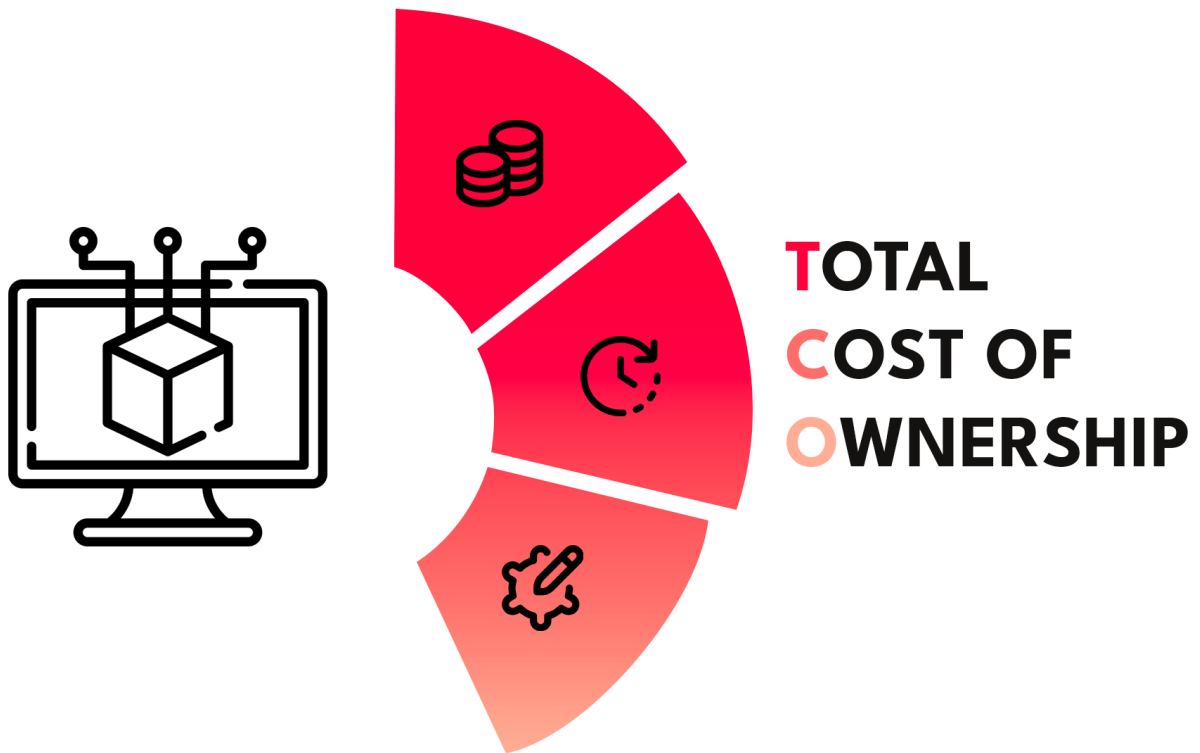




SOFTWARE-KOSTENCHECK: BUILD VS. BUY

Kosten sparen beim Software-Einkauf

KOSTEN RICHTIG KALKULIEREN UND VERGLEICHEN

Einleitung

Unternehmen stehen heute vor der ständigen Herausforderung, neue **Softwarelösungen in ihre Geschäftsprozesse zu integrieren** – sei es zur Optimierung bestehender Abläufe oder zur Einführung neuer Services und Produkte.

Oft existieren bereits Softwarelösungen, die jedoch nicht mehr zu den aktuellen Prozessen passen, wirtschaftlich nicht mehr tragbar sind oder nicht mehr mit der strategischen Ausrichtung des Unternehmens harmonisieren. In anderen Fällen steht man noch ganz am Anfang und sucht nach

einer effizienteren, automatisierten Lösung für bisher manuelle Prozesse.

Bei der Entscheidung für eine neue Software spielen zahlreiche Faktoren eine Rolle – einer der ausschlaggebendsten ist aber immer die Kostenfrage. Dieses Whitepaper beschäftigt sich ausführlich mit der **Berechnung der Total Cost of Ownership (TCO)** einer Software und zeigt praxisnah, wie diese ermittelt werden können – inklusive eines konkreten Anwendungsbeispiels.



