

NC · ASSOCIATE · APPLY

No-Code AI Automation Expert

Teilnehmerhandbuch — erweiterte Ausgabe mit Beispielen, Übungen, Fallstudien und Vorlagen

Automatisierungen in bestehenden Plattformen konfigurieren — und so bauen, dass sie den Betrieb überleben.

Foundation ✓

Associate

Professional

Expert

Umfang	6 Module · ca. 18 Stunden · 6 Übungen · 2 Musterprüfungen
Prüfung	20 Fragen · 40 Minuten · Bestehensgrenze 70 %
Voraussetzung	AI Foundations Professional (AF) oder anerkannte Gleichwertigkeit
Anhänge	Fallstudien · Vorlagen · Betriebshandbuch · Glossar · Index

CertifAI · Version 2.0 · Ausgabe DE · Rechtsstand August 2026

Dieses Handbuch bereitet auf die Zertifizierungsprüfung No-Code AI Automation Expert vor. Die Teilnahme an diesem Kurs ist keine Voraussetzung für die Prüfung; gleichwertige Vorbereitungswege werden anerkannt.

Vor dem Start

Einordnung, Voraussetzungen, Arbeitsweise

Wofür dieser Track steht

Gegenstand ist die **Konfiguration von Automatisierungen in bestehenden Plattformen** — Make, Zapier, n8n, Power Automate und vergleichbare Werkzeuge. Du verbindest Systeme, ordnest Daten zu, behandelst Fehler und entscheidest, an welcher Stelle ein Mensch im Ablauf stehen muss.

Nicht Gegenstand ist der Bau agentischer Systeme in Code — das ist *AI Agent & Automation Architect* (AA). Die Grenze verläuft scharf: Wer eine Plattform konfiguriert, arbeitet hier. Wer eine Orchestrierung programmiert, arbeitet dort. Ebenfalls nicht Gegenstand ist die Gestaltung einzelner Prompts — dafür gibt es *AI Prompt Engineer Specialist* (PE), dessen Modul zur Schemakontrolle in unserem Modul 6 vorausgesetzt, aber nicht wiederholt wird.



Abb. 0.2 Die drei benachbarten Tracks und ihre Trennlinie. Wer eine Plattform konfiguriert, arbeitet links; wer eine Orchestrierung programmiert, rechts.

Plattformneutral, absichtlich

Dieses Handbuch nennt keine Menüpfade und keine Modulnamen einer bestimmten Plattform. Die Werkzeuge ändern ihre Oberfläche zweimal im Jahr; die Konzepte dahinter — Auslöser, Verzweigung, Iteration, Wiederholung, Idempotenz — sind seit Jahrzehnten dieselben. Wer sie versteht, findet sich in jeder Plattform in einer Stunde zurecht. Wer nur Klickpfade gelernt hat, fängt bei jedem Werkzeugwechsel von vorn an.

Für die Übungen brauchst du Zugang zu mindestens einer Automatisierungsplattform. Ein kostenloser Tarif genügt. Wo eine Übung Daten erzeugt, die nach außen wirken würden — Mails, Bestellungen, Zahlungen — arbeitest du mit Testadressen und Testkonten; Abschnitt 2.4 beschreibt, wie eine Testauslösung aufgebaut wird.

DER SATZ, UM DEN ES IN DIESEM TRACK GEHT

Eine Automatisierung ist ein Mitarbeiter, der nie fragt. Sie führt aus, was du eingerichtet hast, tausendmal, ohne zu stutzen, wenn etwas seltsam aussieht. Das ist ihr ganzer Wert und ihr ganzes Risiko. Der Beruf besteht darin, im Voraus zu wissen, wann sie hätte fragen sollen — und genau dort einen Menschen einzubauen.

Was in dieser Ausgabe neu ist

Gegenüber Version 1.0 ist der fachliche Kern unverändert; der Prüfungsumfang ist derselbe. Hinzugekommen sind Abbildungen zu jedem Kernkonzept, ausgearbeitete Rechenmodelle, drei vollständige Fallstudien, ein Vorlagenteil zum direkten Weiterverwenden, ein Betriebskapitel und eine zweite Musterprüfung.

TEIL	UMFANG	PRÜFUNGSRELEVANT
Module 1–6	Lehrtext, 34 Abbildungen, 6 Übungen, 12 Selbsttestfragen	ja
Musterprüfung A und B	je 20 Fragen mit Begründung	Vorbereitung
Anhang A · Lösungen	Übungen, Selbsttests, beide Musterprüfungen	Vorbereitung
Anhang B · Fallstudien	drei Projekte von der Idee bis zum Betrieb	nein
Anhang C · Vorlagen	sechs Arbeitsblätter zum Übernehmen	nein
Anhang D · Betrieb	Übergabe, Überwachung, Governance, Datenschutz	nein
Anhang E–G	Glossar, Prüfungsinformationen, Index	Orientierung

Hinweise zur Qualifikation

CertifAI ist ein unabhängiger Anbieter. Dieses Zertifikat ist ein Industriezertifikat und keine staatlich anerkannte Qualifikation. Es wird von CertifAI ausgestellt und ist nicht behördlich zugelassen.

Teile dieses Handbuchs wurden KI-gestützt erstellt und anschließend fachlich geprüft und freigegeben. Die Prüfungsfragen werden unabhängig vom Kursmaterial entwickelt. Die Bewertung erfolgt deterministisch gegen einen von Fachexperten freigegebenen Antwortschlüssel; KI kommt dabei nicht zum Einsatz.

Genannte Plattformnamen dienen nur der Einordnung. Es besteht keine Verbindung zu den Herstellern, und keine Plattform wird empfohlen. Alle Zahlenbeispiele in diesem Handbuch sind Modellrechnungen mit offengelegten Annahmen; sie dienen dem Verständnis der Rechenwege und nicht als Marktdaten.

Inhaltsverzeichnis

Module, Prüfungsteile und Anhänge

ORIENTIERUNG

Vor dem Start	2
Wegweiser durch dieses Handbuch	6
Das Feld in Zahlen	8
Plattform-Landkarte	10

MODULE

Modul 1 • Was Automatisierung wirklich ist	12
1.1 Auslöser, Aktionen, Szenario	13
1.2 Wann automatisieren — vier Kriterien	14
1.3 Der Multiplikator wirkt in beide Richtungen	16
1.4 Den Prozess beschreiben, bevor du klickst	17
1.5 Anti-Muster der ersten Wochen	18
Modul 2 • Mehrstufige Szenarien	21
2.1 Die vier Bausteine über den Aktionen	22
2.2 Idempotenz	23
2.3 Teilerfolg: Schritt 4 von 7 scheitert	24
2.4 Wartbarkeit und Testauslösung	25
2.5 Laufzeit, Kontingent und Grenzen	27
Modul 3 • Webhooks und API-Grundlagen	31
3.1 Das Nötigste über HTTP	32
3.2 Statuscodes und was sie dir sagen	33
3.3 Authentifizierung	34
3.4 Webhooks empfangen und absichern	35
3.5 Kontingente und Seitenweise-Abruf	36
3.6 Eine Dokumentation lesen — in acht Schritten	38
Modul 4 • Datenzuordnung	41
4.1 Der stille Fehler	42
4.2 Die fünf Fallen	42
4.3 Verschachtelte Strukturen	44
4.4 Die Zuordnungstabelle	45
4.5 Abgleich statt Anlegen	46

Modul 5 • Fehlerbehandlung und der Mensch im Ablauf	49
5.1 Drei Fehlerklassen	50
5.2 Wiederholung richtig einstellen	51
5.3 Fehlerablage und Alarmierung	52
5.4 Wann ein Ablauf einen Menschen braucht	53
5.5 Der stille Ausfall und der Wächter	55
Modul 6 • KI-Agenten und Wirtschaftlichkeit	58
6.1 Ein KI-Schritt ist ein unzuverlässiger Schritt	59
6.2 Automatisierung oder Agent	60
6.3 Wirtschaftlichkeit rechnen	61
6.4 Kipppunkt und Sensitivität	63
6.5 Wenn sich eine Automatisierung nicht lohnt	65
PRÜFUNGSVORBEREITUNG	
Musterprüfung A • 20 Fragen	68
Musterprüfung B • 20 Fragen	71
ANHÄNGE	
A • Lösungen	74
B • Fallstudien	82
C • Vorlagen und Arbeitsblätter	86
D • Betrieb, Übergabe, Governance	89
E • Glossar	93
F • Prüfungsinformationen	96
G • Stichwortverzeichnis	98

Wegweiser durch dieses Handbuch

Aufbau der Module, Bedeutung der Kästen, Arbeitsweise

Jedes Modul folgt demselben Aufbau. Wer den Aufbau kennt, kann gezielt einsteigen — und beim zweiten Durchgang nur noch das lesen, was zählt.

BAUSTEIN	WAS DARIN STEHT	BEIM WIEDERHOLEN
Lernziele	Was du nach dem Modul können sollst — in Handlungen formuliert, nicht in Themen	zuerst lesen
Lehrtext	Der fachliche Kern, in nummerierten Abschnitten	überspringen
Abbildungen	Jedes Kernkonzept einmal als Bild; Bildunterschriften enthalten die Merksätze	lesen
Beispiel	Ein durchgerechneter Fall aus der Praxis mit benannten Zahlen	lesen
Merksatz-Kasten	Die Stelle, an der erfahrene Bauende Geld verlieren	lesen
Übung	Aufgabe ohne Musterlösung im Text — Lösungen in Anhang A	bearbeiten
Kernaussagen	Fünf bis sieben Sätze, die das Modul zusammenfassen	lesen
Selbsttest	Zwei Fragen im Prüfungsformat	bearbeiten

Die Farbcodierung der Kästen

VIOLETT Lernziele und Übungen — alles, was du tust.	GRÜN Kernaussagen — alles, was du behältst.	SANDFARBEN Merksätze und Warnungen — alles, was Geld kostet.	GRAU Beispiele und Fallstu- dien — alles, was schon passiert ist.
---	---	--	---

Zeitplan für die Vorbereitung

Der Track ist auf rund achtzehn Stunden ausgelegt. Die Verteilung unten hat sich als belastbar erwiesen, weil sie den Übungen so viel Zeit gibt wie dem Lehrtext — und weil sie die Musterprüfungen trennt.

ABSCHNITT	LESEN	ÜBUNG AN DER PLATTFORM	SUMME
Modul 1 · Grundlagen	1,0 h	1,0 h	2,0 h
Modul 2 · Szenarien	1,2 h	1,3 h	2,5 h
Modul 3 · Webhooks und API	1,3 h	1,7 h	3,0 h
Modul 4 · Datenzuordnung	1,2 h	1,3 h	2,5 h
Modul 5 · Fehler und Mensch	1,3 h	1,2 h	2,5 h
Modul 6 · KI und Wirtschaftlichkeit	1,5 h	1,0 h	2,5 h
Musterprüfung A + Auswertung	—	1,5 h	1,5 h
Musterprüfung B + Auswertung	—	1,5 h	1,5 h
Gesamt	7,5 h	10,5 h	18,0 h

EMPFEHLUNG ZUR REIHENFOLGE

Musterprüfung A frühestens nach Modul 6 bearbeiten, Musterprüfung B frühestens zwei Tage später. Der Abstand ist kein Ritual: Die zweite Prüfung misst, was nach dem Vergessen übrig geblieben ist — und genau das misst auch die echte Prüfung.

