH in München und seit über zehn Jahren in der digitalen Wirtschaft zu Hause. Sein Schwerpunkt liegt auf **komplexen Migrationen und Replatforming-Projekten** — vom ersten Audit der bestehenden Plattform über die Auswahl des passenden Shop-Systems bis zur Konzeption und Begleitung der Umsetzung.

Karol kam 2017 zu creativestyle, wo er sieben Jahre lang als **Product Owner** zahlreiche Magento-2-, Shopware- und Headless-Commerce-Projekte im deutschsprachigen Raum verantwortet hat. In dieser Zeit hat er auch **MageSuite.io** mitgeprägt — das von creativestyle entwickelte Open-Source-Modul-Ökosystem für Magento 2, das mittlerweile branchenweit eingesetzt wird.

Seit 2024 ist er als **E-Commerce Consultant** bei creativestyle für die strategische Beratung von Händlern verantwortlich: Anforderungsanalyse, Lösungsarchitektur, Plattformauswahl sowie der langfristige Aufbau von Kundenbeziehungen entlang des gesamten Replatforming-Lebenszyklus.

Als Consultant berät Karol Händler bei den Fragen, die diesem Buch zugrunde liegen: **Wann lohnt sich ein Replatforming wirklich? Welche Plattform passt zum Geschäftsmodell? Wie minimiert man die Risiken einer Migration?** Seine Projekte bewegen sich typischerweise an der Schnittstelle zwischen Adobe Commerce (Magento 2), Shopware, commercetools und Headless-Architekturen — ergänzt um Integrationen in ERP-, PIM- und CMS-Systeme.

Im Laufe seiner Karriere ist er zu einem ausgesprochenen Enthusiasten für **Open-Source-Technologien, B2B-Commerce und Projektmanagement** geworden — drei Themen, über die er auch ausführlich in seinem Blog **e-commercianer.io** schreibt.

ANHANG

FUSSNOTEN & QUELLEN

1 z.B. Google PageSpeed Insights oder New Relic

2 z.B. Pingdom oder Uptrends

3 z.B. Sentry

4 z.B. SEMrush, Sistrix oder Searchmetrics

5 z.B. Zendesk

6 z.B. JIRA oder Asana

7 Siehe https://www.cloudflare.com/de-de/

8 Siehe https://www.akeneo.com/

9 Siehe https://www.storyblok.com/

10 Siehe https://www.algolia.com/de

11 Siehe https://slack.com/intl/de-de

12 Siehe https://developers.google.com/search/docs/appearance/core-web-vitals?hl=de

13 Siehe https://eur-lex.europa.eu/eli/dir/2019/2161/oj?locale=de

14 Bei einer Einfachauswahl besteht eine 1 zu 1 Beziehung, z.B. hat ein Schuh genau eine Schuhgröße

15 Bei einer Mehrfachauswahl besteht eine 1 zu N Beziehung, z.B. besteht ein T-Shirt aus mehreren Materialien

16 "Pullover" liefert die gleichen Ergebnisse wie "Sweatshirt", und umgekehrt

17 "Unterhemd" liefert die gleichen Ergebnisse wie "Shirts", aber nicht umgekehrt

18 Damit ist der letzte Bereich der URL nach dem "/" gemeint, wie etwa "shop.com/url-schuessel"

19 Mit nofollow und noindex Meta Tags kann die Indexierung von Inhalten für Suchmaschinen gesteuert werden, siehe https://developers.google.com/search/docs/crawling-indexing/block-indexing?hl=de

20 Für mehr Details zu robot.txt Datei siehe https://developers.google.com/search/docs/crawling-indexing/robots/intro?hl=de

21 Für weiterführende Informationen zur Sitemap siehe https://developers.google.com/search/docs/crawling-indexing/sitemaps/overview?hl=de

22 Mit strukturierten Daten erhalten Suchmaschinen, wie etwa Google oder Bing, indexierte Inhalte in einem vordefinierten Format. Dies verbessert die Indexierung der Inhalte und damit die SEO Relevanz. Siehe https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data?hl=de

23 Mit OpenGraph Meta Tags wird die Sichtbarkeit von geteilten Inhalten auf sozialen Medien, wie Facebook, verbessert. Siehe https://developers.facebook.com/docs/sharing/webmasters/#markup

24 Im Zuge eines Replatforming Projekts können sich die URLs zwischen dem alten Shop und dem neuen Shop unterscheiden, weil etwa eine andere Logik zur Generierung der URL Schlüssel verwendet wird. In diesem Fall ist es ratsam, mithilfe von 301 Redirects Suchmaschinen von den existierenden (und indexierten URLs) auf die neuen Shop-URLs zu führen und damit die SEO Relevanz zu übernehmen. Siehe https://developers.google.com/search/docs/crawling-indexing/301-redirects?hl=de

25 Google Analytics (https://developers.google.com/analytics?hl=de) wird für das Sammeln von Daten zum Nutzerverhalten, wie etwa den Conversions, Sessions, Pageviews, usw. genutzt. Der Google Tag Manager (https://marketingplatform.google.com/intl/de/about/tag-manager/) wird zur Erstellung von

Event-Trackings, z.B. einem Download, sowie zur Integration mit Marketing Pixeln genutzt. Hiermit können etwa Conversions über eine Preissuchmaschine oder andere Marketing-Kanäle festgehalten werden.