INHALTSVERZEICHNIS

Kapitel 1: Wesentliche Schritte vor einer Kostenbestimmung	01
Kapitel 2: Kostenbestimmung bei Software (TCO)	02
Kapitel 3: TCO - Kosten in den drei Lebensphasen einer Software	04
Kapitel 4: Berechnung der Gesamt-TCO-Kosten	08
Kapitel 5: Praxisbeispiel: Kosten Build vs. Buy	12
Kapitel 6: Bilanzielle Betrachtung der Softwarekosten	14
Fazit	15

KAPITEL 1:

WESENTLICHE SCHRITTE VOR EINER KOSTENBESTIMMUNG

Bevor die Kosten einer Software berechnet werden, gibt es in der Regel einige vorbereitende Schritte, die eine fundierte und sinnvolle Kostenbestimmung ermöglichen.

Da wir auf diese Schritte hier nicht detailliert eingehen, verweisen wir für eine vertiefte Auseinandersetzung auf unser weiteres Whitepaper ([Individualsoftware oder Standardsoftware](#)). Der Vollständigkeit halber sollen sie jedoch an dieser Stelle kurz erwähnt werden:



Schritt 1

1

Bedarf und Ziele bestimmen: Der erste Schritt auf dem Weg zur neuen Software besteht darin, den aktuellen Status quo genau zu analysieren und die Zielsetzung, was mit der neuen Software erreicht werden soll, klar zu definieren.

Schritt 2

2

Anforderungen definieren und priorisieren: Im zweiten Schritt sollte eine klare Liste mit funktionalen und nicht-funktionalen Anforderungen an die Software erstellt und entsprechend priorisiert werden.

Schritt 3

3

Build-or-Buy-Entscheidung: Vor der Kostenanalyse sollte nun noch geprüft werden, ob es bereits klare Indikatoren oder Kriterien gibt, die für eine Eigenentwicklung (Build) der Software oder den Einsatz einer Standardsoftware (Buy) sprechen. Falls diese nicht eindeutig sind, empfiehlt es sich, die Kosten für beide Optionen – einschließlich möglicher Hybrid-Varianten – systematisch zu analysieren.

KAPITEL 2:

KOSTENBESTIMMUNG BEI SOFTWARE (TCO)

Häufig passiert es, dass sich Unternehmen entweder nur mit den initialen Kosten für neue Software auseinandersetzen oder nur die offensichtlichen "direkten" Kosten einer Software einbeziehen. Viele weitere Kosten, die langfristig den wahren Preis einer Softwarelösung ausmachen, werden dabei vernachlässigt. **Die TCO (Total Cost of Ownership) umfassen alle Kosten**, die während des gesamten Lebenszyklus einer Software anfallen – von Anschaffung oder Entwicklung über Betrieb bis hin zur Wartung. Dabei zählen nicht nur offensichtliche Kosten wie Kaufpreis oder Entwicklung, sondern auch versteckte oder indirekte Ausgaben.

Wichtig dabei ist:

Berücksichtigung der direkten Kosten über die geplante Laufzeit

Vor allem komplexe und umfangreiche Softwarelösungen werden oft über viele Jahre hinweg genutzt. Für eine Gesamtkostenbetrachtung sollten daher bei einer Kauf-Lösung die Lizenzkosten für die geplante Nutzungsdauer berücksichtigt werden, während bei einer Eigenentwicklung die initialen Kosten sowie die voraussichtlichen zukünftigen Anpassungskosten einbezogen werden sollten.