DEIN STAND

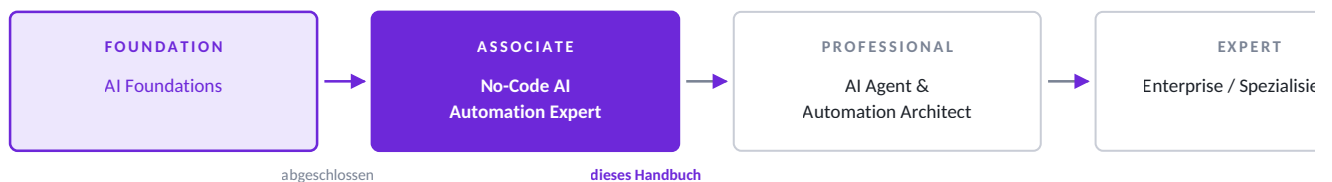


Abb. 0.1 Einordnung des Tracks im CertifAI-Stufenmodell. Der Associate-Track setzt die Foundation-Stufe voraus; gleichwertige Nachweise werden anerkannt.

Das Feld in Zahlen

Größenordnungen, Kostenprofile, typische Störungsbilder

Alle Zahlen auf den folgenden Seiten sind Modellrechnungen mit offengelegten Annahmen. Sie sind keine Marktstudie. Ihr Zweck ist, dir ein Gefühl für Größenordnungen zu geben — damit du im Gespräch mit einem Fachbereich sagen kannst, ob eine Schätzung plausibel ist.

Der Kostenunterschied der Auslösertypen

Eine Abfrage im Minutentakt erzeugt 43.200 Aufrufe im Monat, unabhängig davon, ob etwas passiert. Ein Webhook erzeugt so viele Aufrufe, wie es Ereignisse gibt. Bei einem Geschäftsprozess mit 420 Vorgängen im Monat ist das ein Verhältnis von rund 1 zu 100 — und beide Werte sind reale Rechnungsposten, weil fast alle Plattformen nach Vorgängen abrechnen.

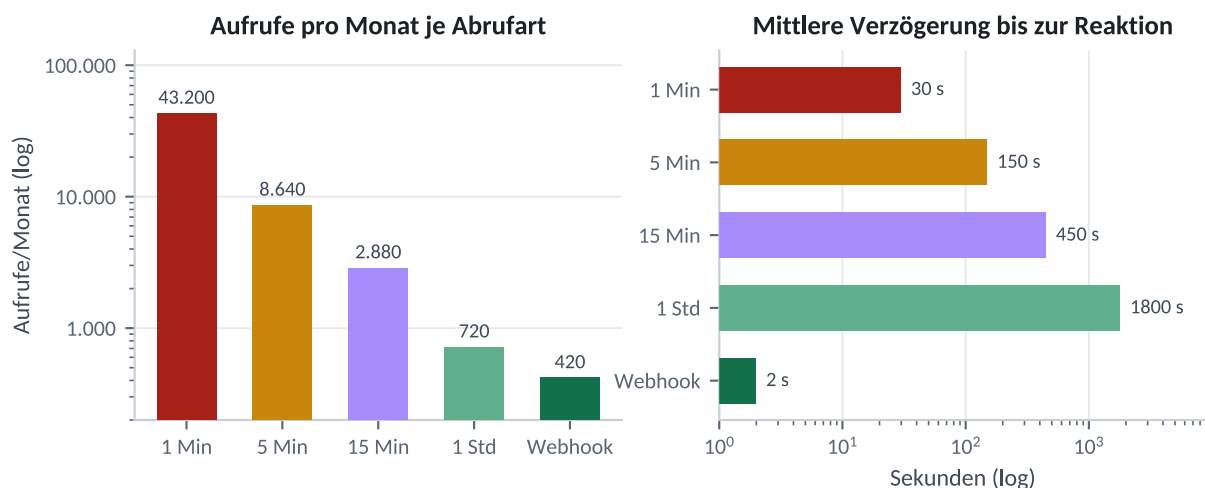


Abb. 0.3 Links die Aufrufzahlen, rechts die Kehrseite: Wer die Aufrufe senkt, erhöht die Verzögerung. Die Wahl ist kein Geschmack, sondern ein Tausch — und das Zeitverhalten des Geschäftsprozesses entscheidet, welcher Tausch richtig ist.

Wohin der Aufbauaufwand fließt

Der häufigste Planungsfehler ist nicht die Unterschätzung der Bauzeit, sondern ihre falsche Verteilung. Wer mehr als die Hälfte des Aufwands im Bauen verbringt, hat die Klärung übersprungen — und zahlt sie später im Betrieb.

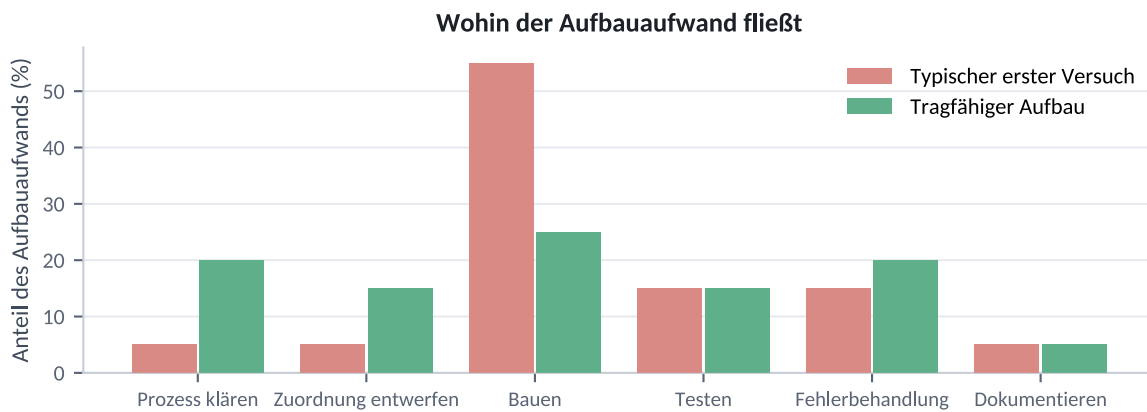


Abb. 0.4 Zwei Aufwandsprofile für dasselbe Vorhaben. Das rechte Profil braucht beim ersten Mal länger und im zweiten Jahr weniger als die Hälfte.

Störungsbilder im laufenden Betrieb

Die folgende Verteilung ist eine Modellannahme, die sich an wiederkehrenden Mustern orientiert: Die häufigsten Störungen sind selten die technisch interessantesten. Beachte die zweite Achse — die Behebungsdauer korreliert nicht mit der Häufigkeit. Genau deshalb lohnt sich Vorsorge dort, wo die Balken niedrig und die Punkte hoch sind.

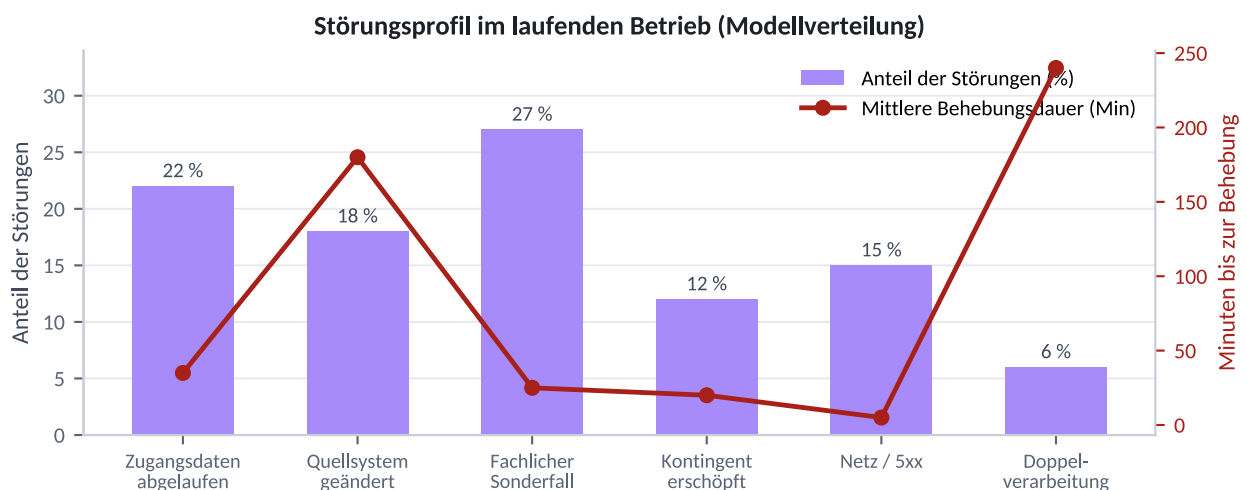


Abb. 0.5 Häufigkeit und Behebungsdauer laufen auseinander. Doppelverarbeitung ist selten — und der teuerste Einzelfall, weil die Korrektur in fremden Systemen stattfindet.

DREI ZAHLEN, DIE DU IM KOPF HABEN SOLLTEST

- **Rund zwei Drittel** der Lebenszykluskosten einer Automatisierung entstehen nach dem Aufbau — Wartung, Anpassung, Fehlerbearbeitung.
- **Fünf Ausführungen pro Woche** ist die Grenze, unterhalb derer sich der Aufbau in aller Regel nicht amortisiert.
- **Zwanzig Prozent Ausnahmen** ist die Grenze, oberhalb derer zuerst der Prozess bereinigt und dann automatisiert wird.

Plattform-Landkarte

Merkmale statt Empfehlungen — worauf bei der Auswahl zu achten ist

Dieses Handbuch empfiehlt keine Plattform. Die folgende Matrix beschreibt Merkmalsklassen, nach denen sich Werkzeuge unterscheiden — und die Fragen, die du vor einer Auswahl beantwortet haben musst.

MERKMAL	WARUM ES ZÄHLT	FRAGE AN DEN ANBIETER
Abrechnungseinheit	Ob je Vorgang, je Schritt oder je Ausführung abgerechnet wird, entscheidet, ob ein Iterator über 200 Positionen 1 oder 200 Einheiten kostet.	Was genau zählt als abrechenbare Einheit?
Fehlerbehandlung	Ohne Wiederaufnahme ab dem gescheiterten Schritt musst du jede Kompensation selbst bauen.	Gibt es Wiederaufnahme, und ab welchem Schritt?
Zustandsspeicher	Idempotenz braucht einen Ort, an dem verarbeitete Kennungen liegen.	Gibt es einen eingebauten Datenspeicher mit Eindeutigkeitsprüfung?
Webhook-Signatur	Ohne Signaturprüfung ist jede Webhook-Adresse eine offene Tür.	Wird die Signatur geprüft, oder muss ich das selbst bauen?
Protokollaufbewahrung	Fehlersuche braucht Ausführungsprotokolle, Datenschutz begrenzt sie.	Wie lange werden Protokolle gespeichert, und was steht darin?
Serverstandort	Bei personenbezogenen Daten entscheidet der Verarbeitungsort über die Zulässigkeit.	Wo wird verarbeitet, und gibt es einen AV-Vertrag?
Versionierung	Ohne Versionen ist jede Änderung an einem laufenden Szenario ein Risiko.	Kann ich eine frühere Fassung wiederherstellen?
Testauslösung	Ohne Testweg wird irgendwann in der Produktion getestet.	Wie starte ich einen Lauf mit Beispieldaten ohne Außenwirkung?
Rechteverwaltung	Technische Konten und kleinstmögliche Rechte sind eine Frage der Plattform.	Kann ich Zugänge je Szenario trennen und einzeln entziehen?
Ausstiegspfad	Exportierbare Definitionen entscheiden, wie teuer ein Wechsel wird.	Kann ich Szenarien maschinenlesbar exportieren?