26 Attribute haben in verschiedenen System grundsätzlich andere Benennungen und Attributcodes. Im PIM kann bspw. der Produktname "product_name" heißen, im E-Commerce System "name". Unter dem Attributmapping versteht man also die Zuordnung der Attribute zwischen den unterschiedlichen Systemen.

27 Der URL Request Header wird in der entsprechenden Schnittstellenimplementierung Auskunft darüber geben, ob ein PULL oder PUSH Verfahren genutzt wird. Falls das System einen "HTTP-GET" Request sendet, werden Daten abgerufen (PULL). Falls ein "HTTP-POST" Request gesendet wird, werden Daten übermittelt (PUSH).

28 Mit "Bedarf" ist gemeint, dass eine Aktion entweder im E-Commerce System oder im PIM die Kommunikation auslöst. Auf der PIM Seite könnte dies etwa die Freischaltung eines Produktes sein. Auf der Seite des E-Commerce Systems könnte dies etwa per Knopfdruck passieren.

29 Das Attributmapping im ERP Kontext beinhaltet natürlich eher ERP relevante Felder, wie z.B. Adressfelder bei einer Bestellung, enthaltene Produkte, Zahlungsmethoden, Rabattpositionen usw. Bspw. kann die im E-Commerce System interne Bezeichnung für Kreditkarten "CreditCard" heißen, und im ERP "credit_card".

30 Eine Warenkorbpreisregel könnte lauten: füge von einem bestimmten Produkt die Anzahl X in den Warenkorb, und erhalte weitere Y gratis. Oder füge Produkte im Warenkorbwert von A € hinzu, und erhalte das Produkt B als Geschenk. Statt Produkte können auch prozentuale oder absolute Rabatte in € als Geschenk denkbar sein. Aufgrund der Vielzahl an Bedingungen und möglichen Rabattausprägungen kann eine solche Integration zwischen E-Commerce System und CRM sehr komplex und zeitaufwendig werden.

31 Dies ist der Fall für die in Deutschland prominenten PSP, wie Adyen, Mollie, Payone, Stripe, Klarna, Paypal, usw.

32 Unter Personalkosten sind interne Kosten für Mitarbeiter, also Löhne gemeint. Denkbar sind z.B. Aufwände für Content-Manager, Online-Marketing Mitarbeiter, usw. Agenturkosten entstehen in der Regel nur, falls eine externe Agentur den Shop betreut oder für relevante Systeme, wie das PIM, das ERP oder Online-Marketing, zuständig ist.

33 Siehe https://www.sistrix.de/ und https://de.semrush.com/

34 Siehe https://matomo.org/

35 Siehe https://www.hotjar.com/de/

36 Siehe https://www.figma.com/de-de/ und https://adobexdplatform.com/

37 Siehe https://baymard.com/ und das Smashing Magazine als UX Blog Empfehlung: https://www.smashingmagazine.com/

38 Siehe https://www.barrierefreiheit-dienstekonsolidierung.bund.de/Webs/PB/DE/gesetze-und-richtlinien/barrierefreiheitsstaerkungsgesetz/barrierefreiheitsstaerkungsgesetz-node.html

39 Siehe https://maxcluster.de/

40 Auf der folgenden Seite können die Marktanteile pro Browser und Version eingesehen werden: https://caniuse.com/usage-table

41 Siehe https://www.testrail.com/

42 Siehe https://playwright.dev/

43 Siehe https://validator.schema.org/

44 In der Praxis bewährte Crawler sind bspw. https://de.ryte.com/ und https://www.screamingfrog.co.uk/seo-spider/

45 Siehe https://www.seo-trainee.de/linkjuice-gezielt-nutzen-linkmaskierung-via-prg-pattern-einfach-erklaert/

46 Siehe https://developers.google.com/search/docs/crawling-indexing/301-redirects?hl=de

47 Bspw. wird bei Google Ads die GCLID als URL Parameter berücksichtigt, siehe https://support.google.com/google-ads/answer/9744275?hl=de

48 Mehr Informationen zu den beiden Ansätzen findet man in der Google Dokumentation: https://support.google.com/tagmanager/answer/7582054?hl=de

49 Siehe die Dokumentation:
https://support.google.com/tagmanager/answer/6107056?hl=de&sjid=18139630263125072541-EU

50 Siehe https://www.atlassian.com/software/confluence

51 Viele gängige PSP Anbieter erlauben die Nutzung einer Testkreditkarte, die nur im Testmodus funktioniert. Mit dieser können unterschiedliche Szenarien der Kreditkartenzahlung, wie “abgelehnt” oder “bezahlt”, simuliert werden.

52 Showstopper sind kritische Bugs, die wesentliche E-Commerce Funktionen beeinträchtigen. Bei Vorliegen eines Showstoppers sollte das Deployment im Zweifel nicht durchgeführt werden.

53 Oftmals muss die Check-out URL eines E-Commerce im PSP hinterlegt werden.