Berücksichtigung aller indirekten/nicht offensichtlichen Kosten

Indirekte Kosten umfassen Support, Wartung, Schulungen, Infrastrukturkosten sowie mögliche Integrations- und Anpassungsausgaben. Bei Standardsoftware sollten zusätzlich nutzungsabhängige Kosten berücksichtigt werden, wie etwa Gebühren pro Benutzer, zusätzliche Kosten für Module oder APIs sowie potenzielle Preissteigerungen bei wachsender Benutzerzahl. Bei Individualsoftware gilt es zudem, die Ausgaben für regelmäßige Updates und Erweiterungen um neue Funktionalitäten einzuplanen.



Zur näherungsweisen Bestimmung der Total Cost of Ownership (TCO) einer Software untergliedern wir deren Lebenszyklus in **drei Phasen**:

1**Initiale Kosten**

Die anfänglichen Ausgaben für die Entwicklung oder Anschaffung sowie die Implementierung der Software.

2**Kosten für Wartung & Support**

Laufende Kosten für die Pflege, Fehlerbehebung und regelmäßige Aktualisierungen der Software.

3**Kosten für Erweiterung & Skalierung**

Kosten, die entstehen, wenn die Software an neue Anforderungen angepasst oder für ein wachsendes Unternehmen skaliert wird.

KAPITEL 3:

TCO-KOSTEN IN DEN DREI LEBENSPHASEN EINER SOFTWARE

Phase 1: Initiale Kosten – Die Grundlagen legen

Der anfängliche Aufwand einer "Build"-Entscheidung, also der individuellen Entwicklung einer Software, ist offensichtlich deutlich höher als die Nutzung einer bereits bestehenden Lösung. Dennoch ist es auch bei einer lizenzbasierten Software wichtig zu prüfen, welche Kosten für die Einführung tatsächlich anfallen.

- Die Kosten der **Buy-Option (Standardsoftware)** in der initialen Phase umfassen:

Lizenzgebühren 1

In der Regel basiert die Kostenstruktur von Standardsoftware auf Abonnements, die monatlich oder jährlich anfallen. Diese Gebühren können je nach Anbieter und Funktionsumfang stark variieren. Sie richten sich häufig nach der Zahl der Nutzenden, dem Funktionsumfang, der Größe des Unternehmens oder der Länge der abgeschlossenen Mindestlaufzeit.

Add-ons und Erweiterungen 2

Nicht vergessen! Viele Buy-Lösungen bieten zusätzliche Module oder Funktionalitäten gegen Aufpreis an, die oft notwendig sind, um den individuellen Anforderungen eines Unternehmens gerecht zu werden.

Implementierung und Einrichtung: 3

Die Einführung einer Standardsoftware kann Implementierungsgebühren mit sich bringen, beispielsweise für das Setup, die Datenmigration oder die Integration in bestehende Systeme. Die meiste Standardsoftware muss in bestehende Unternehmenssysteme integriert werden. Werden Schnittstellen zu Drittsystemen nicht angeboten oder werden weitere Anforderungen nicht erfüllt, erfordert dies in der Regel zusätzliche Entwicklungskosten für nötige Anpassungen der Software.

Runterzuberechnen in:

$$\text{Initiale Kosten (Buy)} = \text{Lizenzkosten} + \text{Implementierungskosten}$$

- Die Kosten der **Build-Option (Individualsoftware)** in der initialen Phase umfassen:

Entwicklungskosten

1

Der größte Kostentreiber einer "Build"-Lösung sind die Entwicklungsstunden, die oft auf Basis von Aufwand abgerechnet werden. Die Anzahl der Arbeitsstunden hängt maßgeblich von der Komplexität, der Anzahl der Anforderungen sowie den gewünschten Features der Software ab.

Anpassungs- und Integrationskosten

2

Ähnlich einer Standardsoftware entstehen auch bei einer individuellen Software Implementierungskosten für die Datenmigration und Schulungen. Diese fallen in der Regel kleiner aus, da die Software schon mit Hinblick auf das Zielsystem entwickelt wird und spezifische Schnittstellenlösungen bereits berücksichtigt werden.