DIE AUSWAHLFRAGE, DIE AM HÄUFIGSTEN FALSCH GESTELLT WIRD

„Welche Plattform ist die beste?“ hat keine Antwort. „Welche Plattform passt zu unserem Mengengerüst, unserer Datenlage und unserem Betriebsmodell?“ hat eine — und sie ergibt sich aus den zehn Zeilen oben. Rechne die Abrechnungseinheit immer am Iterator durch: Dort trennen sich die Tarife um den Faktor der Listenlänge.

Begriffe, die je Plattform anders heißen

KONZEPT IN DIESEM HANDBUCH

VERBREITETE ANDERE BEZEICHNUNGEN

Szenario	Zap, Workflow, Flow, Automation, Pipeline
Auslöser	Trigger, Starterereignis, Auslösemodul
Aktion	Step, Modul, Node, Aktionsschritt
Verzweigung	Router, Switch, Condition, Branch
Iterator	Loop, For-Each, Split-in-Batches, Iterate
Aggregator	Merge, Collect, Array-Aggregator, Join
Fehlerablage	Error Store, Dead-Letter-Queue, Fehlertabelle
Kontingentgrenze	Rate Limit, Quota, Throttle, Operations-Limit
Wiederaufnahme	Resume, Retry from step, Rerun incomplete

WARUM DAS WICHTIG IST

In der Prüfung werden die Begriffe dieses Handbuchs verwendet. In deiner Plattform heißen sie anders. Die Zuordnung oben ist der Übersetzungsschlüssel — lerne die Konzepte, nicht die Wörter.

MODUL 1

Was Automatisierung wirklich ist

Bevor eine einzige Verbindung geklickt wird, fallen drei Entscheidungen: welcher Auslöser, ob überhaupt, und was bei Ausnahmen geschieht. Dieses Modul behandelt genau diese drei — sie bestimmen mehr über den Erfolg als jede spätere technische Feinheit.

ABSCHNITT	INHALT	ZEIT
1.1	Auslöser, Aktionen, Szenario — und warum die Wahl des Auslösers eine Kostenentscheidung ist	20 Min
1.2	Vier Kriterien für die Entscheidung „automatisieren oder nicht“	25 Min
1.3	Der Multiplikator: warum Automatisierung Fehler vervielfacht	20 Min
1.4	Die Prozessbeschreibung in fünf Punkten	20 Min
1.5	Anti-Muster der ersten Wochen	15 Min
Übung 1	Sechs Prozesse bewerten und einordnen	45 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du kennst die drei Auslösertypen und ihre Folgen für Zuverlässigkeit und Kosten.
- Du entscheidest anhand von vier Kriterien, ob ein Prozess automatisiert werden sollte — und begründest die Entscheidung mit Zahlen.
- Du erkennst die Prozesse, bei denen Automatisierung den Schaden vervielfacht statt den Nutzen.
- Du kannst einen Prozess vor der Automatisierung sauber beschreiben.
- Du erkennst die fünf verbreiteten Anti-Muster der ersten Wochen und kennst die Gegenmaßnahme zu jedem.

1.1 Auslöser, Aktionen, Szenario

Jede Automatisierung besteht aus einem **Auslöser** und einer Folge von **Aktionen**. Das Ganze heißt je nach Plattform Szenario, Zap, Workflow oder Flow — gemeint ist dasselbe.

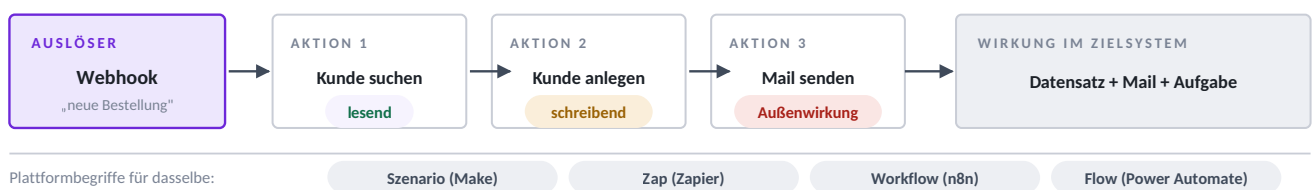


Abb. 1.1 Jede Automatisierung ist ein Auslöser plus eine Folge von Aktionen. Die Einfärbung nach lesend / schreibend / Außenwirkung entscheidet später über Reihenfolge und Wiederholbarkeit (Modul 2.3).

Die Einfärbung in der Abbildung ist keine Dekoration. Sie unterscheidet Aktionen danach, was ein zweiter Durchlauf anrichten würde: Eine lesende Aktion ist beliebig wiederholbar. Eine schreibende Aktion im eigenen System ist korrigierbar. Eine Aktion mit Außenwirkung ist es nicht — eine gesendete Mail kommt nicht zurück. Diese Unterscheidung trägt durch das ganze Handbuch: Sie bestimmt die Reihenfolge der Schritte (2.3), die Wiederholungsstrategie (5.2) und die Werkzeugvergabe an einen Agenten (6.2).

Die drei Auslösertypen

AUSLÖSERTYP	WIE ER FUNKTIONIERT	WORAUF ZU ACHTEN IST
Webhook sofort	Das Quellsystem ruft dich an, wenn etwas passiert	Schnellste Variante. Verpasste Aufrufe sind weg, wenn niemand wiederholt.
Abfrage Polling	Du fragst regelmäßig nach Neuem	Verzögerung bis zum Intervall. Verbraucht Aufrufe auch dann, wenn nichts passiert.
Zeitplan Scheduler	Läuft zu festen Zeiten	Vorhersagbar. Achtung bei Zeitumstellung und Feiertagen.

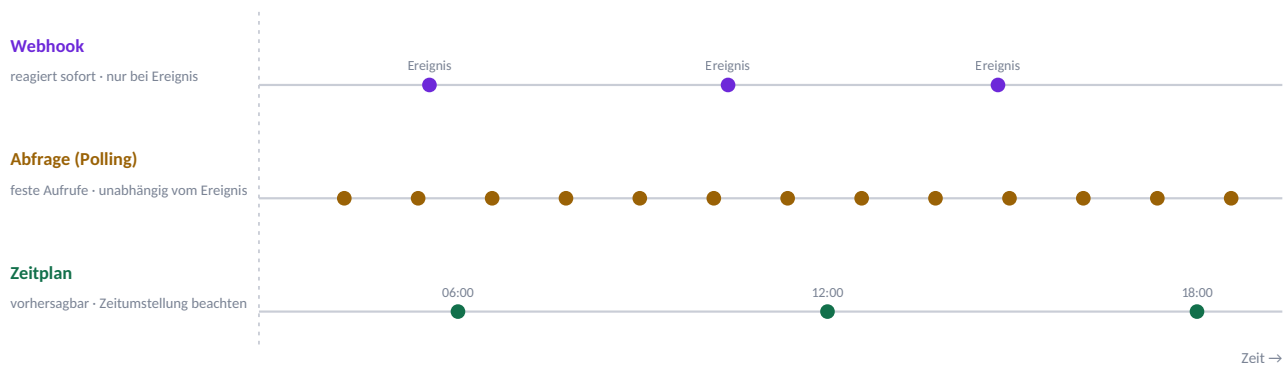


Abb. 1.2 Dieselbe Zeitspanne, drei Auslösertypen. Die Abfrage erzeugt Aufrufe auch dann, wenn nichts passiert — das ist der Kostenunterschied aus Abb. 1.3.

Der Unterschied ist keine Geschmacksfrage. Ein Webhook, den du beim Empfang nicht schnell genug bestätigst, wird vom Absender wiederholt — und dein Szenario läuft zweimal. Eine Abfrage, die alle fünf Minuten läuft und in 99 Prozent der Fälle nichts findet, verbraucht trotzdem 8.640 Aufrufe pro Monat. Beides sind reale Rechnungsposten und reale Fehlerquellen.

Wann welcher Auslöser

SITUATION	AUSLÖSER	BEGRÜNDUNG
Quellsystem bietet Webhooks an und die Reaktion soll unmittelbar erfolgen	Webhook	Niedrigste Kosten bei höchster Geschwindigkeit
Quellsystem bietet keine Webhooks, Verzögerung von Minuten ist unkritisch	Abfrage	Intervall so lang wählen wie fachlich vertretbar
Verarbeitung ist ohnehin stapelweise — Tagesabschluss, Preisliste, Auswertung	Zeitplan	Ein Lauf statt vieler, planbare Last
Webhook vorhanden, aber Ereignisse dürfen unter keinen Umständen verloren gehen	Webhook + Zeitplan	Der Zeitplan holt nach, was der Webhook verpasst hat — die Idempotenz aus Modul 2 verhindert Doppelverarbeitung

DIE KOMBINATION, DIE NIEMAND BAUT UND JEDER BRAUCHT

Die vierte Zeile ist der Fall, den Anleitungen auslassen. Ein Webhook ist schnell, aber nicht zuverlässig: Fällt deine Plattform für zwanzig Minuten aus, sind die Aufrufe dieser zwanzig Minuten in vielen Quellsystemen unwiederbringlich verloren. Ein zusätzlicher Abgleichslauf einmal am Tag — „gibt es Vorgänge der letzten 48 Stunden, die ich nicht verarbeitet habe?“ — kostet eine halbe Stunde Aufbau und schließt genau diese Lücke. Voraussetzung ist, dass der Ablauf idempotent ist; sonst verdoppelt der Abgleich, was er reparieren soll.

1.2 Wann automatisieren — vier Kriterien

Die Frage lautet nie „lässt sich das automatisieren?“ — fast alles lässt sich. Sie lautet: „Lohnt es sich, und was passiert, wenn es schiefgeht?“ Vier Kriterien beantworten das, und sie stehen in einer Reihenfolge, die Arbeit spart.

Häufigkeit

Unter etwa fünf Ausführungen pro Woche lohnt sich der Aufbau selten. Die Faustregel: Zeiterparnis pro Jahr muss die Aufbau- und Wartungszeit deutlich übersteigen, nicht nur die Aufbauzeit. Wer nur den Aufbau rechnet, kommt bei jedem zweiten Vorhaben zu einem Ergebnis, das im zweiten Jahr negativ wird.

Stabilität der Regel

Ein Prozess, dessen Regeln sich monatlich ändern, wird zu einer Automatisierung, die monatlich umgebaut wird. Das ist teurer als Handarbeit. Die Prüffrage: Steht die Regel seit mindestens sechs Monaten unverändert — und ist absehbar, dass sie die nächsten zwölf übersteht?

Anteil der Ausnahmen

Wenn jeder dritte Vorgang eine Sonderbehandlung braucht, automatisierst du den Normalfall und erzeugst einen Sonderfallstau. Ab etwa 20 Prozent Ausnahmen lohnt sich zuerst die Bereinigung des Prozesses, dann die Automatisierung.

Fehlerkosten mal Häufigkeit

Das ist das entscheidende Kriterium und das am häufigsten übersehene. Ein Fehler, der einmal pro Woche in Handarbeit passiert und auffällt, passiert automatisiert zweihundertmal und fällt nicht auf.