Runterzuberechnen in:

$$\text{Initiale Kosten (Build)} = \text{Entwicklungskosten} + \text{Anpassungskosten}$$



Phase 2: Betrieb, Wartung & Support – Stabilität sicherstellen

War in Phase 1 in der Regel noch ein Kostenvorteil zugunsten der Buy-Variante erkennbar, kehrt sich das Bild nun in der Betriebs- und Wartungsphase um. Hier kann die Individualsoftware ihre Vorteile gegenüber einer Standardsoftware ausspielen, da keine laufenden Lizenzkosten anfallen.

- Die Kosten der **Buy-Option (Standardsoftware)** in der Betriebs-Phase umfassen:

1 Lizenzgebühren

Die Lizenzgebühren fallen während der gesamten Nutzungsdauer der jeweiligen Software in monatlichem oder jährlichem Rhythmus an. Außerdem muss mit regelmäßigen Erhöhungen gerechnet werden. Die Lizenzkosten für eine Standardsoftware decken in der Regel Wartung und regelmäßige Updates ab. Die Gebühren steigen jedoch oft mit der Anzahl der Nutzer oder zusätzlicher Funktionen.

2 Anpassungs- und Integrationskosten bei Änderungen

Standardsoftware erfordert oft Anpassungen, wenn sich die Anforderungen oder Schnittstellen der Umgebung ändern oder sich die Prozesse im Unternehmen ändern und diese in Software abgebildet werden müssen. Zu klären bleibt die Anpassbarkeit der Standardsoftware an sich.

Runterzuberechnen in:

$$\text{Laufende Kosten (Buy)} = \text{Lizenzkosten} + \text{Anpassungskosten}$$

- Die Kosten der **Build-Option (Individualsoftware)** in der Betriebs-Phase umfassen:

1 Server- und Hostingkosten

Individuelle Softwarelösungen erfordern immer einen selbständigen Betrieb und daher dedizierte Hosting-Infrastrukturen.

2 Wartungs- und Supportkosten

Nach der Bereitstellung der Software fallen laufende Kosten für Updates, Bugfixes und allgemeine Wartung an. Diese sind in der Regel abhängig von der Komplexität der Software und den Anforderungen des Unternehmens.

Runterzuberechnen in:

$$\text{Laufende Kosten (Build)} = \text{Hostingkosten} + \text{Wartungskosten}$$

Phase 3: Erweiterung & Skalierung – Mit den Anforderungen wachsen

Unternehmen verändern sich, wachsen und passen sich an neue Marktanforderungen an. Eine Softwarelösung muss in der Lage sein, diesen Wandel zu unterstützen. Die Fähigkeit zur Anpassung und Skalierung ist daher ein wichtiger Faktor in der langfristigen Nutzung von Software.

- Die Kosten der **Buy-Option (Standardsoftware)** in der Skalierungs-Phase umfassen:

1 Anstieg der Lizenzgebühren

Die Lizenzgebühren steigen in der Regel bei zusätzlichen Nutzungslizenzen oder kostenpflichtigen Add-ons.

3 Forcierte Updates

Viele Anbieter von Standardsoftware erzwingen bestimmte, meist größere Updates. Diese bringen oft erheblichen Anpassungsaufwand, potenzielle Inkompatibilitäten und zusätzlichen Schulungsbedarf mit sich.

2 Eingeschränkte Anpassungsfähigkeit

Wenn die Standardsoftware bestimmte Anforderungen nicht erfüllt, können ineffiziente Workarounds oder Drittanbieter-Lösungen erforderlich sein. Beide Optionen sind mit zusätzlichen Kosten verbunden.

Runterzuberechnen in:

$$\text{Skalierungskosten (Buy)} = \text{Lizenzkosten} + \text{Workaroundkosten}$$

- Die Kosten der **Build-Option (Individualsoftware)** in der Skalierungs-Phase umfassen:

1 Entwicklungskosten

Die Erweiterung bestehender Funktionalitäten erfordert zusätzliche Entwicklungszeit.

2 Server- und Hostingkosten

Wachstum führt oft zu einem erhöhten Bedarf an Serverkapazität, Datenbanken und anderen Infrastrukturanpassungen.

Runterzuberechnen in:

$$\text{Skalierungskosten (Build)} = \text{Entwicklungskosten} + \text{Hostingkosten}$$

KAPITEL 4:

BERECHNUNG DER GESAMT-TCO-KOSTEN

Eine Analyse der Kosten von Individual- und Standardsoftware über den gesamten Softwarelebenszyklus hinweg zeigt, dass die optimale Lösung stark von der jeweiligen Phase sowie den spezifischen Anforderungen des Unternehmens abhängt. Gleichzeitig ist es wichtig, sowohl die direkten als auch die indirekten Kosten einer Software zu berücksichtigen.

Die **Gesamtkosten** einer Software setzen sich damit zusammen aus:

$$\text{Gesamtkosten (TCO)} = \text{initiale Kosten} + \text{laufende Kosten} + \text{Skalierungskosten}$$

Nicht immer stehen jedoch alle Parameter zur Verfügung, die die Kosten einer Software in den verschiedenen Lebensphasen genau bestimmen. In solchen Fällen wäre eine Methode hilfreich, die es ermöglicht, die Gesamtkosten (TCO) einer Software näherungsweise zu schätzen, um eine erste Orientierung zu erhalten. Im Folgenden wird ein Ansatz hierfür vorgestellt.



Zur vereinfachten **Berechnung der TCO einer Buy-Lösung** (Standardsoftware) lässt sich mit folgender Formel arbeiten:

$$\text{Gesamtkosten (Buy)} = \sum_{t=1}^n (\text{Lizenzgebühren}_1 \times (1+r)^{t-1}) + \text{Anpassungskosten}$$

Dabei gilt:

- **n:** Geplante Nutzungsdauer in Jahren
- **t:** Jahr *t* der Nutzung
- **Lizenzgebühren:** Lizenzgebühren im ersten Jahr
- **r:** Gesamtrate der Preissteigerung verursacht durch eine Preiserhöhungen des Anbieters und zusätzliche Kosten durch Addons oder neue/andere Lizenzen, berechnet als $r = r_{\text{Preis}} + r_{\text{Addons}}$

Erklärung

Für eine näherungsweise Bestimmung der Gesamtkosten einer Buy-Lösung werden insbesondere die Lizenzkosten der jeweiligen Lösung berücksichtigt. Lizenzkosten entwickeln sich in der Regel über längere Zeiträume nicht linear. Sie müssen um Faktoren wie Preiserhöhungen durch die Anbieter sowie zusätzliche Module und Funktionen, die mit steigenden Lizenzkosten verbunden sind, angepasst werden. Zusätzlich fallen Implementierungs-, Onboarding- und Anpassungskosten an, die im Laufe der Nutzungszeit entstehen. Diese können häufig näherungsweise anhand der Lizenzkosten geschätzt werden, da komplexere Softwarelösungen mit höheren Lizenzkosten in der Regel auch einen größeren Anpassungsaufwand nach sich ziehen.

Zur vereinfachten **Berechnung der TCO einer Build-Lösung (Individualsoftware)** lässt sich mit folgender Formel arbeiten:

$$\text{Gesamtkosten (Build)} = \text{Initialkosten} + \sum_{t=1}^n (\text{Betriebskosten}_t \times (1+r)^{t-1}) + \text{Anpassungskosten}$$

Dabei gilt:

- **Initialkosten:** In der Regel Erstentwicklungszeit Tagessatz
- **n:** Geplante Nutzungsdauer in Jahren
- **t:** Jahr t der Nutzung
- **Betriebskosten₁:** Betriebs- und Wartungskosten im ersten Jahr
- **r:** Jährliche Preissteigerungsrate der Betriebs- und Wartungskosten
- **Anpassungskosten:** Gesamtkosten für Anpassungen und Erweiterungen während der Nutzungsdauer

Erklärung

Für eine näherungsweise Bestimmung der Gesamtkosten einer Build-Lösung werden insbesondere die Initialkosten und die jährlichen Betriebs- und Wartungskosten der jeweiligen Lösung berücksichtigt. In der Regel sind bei längeren Nutzungszeiten und komplexen Softwarelösungen noch gewisse Anpassungskosten nötig, die näherungsweise auf Basis der Initialkosten bestimmt werden können.