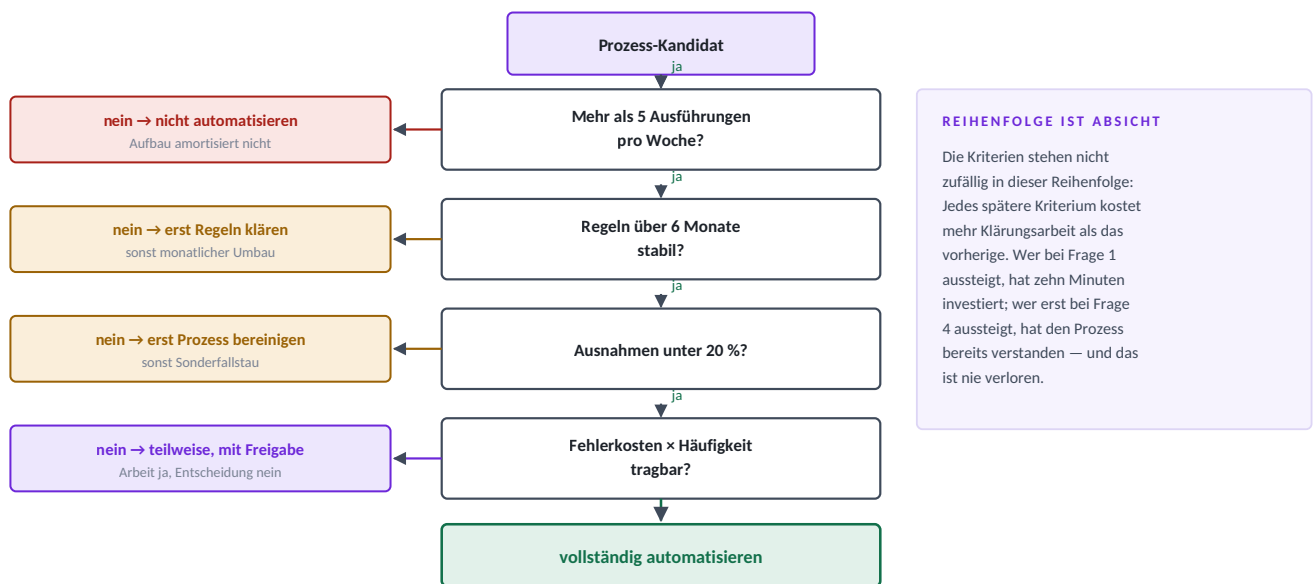


Abb. 1.4 Vier Kriterien als Reihenfolge gelesen. Jedes „nein“ hat eine benannte Folgehandlung — „nicht automatisieren“ ist nur bei Kriterium 1 das Ergebnis.

DIE KRITERIEN ALS PUNKTWERT

Für Vorhaben, die gegeneinander abgewogen werden müssen, hat sich eine einfache Bewertung bewährt. Jedes Kriterium wird mit 0 bis 3 bewertet; das Produkt — nicht die Summe — ergibt die Rangfolge. Das Produkt ist richtig, weil eine Null in einem Kriterium das ganze Vorhaben kippt, während eine Summe sie wegmittelt.

KRITERIUM	0	1	2	3
Häufigkeit / Woche	< 5	5-20	20-100	> 100
Regelstabilität	monatlich	quartalsweise	jährlich	stabil
Ausnahmeanteil	> 30 %	20-30 %	5-20 %	< 5 %
Fehlerfolgen	irreversibel, unentdeckt	teuer, spät entdeckt	korrigierbar	sofort sichtbar

1.3 Der Multiplikator wirkt in beide Richtungen

Automatisierung vervielfacht nicht den Nutzen, sondern den *Prozess* — mitsamt seinen Fehlern. Ein Prozess, der zu 95 Prozent funktioniert, ist in Handarbeit erträglich: Die fünf Prozent fallen der Sachbearbeiterin auf, weil etwas seltsam aussieht.

Automatisiert werden aus fünf Prozent bei 2.000 Vorgängen hundert falsche Ergebnisse, die niemand ansieht. Deshalb gilt die Reihenfolge: **erst den Prozess klären, dann automatisieren**. Wer eine unklare Vorgehensweise automatisiert, bekommt eine schnelle unklare Vorgehensweise.

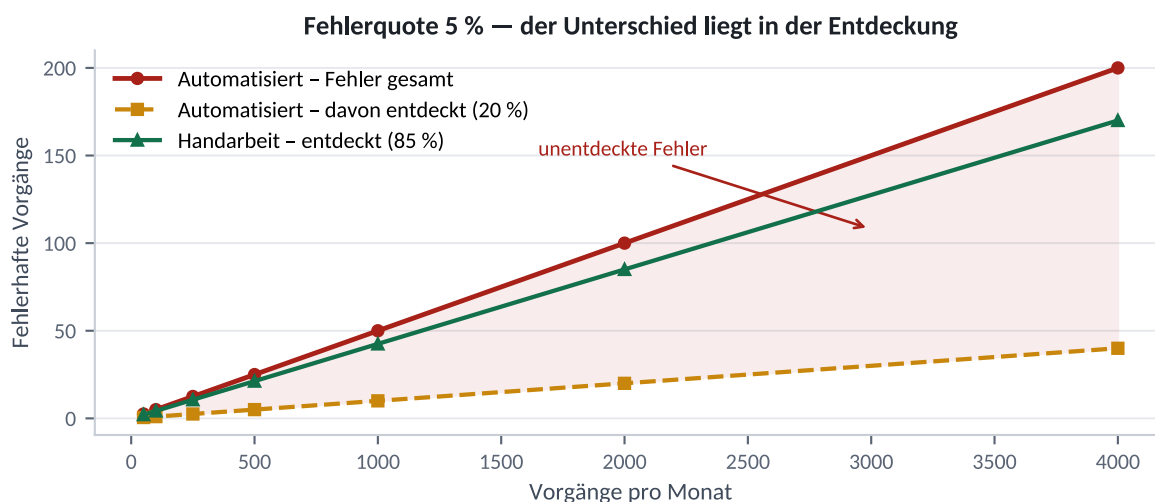


Abb. 1.3 Dieselbe Fehlerquote, zwei Welten. Der entscheidende Unterschied ist nicht die Zahl der Fehler, sondern der Anteil, der bemerkt wird: In Handarbeit stolpert ein Mensch über das Ungewöhnliche, automatisiert läuft es durch.

DER MULTIPLIKATOR IN ZAHLEN

Rechne den Effekt einmal für dein eigenes Vorhaben durch. Die Formel hat drei Faktoren:

Unentdeckte Fehler pro Jahr
= Vorgänge/Jahr × Fehlerquote × (1 – Entdeckungsquote)

Beispiel Handarbeit: $1.800 \times 5 \% \times (1 - 0,85) = 13,5$
Beispiel automatisiert: $1.800 \times 5 \% \times (1 - 0,20) = 72,0$

Faktor: 5,3 – bei identischer Fehlerquote.

Die Entdeckungsquote ist der einzige Faktor, den du beim Bauen wirklich in der Hand hast. Sie ist der Grund, warum Modul 5 (Fehlerablage, Plausibilitätsprüfung, Wächter) wirtschaftlich der wertvollste Teil dieses Tracks ist.

1.4 Den Prozess beschreiben, bevor du klickst

Vor jedem Szenario steht eine schriftliche Beschreibung. Fünf Punkte, eine halbe Seite:

1. **Auslöser:** Was genau startet den Ablauf? Wie oft tritt das auf?
2. **Eingaben:** Welche Daten liegen vor, aus welchem System, in welcher Form?
3. **Regeln:** Welche Entscheidungen werden getroffen, nach welchen Kriterien?
4. **Wirkung:** Was verändert sich am Ende — und in welchen Systemen?
5. **Ausnahmen:** Welche Fälle passen nicht ins Muster, und was soll dann geschehen?

EINE HALBE SEITE, VOR DEM ERSTEN KLICK

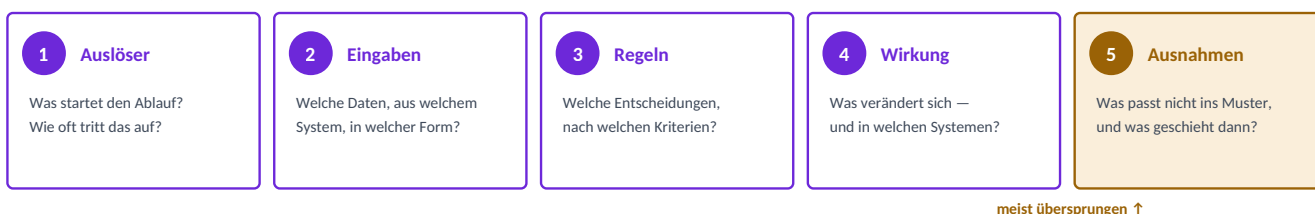


Abb. 1.5 Die fünf Punkte der Prozessbeschreibung. Punkt 5 ist der, den man beim direkten Losbauen überspringt — und drei Wochen später mitten in einem gewachsenen Szenario nachholt.

Punkt 5 ist der Teil, den man beim direkten Losbauen überspringt und drei Wochen später nachholt — dann aber mitten in einem gewachsenen Szenario, in dem jede Ergänzung eine Verzweigung mehr bedeutet.

BEISPIEL 1.1 — ZWEI PROZESSE, ZWEI ENTSCHEIDUNGEN

A — Eingehende Bewerbungen ins Bewerbermanagement übertragen. Häufigkeit: 15 pro Woche. Regeln stabil. Ausnahmen: etwa jede zehnte Bewerbung kommt ohne Anhang oder als Fließtext. Fehlerkosten: mittel — eine verlorene Bewerbung ist ärgerlich, aber nachvollziehbar, weil die E-Mail bleibt.

Entscheidung: automatisieren, mit einem Zweig für Bewerbungen ohne Anhang, der in eine manuelle Ablage läuft statt zu scheitern.

B — Rechnungen freigeben und zur Zahlung anweisen. Häufigkeit: 40 pro Woche. Regeln stabil. Ausnahmen: gering. Fehlerkosten: *hoch und schwer umkehrbar* — eine fälschlich angewiesene Zahlung ist Geld, das weg ist.

Entscheidung: teilweise automatisieren. Prüfung, Zuordnung und Vorbereitung automatisch; die Freigabe bleibt ein Klick eines Menschen. Der Zeitgewinn liegt ohnehin in der Vorbereitung, nicht im Klick — die Freigabe ist der billigste Teil des Prozesses und der teuerste Fehler.

Das Muster gilt weit über dieses Beispiel hinaus: Automatisiere die Arbeit, nicht die Entscheidung.

1.5 Anti-Muster der ersten Wochen

Die folgenden fünf Muster tauchen in fast jedem ersten Projekt auf. Sie sind keine Anfängerfehler im abwertenden Sinn — sie entstehen, weil die Plattformen genau dazu einladen.

ANTI-MUSTER	WORAN DU ES ERKENNST	GEGENMASSNAHME
Das Alles-Szenario	Ein Szenario mit 30 Schritten, das drei Dinge gleichzeitig tut. In der Beschreibung kommt „und außerdem“ vor.	Ein Szenario, ein Zweck. Teilen — auch wenn beide danach denselben Auslöser haben.
Der blinde Optimist	Kein einziger Fehlerpfad. Das Szenario geht davon aus, dass jede Antwort kommt und jedes Feld gefüllt ist.	Mindestens Fehlerablage und Alarmierung (5.3), bevor das Szenario scharf geschaltet wird.
Der Zeitbomben-Iterator	Ein Iterator ohne Blick auf die Listenlänge. Läuft im Test über 3 Positionen, in der Produktion über 400.	Listenlänge im Test künstlich hochsetzen; Kontingent gegen die maximale Länge rechnen (2.5).
Der persönliche Zugang	Die Verbindung läuft über das Konto der Kollegin, die das Szenario gebaut hat.	Technisches Konto je Automatisierung, kleinstmögliche Rechte (3.3).
Der Produktions-test	Es gibt keinen Weg, das Szenario ohne echte Wirkung zu starten. Getestet wird „einmal kurz scharf“.	Testauslösung mit Beispieldaten und abgeschalteter Außenwirkung (2.4).