Bei der Entscheidung im Fall der Software können wir Ihnen helfen. Die Kosten sind ein sehr wichtiger Faktor im Entscheidungsfindungsprozess.

Die **folgenden Variablen** haben den größten Einfluss auf die Entscheidung zwischen einer Build- und einer Buy-Lösung:

Geplante Nutzungsdauer

1

Dies ist einer der wichtigsten Faktoren, der deutliche Hinweise darauf gibt, welche Lösung kostengünstiger ist. Bei kurzen geplanten Nutzungsdauern sind Buy-Varianten in der Regel günstiger. Mit steigender Nutzungsdauer gewinnen Build-Varianten zunehmend an Vorteil, da die initialen Entwicklungskosten durch den Wegfall laufender Lizenzkosten amortisiert werden.

Zahl der Nutzer:innen

2

Die Anzahl der Nutzer:innen ist in der Regel der größte Kostentreiber für die Lizenzgebühren einer Buy-Lösung, während sie für die Initialkosten einer Build-Lösung meist nur eine untergeordnete Rolle spielt. Mit steigender Nutzendenzahl erhöhen sich die Lizenzkosten einer Buy-Variante erheblich, wodurch eine Build-Lösung zunehmend attraktiver wird. Bei eher geringen Nutzendenzahlen bleibt die Buy-Lösung häufig die kosteneffizientere Wahl.

Anpassungs- und Integrationsaufwand

3

Eine Build-Lösung lässt sich gezielt an individuelle Anforderungen anpassen, während eine Buy-Lösung möglicherweise nicht alle Bedürfnisse erfüllt oder nur mit erheblichem Workaround- und Integrationsaufwand nutzbar ist. Je höher der erwartete Anpassungsbedarf in der Zukunft, desto vorteilhafter wird eine Eigenentwicklung.

KAPITEL 5:

PRAXISBEISPIEL: KOSTEN BUILD VS. BUY

Um den Nutzen und die Anwendung der beiden Formeln zur näherungsweisen Bestimmung der TCO von Build- und Buy-Softwarelösungen anschaulich darzustellen, betrachten wir ein konkretes Praxisbeispiel.



Praxisbeispiel:

In diesem konkreten Fall suchte ein mittelständisches Unternehmen nach einer neuen **Lösung für den internen Wissensaustausch** und die Organisation seines unternehmenseigenen Know-hows. Bisher wurde dafür Confluence genutzt und allen Abteilungen sowie Mitarbeitenden zur Verfügung gestellt. Das Unternehmen hat sich aus folgenden Gründen nach Alternativen umgesehen:

- **Erzwungener Wechsel zur Cloud:** Atlassian hat begonnen, alle Kunden zur Cloud-Migration zu drängen und On-Premise-Versionen entweder eingestellt oder nur mit erheblichen Zusatzkosten und Umwegen weiterhin ermöglicht.
- **Datenschutzrechtliche Bedenken:** Gerade in Anbetracht der steigenden Anforderungen an Datenschutz, Datensouveränität und unternehmensinterne Compliance-Richtlinien stellt die Cloud-Lösung für viele Unternehmen keine tragfähige Option dar.
- **Überdurchschnittliche Preiserhöhungen:** Seit Jahren steigen die Kosten für Confluence überproportional stark.
- **Geringe Nutzung & veraltetes Wissen:** Das Confluence-Wiki wurde von den Mitarbeitenden nur spärlich genutzt. Viele Inhalte waren veraltet, und eine interne Umfrage ergab, dass das Wiki im operativen Tagesgeschäft kaum eine Rolle spielte.
- **Bedarf an einer neuen Lösung:** Die Mitarbeitenden wünschten sich dennoch eine Möglichkeit, Best Practices, Prozesse und Wissen innerhalb der Abteilungen zu teilen – ohne auf SharePoint oder ähnliche Lösungen zurückgreifen zu müssen. Idealerweise sollte die neue Lösung schlank, flexibel und zusätzlich als Task-Management- sowie Reminder-Tool nutzbar sein.