Vier der fünf Muster haben dieselbe Wurzel: Das Szenario wurde für den Erfolgsfall gebaut und nicht für den Betrieb. Die Module 2 bis 5 behandeln genau die Gegenmaßnahmen — in der Reihenfolge, in der sie beim Bauen gebraucht werden.

ÜBUNG 1 — AUTOMATISIEREN ODER NICHT

Bewerte jeden Prozess nach den vier Kriterien aus 1.2 und entscheide: vollständig automatisieren, teilweise automatisieren (mit benanntem Freigabeschritt) oder nicht automatisieren.

1. Monatliche Umsatzauswertung aus drei Systemen zu einer Tabelle zusammenführen.
2. Kündigungen von Kundenverträgen im System als beendet markieren.
3. Neue Leads aus dem Webformular ins CRM übertragen und eine Bestätigungsmail senden.
4. Bei Lagerbestand unter Mindestmenge automatisch beim Lieferanten nachbestellen.
5. Krankmeldungen aus dem E-Mail-Postfach in die Zeiterfassung übernehmen.
6. Zweimal im Jahr die Preisliste auf der Website aktualisieren.

Teil B: Für die beiden Prozesse, die du als „teilweise“ eingestuft hast: An welcher exakten Stelle sitzt der Mensch, und welche Information braucht er, um in zehn Sekunden entscheiden zu können?

Teil C: Wähle den Prozess mit dem höchsten Punktwert und schreibe die Prozessbeschreibung nach 1.4 — alle fünf Punkte, eine halbe Seite. Punkt 5 vollständig: mindestens drei benannte Ausnahmen mit jeweils einer Festlegung, was geschehen soll.

KERNAUSSAGEN

- Drei Auslösertypen mit verschiedenen Kosten- und Fehlerprofilen; die Kombination aus Webhook und nachholendem Zeitplan schließt die Zuverlässigkeitslücke.
- Vier Kriterien: Häufigkeit, Stabilität der Regel, Ausnahmeanteil, Fehlerkosten mal Häufigkeit — als Produkt bewertet, nicht als Summe.
- Automatisierung vervielfacht den Prozess samt seinen Fehlern. Erst klären, dann automatisieren.
- Die Entdeckungsquote ist der Faktor, den du beim Bauen wirklich beeinflusst.
- Automatisiere die Arbeit, nicht die Entscheidung.
- Fünf Punkte schriftlich, bevor du das erste Modul anlegst.

Selbsttest Modul 1

1. Welches Kriterium wird bei der Entscheidung für eine Automatisierung am häufigsten übersehen?

- a) Die Häufigkeit der Ausführung.
- b) Fehlerkosten multipliziert mit der Häufigkeit.
- c) Die Kosten der Plattform.
- d) Die Zahl der beteiligten Systeme.

2. Ein Prozess funktioniert in Handarbeit zu 95 Prozent. Was folgt für die Automatisierung?

- a) 95 Prozent sind ein guter Ausgangswert.
- b) Die fehlenden 5 Prozent werden vervielfacht und fallen niemandem mehr auf.
- c) Die Automatisierung verbessert die Quote automatisch.
- d) Der Prozess ist für Automatisierung ungeeignet.

MODUL 2

Mehrstufige Szenarien

Ein Szenario mit drei Schritten funktioniert fast immer. Ab sieben Schritten entscheidet nicht mehr die Logik über den Erfolg, sondern der Umgang mit Wiederholung, Teilerfolg und Wachstum. Dieses Modul enthält den wichtigsten einzelnen Begriff des ganzen Tracks: Idempotenz.

ABSCHNITT	INHALT	ZEIT
2.1	Filter, Verzweigung, Iterator, Aggregator — und ihre Kostenwirkung	25 Min
2.2	Idempotenz und der Deduplikationsschlüssel	35 Min
2.3	Teilerfolg: Wiederaufnahme oder Kompensation	20 Min
2.4	Wartbarkeit, Benennung, Testauslösung	20 Min
2.5	Laufzeit, Kontingent und die Grenzen der Plattform	20 Min
Übung 2	Idempotenz in einem gegebenen Szenario herstellen	50 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du setzt Filter, Verzweigungen, Iteratoren und Aggregatoren gezielt ein und kennst ihre Kostenwirkung.
- Du erkennst und behebst fehlende Idempotenz — die häufigste Ursache doppelter Wirkungen.
- Du weißt, in welchem Zustand ein Szenario ist, wenn Schritt 4 von 7 scheitert, und entscheidest begründet zwischen Wiederaufnahme und Kompensation.
- Du hältst Szenarien wartbar, statt sie wachsen zu lassen.
- Du rechnest Laufzeit und Kontingentverbrauch vor dem Bauen aus.

2.1 Die vier Bausteine über den Aktionen

Über den eigentlichen Aktionen liegen vier Bausteine, die in jeder Plattform existieren — unter unterschiedlichen Namen. Wer sie kennt, liest jedes fremde Szenario in Minuten.



Abb. 2.1 Die vier Bausteine über den Aktionen. Drei davon haben eine unmittelbare Kostenwirkung: Filter senkt, Iterator vervielfacht, Aggregator bündelt.

Filter

Filter lassen einen Ablauf nur unter Bedingungen weiterlaufen. Setze sie so früh wie möglich — jeder Schritt hinter einem Filter, der ohnehin abbricht, kostet Geld und Zeit. Die Rechnung ist einfach: Ein Filter, der 70 Prozent der Vorgänge aussortiert und an Position 1 statt Position 5 steht, spart 70 Prozent von vier Schritten.

Verzweigung

Verzweigung teilt den Ablauf in mehrere Wege. Wichtig ist die Frage, ob sich die Wege gegenseitig ausschließen: Bei überlappenden Bedingungen laufen zwei Wege gleichzeitig, und beide schreiben ins Zielsystem. Prüfe jede Verzweigung mit der Frage: Gibt es einen Datensatz, der auf zwei Zweige passt? Und: Gibt es einen, der auf keinen passt — und was geschieht dann mit ihm?

Iterator

Der Iterator zerlegt eine Liste in Einzelschritte. Der teure Teil: Bei 200 Positionen läuft alles Nachfolgende 200-mal. Ein unbedachter Iterator ist die häufigste Ursache einer plötzlich erschöpften Kontingentgrenze — und, weil die Testdaten meist drei Positionen haben, fällt er im Test nie auf.

Aggregator

Der Aggregator fügt Einzelschritte wieder zusammen. Wer iteriert, muss meist auch aggregieren — sonst entstehen 200 einzelne E-Mails statt einer Übersicht. Die Frage „eine Zeile je Position oder eine Zeile je Auftrag?“ ist keine technische, sondern eine fachliche; sie gehört in die Prozessbeschreibung aus 1.4.

BAUSTEIN	WIRKUNG AUF DIE VORGANGSZAHL	TYPISCHER FEHLER
Filter	senkt — je früher, desto stärker	steht zu spät im Ablauf
Verzweigung	neutral bis verdoppelnd	überlappende Bedingungen, kein Sonst-Zweig
Iterator	multipliziert mit der Listenlänge	Listenlänge im Test zu klein
Aggregator	senkt wieder auf 1	fehlt — Nachrichtenflut statt Übersicht

2.2 Idempotenz

Ein Ablauf ist **idempotent**, wenn er zweimal ausgeführt dasselbe Ergebnis hat wie einmal ausgeführt. Das klingt akademisch und ist der wichtigste praktische Begriff dieses Moduls.

Denn Wiederholungen sind kein Ausnahmefall, sondern Normalbetrieb: Ein Webhook wird wiederholt gesendet, weil die Bestätigung zu spät kam. Ein Wiederholungsversuch nach einem Netzwerkfehler startet den Ablauf neu. Jemand löst ein Szenario zum Test von Hand aus. Eine Abfrage liefert denselben Datensatz zweimal, weil das Zeitfenster überlappt. Ein Kollege importiert eine CSV-Datei ein zweites Mal, weil er nicht sicher war, ob der erste Import durchlief.

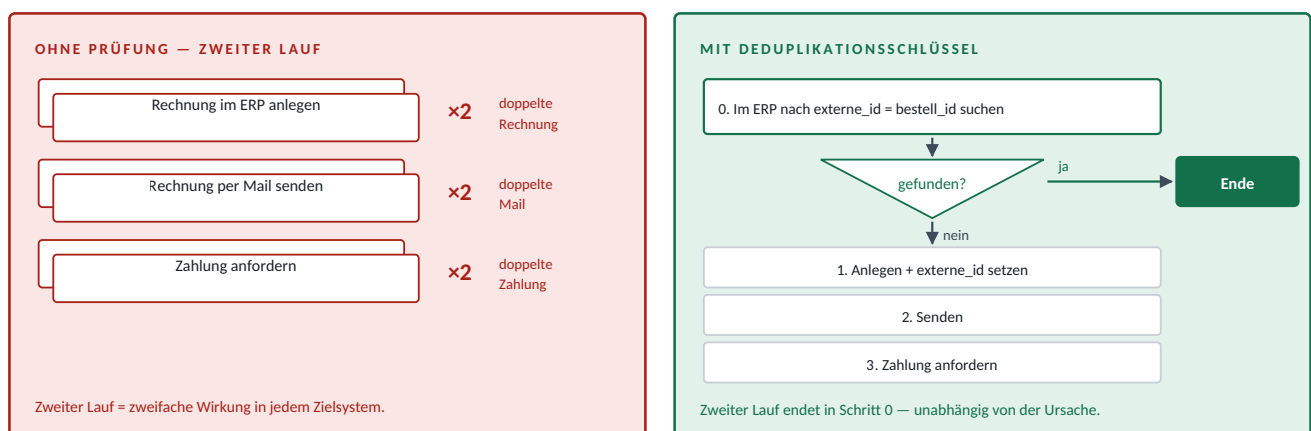


Abb. 2.2 Idempotenz ist kein Sonderfall-Schutz, sondern die einzige Absicherung, die unabhängig von der Ursache der Wiederholung wirkt.

Der Kern ist der **Deduplikationsschlüssel**: eine Kennung aus dem Quellsystem, die im Zielsystem gespeichert wird und dort eindeutig sein muss. Ohne sie hilft keine Prüfung, weil du nicht wiedererkennen kannst, was du schon getan hast.

Wo der Schlüssel herkommt

Gute Schlüssel sind stabil und stammen aus dem Quellsystem: Bestellnummer, Nachrichten-ID, Datensatz-ID, Rechnungsnummer. Schlechte Schlüssel sind Zeitstempel, Zeilennummern oder aus mehreren Feldern zusammengesetzte Zeichenketten — sie ändern sich, sobald jemand einen Tippfehler korrigiert.

KANDIDAT	TAUGLICH?	BEGRÜNDUNG
Bestellnummer aus dem Shop	ja	Stabil, im Quellsystem eindeutig, ändert sich nie
Nachrichten-ID einer E-Mail	ja	Wird vom Mailserver vergeben und bleibt bestehen
Zeilen-ID eines Tabellenblatts	meist ja	Stabil, solange Zeilen nicht neu nummeriert werden
E-Mail-Adresse des Kunden	bedingt	Eindeutig je Person, aber änderbar — als Abgleichsschlüssel geeignet, als Vorgangsschlüssel nicht
Zeitstempel des Laufs	nein	Bei jedem Lauf anders — erkennt niemals eine Wiederholung
Zeilennummer der Datei	nein	Verschiebt sich, sobald eine Zeile eingefügt wird
Name + Datum + Betrag	nein	Ändert sich bei jeder Korrektur; erzeugt bei gleichartigen Vorgängen falsche Treffer

Wenn das Zielsystem kein Feld für eine Fremdkennung hat, ist die zweitbeste Lösung eine eigene Tabelle (Datenbank, Tabellenblatt, Plattform-Datenspeicher) mit den Spalten `quelle_id`, `verarbeitet_am`, `ergebnis`. Das ist zwei Schritte Aufwand und verhindert die peinlichste Fehlerklasse überhaupt.

PRÜFEN UND SCHREIBEN SIND NICHT ATOMAR

Zwischen „ich habe nachgesehen, es gibt den Datensatz noch nicht“ und „ich lege ihn an“ vergehen Millisekunden. Laufen zwei Durchgänge gleichzeitig — etwa weil ein Webhook doppelt ankommt — können beide die Prüfung bestehen und beide anlegen.

Drei Gegenmittel, in dieser Reihenfolge der Wirksamkeit: **Erstens** eine Eindeutigkeitsregel im Zielsystem selbst; dann scheitert der zweite Schreibvorgang mit `409`, und das ist der beste Fall. **Zweitens** ein Schreibvorgang, der anlegt oder aktualisiert (oft `PUT` auf eine bekannte Kennung) statt anzulegen. **Drittens**, wenn beides fehlt: eine Sperre über die Verarbeitungstabelle — Zeile mit Status „in Arbeit“ schreiben, bevor die eigentliche Verarbeitung beginnt.

2.3 Teilerfolg: Schritt 4 von 7 scheitert

Automatisierungsplattformen sind keine Datenbanken. Es gibt kein „alles oder nichts“: Schritt 1 bis 3 sind ausgeführt, Schritt 4 scheitert, Schritt 5 bis 7 laufen nie. Dein System steht jetzt in einem Zwischenzustand, den niemand entworfen hat.

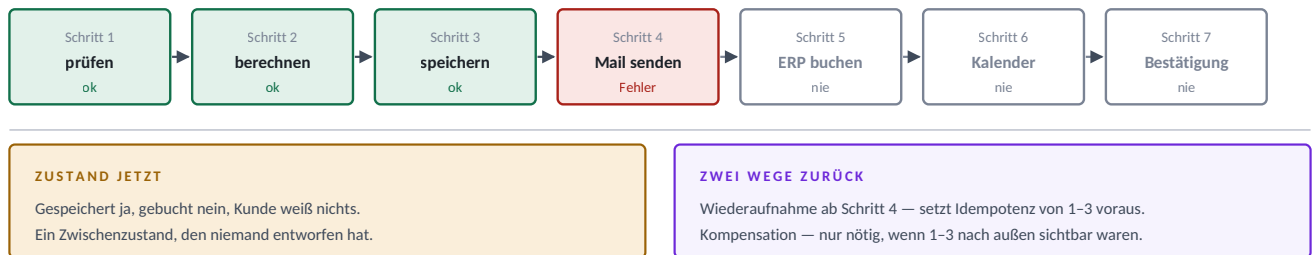


Abb. 2.3 Automatisierungsplattformen kennen kein „alles oder nichts“. Der Teilerfolg ist der Normalfall, für den ein Entwurf vorliegen muss.

Zwei Umgangsweisen, und die Wahl richtet sich nach der Umkehrbarkeit:

ANSATZ	VORGEHEN	WANN
Wiederaufnahme	Der Ablauf wird ab dem gescheiterten Schritt fortgesetzt — setzt Idempotenz der vorherigen Schritte voraus	Standardfall, wenn die bisherigen Wirkungen bestehen bleiben dürfen
Kompensation	Die bereits erfolgten Schritte werden rückgängig gemacht	Wenn ein Zwischenzustand nach außen sichtbar oder schädlich ist

Die praktische Konsequenz für den Entwurf: **Ordne die Schritte so, dass die nach außen wirkenden am Ende stehen.** Erst prüfen, berechnen, im eigenen System speichern — dann die Mail senden, dann die Zahlung anweisen. Ein Ablauf, der in Schritt 2 eine Kundenmail sendet und in Schritt 6 feststellt, dass die Daten unvollständig waren, ist falsch gebaut.

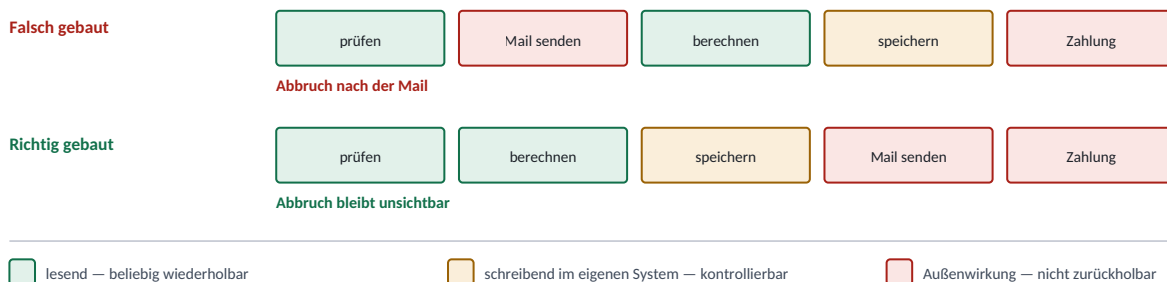


Abb. 2.4 Das Ordnungsprinzip in einem Bild: erst lesen, dann rechnen, dann im eigenen System speichern — und alles mit Außenwirkung ans Ende.

KOMPENSATION IST TEURER, ALS SIE AUSSIEHT

Eine Kompensation ist selbst ein Ablauf, der scheitern kann — und sie läuft ausgerechnet dann, wenn schon etwas schiefgegangen ist. Deshalb gilt: Kompensation ist die Notlösung für Fälle, in denen die Umordnung der Schritte nicht möglich ist. In der Praxis lassen sich vier von fünf Kompensationsbedarfen durch eine andere Reihenfolge vermeiden.

2.4 Wartbarkeit und Testauslösung

Szenarien wachsen. Nach einem Jahr hat das ursprünglich siebenschrittige Szenario 34 Module, vier Verzweigungen und drei Sonderfälle, deren Anlass niemand mehr kennt. Vier Gewohnheiten halten dagegen:

- **Benenne jeden Schritt fachlich** — „Kundennummer aus CRM holen“, nicht „HTTP-Anfrage 3“.
- **Ein Szenario, ein Zweck.** Wenn du „und außerdem“ sagst, um es zu beschreiben, sind es zwei.
- **Notiere das Warum** zu jedem Sonderfall in der Beschreibung, mit Datum und Anlass.
- **Baue eine Testauslösung** — einen Weg, das Szenario mit Beispieldaten ohne Außenwirkung zu starten. Ohne sie testet irgendwann jemand in der Produktion.

Wie eine Testauslösung aufgebaut wird

Eine Testauslösung ist kein zweites Szenario, sondern ein Schalter im bestehenden. Drei Bestandteile:

1. Eingang
 - Auslöser akzeptiert ein Feld `test = true`
 - Fehlt das Feld, gilt `test = false` (nie umgekehrt)
2. Umleitung der Außenwirkung
 - Mailempfänger -> Testadresse
 - Zielsystem -> Testmandant oder Testtabelle
 - Zahlung -> Schritt wird übersprungen, Eintrag ins Protokoll
3. Kennzeichnung
 - Jeder erzeugte Datensatz bekommt `quelle = "TEST"`
 - Deduplikationsschlüssel bekommt ein Präfix `TEST-`
(sonst blockiert der Testlauf den späteren echten Lauf)

DER DRITTE PUNKT WIRD FAST IMMER VERGESSEN

Wenn der Testlauf denselben Deduplikationsschlüssel schreibt wie der echte Lauf, hält die Idempotenzprüfung später den echten Vorgang für bereits verarbeitet — und überspringt ihn. Der Test hat dann den Produktionsfall zerstört, ohne eine einzige Fehlermeldung zu erzeugen. Präfix nicht vergessen.

BEISPIEL 2.1 — EIN SZENARIO, DREIMAL AUSGELÖST

Aufbau: Webhook aus dem Onlineshop → Kunde im CRM anlegen → Willkommensmail → Aufgabe für den Vertrieb.

Was tatsächlich passierte: Der Shop wiederholt Webhooks dreimal, wenn er nicht innerhalb von fünf Sekunden eine Bestätigung erhält. Das Szenario brauchte durch den langsamen CRM-Aufruf sieben Sekunden bis zur Bestätigung. Ergebnis nach einem Monat: 340 Kunden dreifach im CRM, 1.020 Willkommensmails, 1.020 Vertriebsaufgaben. Aufgefallen ist es dem Vertrieb, nicht der IT.

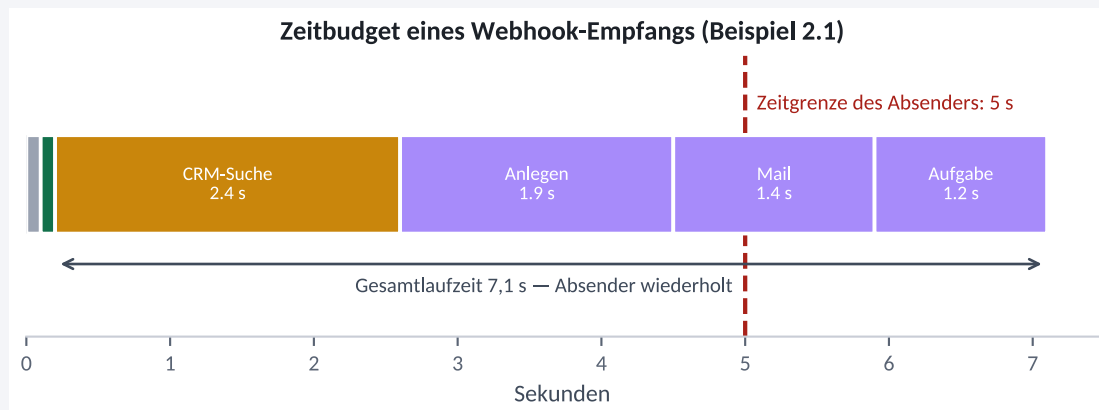


Abb. 2.5 Das Zeitbudget des Falls: Die Bestätigung kommt nach 7,1 Sekunden, der Absender wartet 5. Zwischen Sekunde 5 und 7,1 entsteht der zweite und dritte Lauf — ohne dass irgendwo ein Fehler protokolliert wird.

Zwei Fehler, zwei Korrekturen. Erstens die Bestätigung: Der Webhook-Empfang wird sofort bestätigt, die Verarbeitung läuft danach — die meisten Plattformen bieten dafür eine Einstellung oder ein vorgeschaltetes Speichern. Zweitens die Idempotenz: Vor dem Anlegen wird im CRM nach der Shop-Kundennummer gesucht; ist sie vorhanden, endet der Ablauf.