Abgesehen von den Bedenken und anderen Gründen des Unternehmens, eine individuelle Lösung einzuführen, vergleichen wir nachfolgend die **Kosten der Build- und Buy-Varianten** über einen Nutzungszeitraum von fünf Jahren :

BUY-KOSTEN

Weitere Nutzung von Confluence

Preisliche Parameter:

- Die **Grundgebühren** für die Confluence-Lizenzen lagen zum Zeitpunkt der Anfrage des Unternehmens für 600 Nutzer bei ca. 35.000 €.
- Die **Preissteigerungen** bei Confluence waren erheblich in den letzten Jahren und sind mit 5–10 % pro Jahr im Mittel bei 7,5 % anzunehmen.

Lizenzgebühren pro Jahr:

- Jahr 1: 35.000 €
- Jahr 2: 38.000 €
- Jahr 3: 41.000 €
- Jahr 4: 44.000 €
- Jahr 5: 47.000 €

Gesamtkosten der Lizenzgebühren für Confluence über einen Nutzungszeitraum von fünf Jahren: 205.000 €.

** Zur Vereinfachung der Berechnung lassen wir in diesem Fall die Einführungs- und Implementierungskosten*

BUILD-KOSTEN

Eigenentwicklung

Preisliche Parameter:

- Die **initiale Entwicklung der Plattform** für Wissensaustausch und Aufgabenmanagement wurde in 85 Personentagen umgesetzt. Nach drei Jahren wurden noch zusätzliche Features in 40 Personentagen entwickelt (Tagessatz: 1.000 €/Tag).
- Die **Hostingkosten** lagen in den ersten drei Jahren bei 12.000 € pro Jahr und stiegen in den letzten beiden Jahren auf 14.000 € pro Jahr.

Gesamtkosten der Eigenentwicklung über einen Nutzungszeitraum von fünf Jahren: 189.000 €.

Im konkreten Beispiel zeigt sich, dass die Firma mit der Build-Variante nicht nur kostenseitig besser gefahren ist, sondern gleichzeitig auch zentrale Herausforderungen lösen konnte. Die neue Plattform wird von den Mitarbeitenden wieder häufiger genutzt und kann weiterhin on-premise betrieben werden – ein entscheidender Vorteil in Bezug auf Datensouveränität und Compliance.

Auch finanziell hat sich die Eigenentwicklung ausgezahlt, insbesondere weil viele Unternehmen oft teure Standardlösungen nutzen, obwohl sie nur einen Bruchteil der angebotenen Funktionen tatsächlich benötigen.

Ein weiterer Vorteil: Mit zunehmender Nutzungsdauer wird die Eigenentwicklung immer wirtschaftlicher, da die hohen Initialkosten bereits angefallen sind, während die laufenden Kosten vergleichsweise niedrig bleiben. Selbst bei einer steigenden Zahl an Nutzern bleibt die Eigenlösung langfristig die vorteilhaftere Option.

KAPITEL 6:

BILANZIELLE BETRACHTUNG DER SOFTWAREKOSTEN: AKTIVIERUNG VS. DIREKTER AUFWAND

Aufgrund der praktischen Relevanz wollen wir auch noch ein paar Sätze zur Bilanzierung von Softwarekosten verlieren. Denn neben der reinen Kostenbetrachtung im Rahmen der Total Cost of Ownership (TCO) spielt auch die buchhalterische Behandlung von Software eine entscheidende Rolle. Hierbei gibt es insbesondere zwei unterschiedliche Ansätze:



1. Aktivierung der Build-Lösung (Individualsoftware)

Wenn ein Unternehmen eine eigene Software entwickelt oder eine maßgeschneiderte Lösung beauftragt, können die anfallenden Entwicklungskosten unter bestimmten Voraussetzungen als immaterieller Vermögenswert in der Bilanz aktiviert werden. Dies bedeutet, dass die Kosten nicht sofort als einmaliger Aufwand in die Gewinn- und Verlustrechnung (GuV) einfließen, sondern über mehrere Jahre abgeschrieben werden. Die jährliche Abschreibung wird dann als Aufwand verbucht.