Die Lehre: Die zweite Korrektur allein hätte gereicht, um den Schaden zu verhindern. Die erste allein nicht — es gibt genug andere Gründe für einen zweiten Lauf. **Idempotenz ist die Absicherung, die unabhängig von der Ursache wirkt.**

2.5 Laufzeit, Kontingent und Grenzen

Jede Plattform hat drei Grenzen, die vor dem Bauen bekannt sein müssen: die maximale Laufzeit eines Durchgangs, die Zahl der Vorgänge je Zeitraum und die maximale Datenmenge je Schritt. Sie werden erst dann sichtbar, wenn die Produktionsmenge auf das Szenario trifft.

GRENZE	TYPISCHE GRÖSSENORD- NUNG	WAS ZU TUN IST
Laufzeit je Durchgang	40 Sekunden bis 15 Minuten	Lange Listen in Stapel teilen und je Stapel einen eigenen Durchgang starten
Vorgänge je Monat	je nach Tarif	Iterator gegen die maximale Listenlänge rechnen, nicht gegen die durchschnittliche
Gleichzeitige Durchgänge	1 bis 40	Bei 1 staut sich alles hinter einem langsamen Aufruf — Warteschlange einplanen
Datenmenge je Schritt	1 bis 100 MB	Dateien nicht durch das Szenario schleusen, sondern Verweise übergeben
Aufbewahrung der Protokolle	7 bis 90 Tage	Fehlerablage in ein eigenes System schreiben, nicht auf Protokolle verlassen

Die Rechnung vor dem Bauen

Vorgänge je Lauf = 1 (Auslöser)
 + Schritte vor dem Iterator
 + Listenlänge × Schritte im Iterator
 + Schritte nach dem Aggregator

Beispiel: Tageslauf über 120 Bestellungen mit je 4 Schritten
 = 1 + 3 + 120 × 4 + 2 = 486 Vorgänge je Lauf
 × 22 Arbeitstage = 10.692 Vorgänge je Monat

Gegenprobe mit der Spitzenmenge (Monatsende, 400 Bestellungen):
 = 1 + 3 + 400 × 4 + 2 = 1.606 Vorgänge an einem Tag

Die Gegenprobe mit der Spitzenmenge ist der Teil, der entscheidet. Tarife werden nach Monatssummen gewählt und an Spitzentagen gerissen — und ein gerissenes Kontingent hält am Monatsende genau den Lauf an, der am wichtigsten ist.

ÜBUNG 2 — IDEMPOTENZ HERSTELLEN

Gegeben ist dieses Szenario:

Auslöser: Neue Zeile in der Tabelle "Urlaubsanträge"

1. Verbleibende Urlaubstage aus der Personaldatenbank lesen
2. Prüfen, ob genug Tage vorhanden sind
3. Genehmigungsmail an die Führungskraft senden
4. Tage in der Personaldatenbank abziehen
5. Eintrag im Teamkalender anlegen
6. Bestätigungsmail an den Antragsteller

1. Welche drei Schritte richten bei einem zweiten Lauf konkreten Schaden an? Beschreibe den Schaden je Schritt.
2. Welchen Deduplikationsschlüssel wählst du, und wo speicherst du ihn?
3. Baue die Prüfung ein — an welcher Stelle genau, und warum nicht an einer anderen?
4. Die Reihenfolge ist in einem Punkt riskant. Welchem, und wie ordnest du um?
5. Schritt 4 scheitert, weil die Personaldatenbank nicht erreichbar ist. In welchem Zustand ist das System, und was tust du: Wiederaufnahme oder Kompensation?
6. **Zusatz:** Entwirf die Testauslösung nach 2.4 für dieses Szenario — welche drei Außenwirkungen musst du umleiten, und wie lautet dein Testpräfix?

KERNAUSSAGEN

- Filter früh, Verzweigungen sich ausschließend und mit Sonst-Zweig, Iteratoren mit Blick auf das Kontingent, Aggregator nicht vergessen.
- Idempotenz ist Normalbetrieb, nicht Sonderfall — Wiederholungen kommen aus vielen Richtungen.
- Ein Deduplikationsschlüssel aus dem Quellsystem, im Zielsystem gespeichert und dort eindeutig; Prüfen und Schreiben sind nicht atomar.
- Nach außen wirkende Schritte gehören ans Ende des Ablaufs — das erspart vier von fünf Kompensationen.
- Fachliche Benennung, ein Zweck je Szenario, dokumentierte Sonderfälle, Testauslösung mit eigenem Schlüsselpräfix.
- Laufzeit, Vorgänge und Datenmenge vor dem Bauen gegen die Spitzenmenge rechnen.

Selbsttest Modul 2

1. Was macht einen guten Deduplikationsschlüssel aus?
 - a) Er wird beim Start des Szenarios erzeugt.
 - b) Er stammt stabil aus dem Quellsystem und ist im Zielsystem eindeutig.
 - c) Er enthält den Zeitstempel der Ausführung.
 - d) Er wird aus mehreren Feldern zusammengesetzt.

2. Warum gehören nach außen wirkende Schritte ans Ende?

- a) Weil sie langsamer sind.
- b) Weil ein Abbruch davor keinen sichtbaren Zwischenzustand hinterlässt.
- c) Weil sie mehr Kontingent verbrauchen.
- d) Weil Plattformen das vorschreiben.

MODUL 3

Webhooks und API-Grundlagen

Dies ist das technischste Modul des Tracks und dasjenige mit dem größten Prüfungsgewicht. Du musst keine Zeile Code schreiben — aber du musst eine Dokumentation lesen, einen Statuscode richtig einordnen und wissen, warum eine geheime Adresse kein Schutz ist.

ABSCHNITT	INHALT	ZEIT
3.1	Anatomie einer Anfrage — Methode, Adresse, Kopfzeilen, Rumpf	20 Min
3.2	Statuscodes nach Reaktion sortiert	25 Min
3.3	Authentifizierung und der Umgang mit Zugangsdaten	25 Min
3.4	Webhooks empfangen und absichern	30 Min
3.5	Kontingente, Wartezeiten, Seitenweise-Abruf	25 Min
3.6	Eine Dokumentation lesen — in acht Schritten	20 Min
Übung 3	Statuscodes zuordnen, Abrufstrategie entwerfen	55 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du liest eine API-Dokumentation und baust daraus einen funktionierenden Aufruf.
- Du ordnest Statuscodes richtig ein und leitest daraus die Fehlerbehandlung ab.
- Du wählst ein Authentifizierungsverfahren begründet und gehst mit Zugangsdaten sicher um.
- Du richtest Webhooks sicher ein — einschließlich Signaturprüfung und Eingangsvalidierung.
- Du gehst mit Kontingentgrenzen und Seitenweise-Abruf um und rechnest beides vorher aus.

3.1 Das Nötigste über HTTP

Jeder Aufruf besteht aus **Methode**, **Adresse**, **Kopfzeilen** und manchmal einem **Rumpf**. Die Antwort besteht aus **Statuscode**, **Kopfzeilen** und meist einem **Rumpf**. Mehr ist für diesen Track nicht nötig.



Abb. 3.1 Vier Teile hin, drei Teile zurück. Wer eine Dokumentation liest, sucht genau diese sieben Felder — alles andere ist Beiwerk.

METHODE	BEDEUTUNG	WIEDERHOLBAR?
GET	etwas lesen	Ja, unbedenklich
POST	etwas anlegen oder auslösen	Nein — zweimal heißt zweimal
PUT / PATCH	etwas ganz oder teilweise ändern	Meist ja — derselbe Zielzustand
DELETE	etwas löschen	Ja, im Ergebnis

Die letzte Spalte ist die praktisch wichtigste: Ein automatischer Wiederholungsversuch ist bei **GET** und **PUT** harmlos und bei **POST** genau der Vorgang, der doppelte Rechnungen erzeugt. Wer Wiederholung einschaltet, muss wissen, welche Methode dahintersteht.

WARUM PUT WIEDERHOLBAR IST UND POST NICHT

Der Unterschied liegt nicht in der Technik, sondern in der Bedeutung. POST heißt „lege etwas Neues an“ — zweimal ausgeführt entstehen zwei Dinge. PUT heißt „bringe die Ressource mit dieser Kennung in diesen Zustand“ — zweimal ausgeführt ist der Zustand derselbe. Wo eine API ein PUT auf eine von dir gewählte Kennung anbietet, ist das der bequemste Weg zur Idempotenz aus Modul 2: Du brauchst dann keine eigene Prüfung mehr.

3.2 Statuscodes und was sie dir sagen

DIE EINZIGE UNTERSCHIEDUNG, DIE DU DIR MERKEN MUSST

2xx Erfolg	4xx Dein Fehler	429 Zu viele Anfragen	5xx Fehler des Anbieters
200 OK 201 Created 202 Accepted 204 No Content weiter im Ablauf	400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 409 Conflict 422 Unprocessable Wiederholen hilft nie	429 Too Many Requests Warten, dann wiederholen	500 Internal 502 Bad Gateway 503 Unavailable 504 Timeout Wiederholen mit Abstand

409 ist der Sonderfall: kein Fehler, sondern der Beweis, dass deine Idempotenzprüfung greift — verzweige darauf, statt abzubrechen.

Abb. 3.2 Statuscodes nach Reaktion sortiert, nicht nach Nummer. Wer bei 4xx wiederholt, verbrennt Kontingent und findet den Fehler nie.

CODE	BEDEUTUNG	DEINE REAKTION
200 / 201	erfolgreich	weiter
202	angenommen, aber noch nicht verarbeitet	Ergebnis später abholen — nicht als Erfolg werten
400	Anfrage fehlerhaft	Nicht wiederholen. Deine Daten sind falsch.
401	nicht angemeldet	Nicht wiederholen. Zugangsdaten oder Token prüfen.
403	angemeldet, aber nicht berechtigt	Nicht wiederholen. Rechte prüfen.
404	nicht gefunden	Nicht wiederholen — außer der Datensatz entsteht erst noch.
409	Konflikt, existiert bereits	Oft ein <i>gutes</i> Zeichen: deine Idempotenz greift.
422	Format richtig, Inhalt unzulässig	Nicht wiederholen. Fachlicher Fehler in den Daten.
429	zu viele Anfragen	Wiederholen — mit Wartezeit, siehe 3.5.
5xx	Fehler beim Anbieter	Wiederholen mit wachsendem Abstand.

Die einzige Unterscheidung, die du dir merken musst: **4xx ist dein Fehler und wird durch Wiederholen nicht besser. 429 und 5xx sind vorübergehend und werden es.** Wer bei 400 wiederholt, verbrennt Kontingent und findet den Fehler nie.

DIE AUSNAHME, DIE MAN KENNEN MUSS: 404 NACH DEM ANLEGEN

Manche Systeme antworten auf `POST` sofort mit `201`, legen den Datensatz aber erst Sekundenbruchteile später wirklich an. Ein unmittelbar folgendes `GET` auf die neue Kennung antwortet dann mit `404` — obwohl nichts kaputt ist. Das ist der einzige Fall, in dem ein `404` wiederholt werden darf: zwei bis drei Versuche mit einer Sekunde Abstand. Steht in der Dokumentation ein Hinweis auf „eventual consistency“, ist genau das gemeint.

3.3 Authentifizierung

API-Schlüssel

Eine Zeichenkette in einer Kopfzeile.

Einrichtung	Minuten
Gültigkeit	oft unbegrenzt
Umfang	volle Rechte
Entzug	nur global

Bequem und gefährlich

Bearer-Token

Wie ein Schlüssel, aber befristet.