Vorteile der Aktivierung:

- Die Kosten werden über die wirtschaftliche Nutzungsdauer (in der Regel 3–5 Jahre) verteilt.
- Das EBITDA bleibt in den ersten Jahren höher, da nur die Abschreibungen und nicht die gesamte Investition auf einmal in die GuV einfließen.
- Die Investition in Software wird bilanziell als Wertbestandteil des Unternehmens ausgewiesen.

2. Direkter Aufwand bei SaaS-Lösungen

Im Gegensatz dazu wird bei Software-as-a-Service (SaaS) keine eigene Software erworben oder entwickelt, sondern lediglich ein Service genutzt. Die monatlichen oder jährlichen Abonnementkosten gelten als operativer Aufwand (OPEX) und werden sofort in der GuV erfasst – eine Aktivierung ist hier nicht möglich.

Unterschied zur aktivierten Software:

- Alle Kosten werden direkt in dem Jahr verbucht, in dem sie anfallen.
- Es gibt keine Abschreibung – der gesamte Betrag reduziert das Jahresergebnis sofort.
- SaaS belastet die Bilanz nicht mit zusätzlichen Vermögenswerten, bietet jedoch eine höhere Flexibilität und geringere Anfangsinvestitionen.

FAZIT

Unternehmen stehen bei der Auswahl neuer Software häufig vor der Entscheidung zwischen einer Eigenentwicklung (Build) und dem Kauf einer Standardsoftware (Buy). Ein entscheidendes Kriterium ist dabei oft der Preis. Neben den offensichtlichen direkten Kosten einer Software sollten immer auch indirekte Kosten berücksichtigt werden, darunter Support, Wartung, Schulungen, Infrastruktur sowie potenzielle Integrations- und Anpassungsausgaben.

Außerdem sollte man sich frühzeitig überlegen, wie lange die Software voraussichtlich genutzt wird, da diese Entscheidung maßgeblichen Einfluss auf die Wahl zwischen Build und Buy hat. Gleichmaßen spielen die Zahl der Nutzenden und der Umfang des Integrations- und Anpassungsaufwands eine große Rolle. Wir empfehlen in jedem Fall, verschiedene Szenarien durchzuspielen und potenzielle Entwicklungen sowie mögliche Stolpersteine zu berücksichtigen, die zu unerwartetem Mehraufwand führen könnten. Auch wenn bei jeder Variante unvorhergesehene Kosten auftreten können, erhält man so eine realistischere Einschätzung der tatsächlichen Gesamtkosten.

Wie geht es jetzt weiter?

Wir hoffen, dass dieses Whitepaper Unklarheiten aus dem Weg räumen und Ihnen Ihre Entscheidung erleichtern konnte. Wenn Sie noch Fragen haben, rufen Sie uns gerne an oder vereinbaren Sie einen Termin.

Wir melden uns umgehend und stehen Ihnen in einem ersten Gespräch gerne kostenfrei beratend zur Seite.

Wir bei makandra haben Erfahrung aus über 200 Projekten und eine überragende Kundenzufriedenheit. Warum? Wir arbeiten seit 14 Jahren getreu unseren Werten: Technische Exzellenz, Integrität, Förderkultur und Eigeninitiative. Davon profitieren nicht nur unser Team von Web-, DevOps-, Hosting- und UX/UI-Design-Expert*innen in Augsburg, sondern natürlich auch unsere Kund*innen. Wir beraten ehrlich und liefern nur Resultate ab, auf die wir auch stolz sein können.



KONTAKTIEREN SIE UNS



+49 821 58866 167



fabian.rimpl@makandra.de



<https://makandra.de>



Melli-Beese-Straße 5
86159 Augsburg