Einrichtung	Minuten
Gültigkeit	Stunden bis Tage
Umfang	meist voll
Entzug	durch Ablauf

Braucht Erneuerung

OAuth

Anmeldung im Namen eines Kontos.

Einrichtung	Stunden
Gültigkeit	erneuerbar
Umfang	begrenzbar (Scopes)
Entzug	einzeln möglich

Sauberste Variante

Abb. 3.3 Drei Verfahren, drei Risikoprofile. Die entscheidende Zeile ist „Entzug“: Was passiert, wenn der Schlüssel abhandenkommt?

API-Schlüssel — eine Zeichenkette in einer Kopfzeile. Einfach, verbreitet und gefährlich, weil sie oft dauerhaft gültig ist und volle Rechte hat.

Bearer-Token — wie ein Schlüssel, aber meist zeitlich begrenzt. Die Erneuerung muss eingeplant werden; ein abgelaufener Token ist die häufigste Ursache eines stillen Ausfalls (5.5).

OAuth — Anmeldung im Namen eines Nutzers, mit begrenztem Umfang und Erneuerung. Aufwendiger einzurichten, aber die sauberste Variante, weil die Rechte auf das Nötige begrenzt werden können.

DREI REGELN FÜR ZUGANGSDATEN

1. **Nie ins Szenario schreiben.** Plattformen haben Verbindungen oder Tresore. Ein Schlüssel im Klartext in einem Modul steht in jedem Ausführungsprotokoll und in jedem Export.
2. **Eigenes Konto je Automatisierung.** Nicht der persönliche Zugang der Kollegin, die vor zwei Jahren gegangen ist. Ein technisches Konto überlebt Personalwechsel und lässt sich einzeln entziehen.
3. **Kleinstmögliche Rechte.** Wenn das Szenario nur liest, bekommt es keinen Schreibzugriff. Das begrenzt den Schaden, wenn der Schlüssel abhandenkommt.

Was zu tun ist, wenn ein Schlüssel abhandengekommen ist

1. Schlüssel sofort im Quellsystem widerrufen — nicht erst den Ersatz einrichten.
2. Neuen Schlüssel erzeugen und in der Verbindung der Plattform hinterlegen.
3. Ausführungsprotokolle des betroffenen Zeitraums auf fremde Aufrufe prüfen.

4. Prüfen, wo der Schlüssel überall lag: Exporte, Screenshots, Chatnachrichten, Sicherungskopien von Szenarien.
5. Festhalten, wie es passiert ist — und ob eine der drei Regeln oben es verhindert hätte.

3.4 Webhooks empfangen und absichern

Beim Empfang gelten drei Dinge, die in Anleitungen selten stehen:

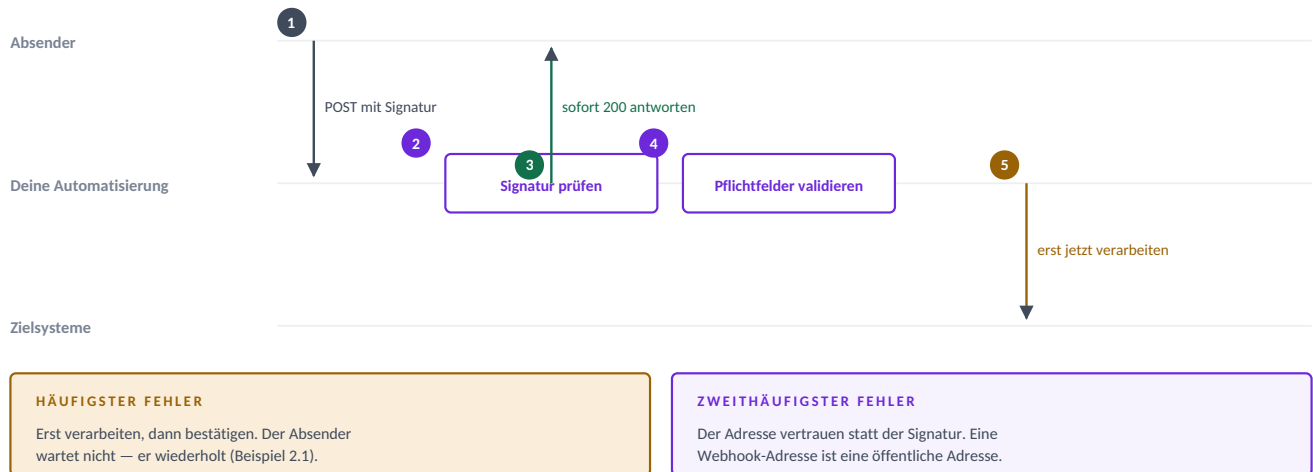


Abb. 3.4 Reihenfolge beim Webhook-Empfang: prüfen, bestätigen, validieren, verarbeiten. Schritt 3 steht bewusst vor Schritt 4.

Schnell bestätigen

Antworte mit `200`, sobald du die Daten hast — nicht, nachdem du sie verarbeitet hast. Die meisten Absender warten nur wenige Sekunden und wiederholen sonst. Das ist die Ursache aus Beispiel 2.1.

Signatur prüfen, nicht auf die geheime Adresse vertrauen

Eine Webhook-Adresse ist eine öffentliche Internetadresse. Wer sie kennt — aus einem Protokoll, einem Screenshot, einem Export — kann beliebige Daten hineinschicken. Seriöse Anbieter signieren jeden Aufruf; prüfe die Signatur, bevor du irgendetwas mit den Daten tust. Wenn dein Anbieter nicht signiert, nimm mindestens ein gemeinsames Geheimnis in einer Kopfzeile auf und prüfe es.

Nichts blind vertrauen

Auch ein signierter Webhook enthält Daten, keine Anweisungen. Prüfe Pflichtfelder und Wertebereiche, bevor du sie weiterverwendest. Das gilt besonders, wenn die Daten später in einen KI-Schritt laufen (6.1): Freitext aus einer fremden Quelle ist genau der Ort, an dem eingeschleuste Anweisungen stehen könnten.

Signatur oder gemeinsames Geheimnis	Fremde Absender, die die Adresse kennen
Pflichtfelder vorhanden	Abbruch mitten im Ablauf, halb angelegte Datensätze
Wertebereiche plausibel (Betrag > 0, Datum nicht in der Zukunft)	Fachliche Fehler, die still ins Zielsystem laufen
Erwarteter Ereignistyp	Verarbeitung von Ereignissen, für die das Szenario nicht gebaut ist
Deduplikationsschlüssel vorhanden	Doppelverarbeitung bei Wiederholung durch den Absender
Alter des Aufrufs (Zeitstempel nicht älter als wenige Minuten)	Wiedereinspielen alter, abgefangener Aufrufe

3.5 Kontingente und Seitenweise-Abruf

Kontingentgrenzen begrenzen Aufrufe je Zeiteinheit. Bei Überschreitung kommt 429, oft mit einer Kopfzeile, die die Wartezeit nennt. Die richtige Reaktion ist warten und wiederholen, mit wachsendem Abstand — 1, 2, 4, 8 Sekunden, höchstens fünf Versuche. Wer sofort wiederholt, verlängert die Sperre.

60 AUFRUFE JE MINUTE — 500 KONTAKTE IN EINEM ITERATOR

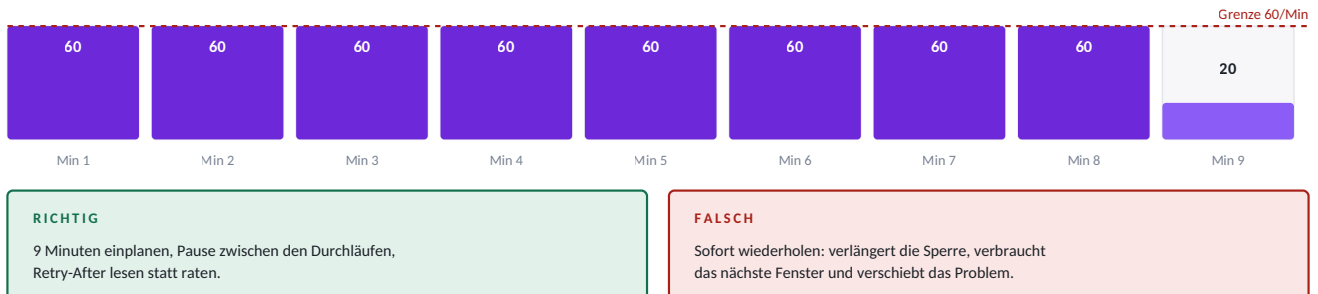


Abb. 3.5 Ein Iterator über 500 Kontakte ist bei 60 Aufrufen je Minute kein Sekunden-, sondern ein Neun-Minuten-Vorgang. Das gehört in die Planung.

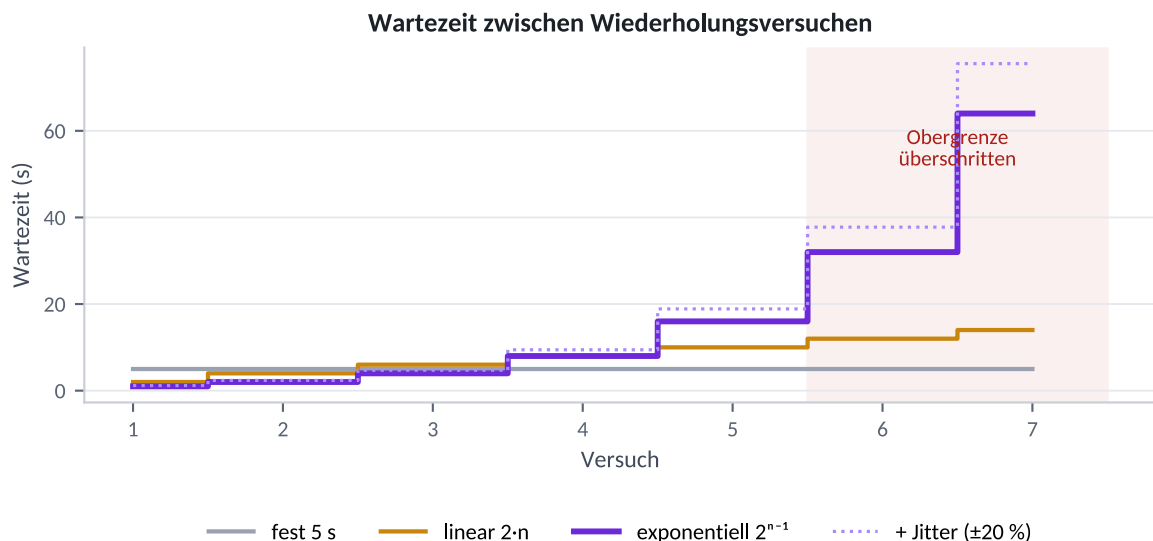


Abb. 3.6 Vier Wartestrategien im Vergleich. Der exponentielle Abstand gibt dem überlasteten System Zeit, sich zu erholen; der Zufallsanteil (Jitter) verhindert, dass viele gleichzeitig blockierte Abläufe im selben Moment erneut anfragen.

Seitenweise-Abruf liefert lange Listen in Portionen. Wer nur die erste Seite verarbeitet, bekommt ein Szenario, das monatelang unauffällig 100 von 3.400 Datensätzen bearbeitet. Prüfe in der Dokumentation, ob die Antwort einen Hinweis auf weitere Seiten enthält, und baue die Schleife.

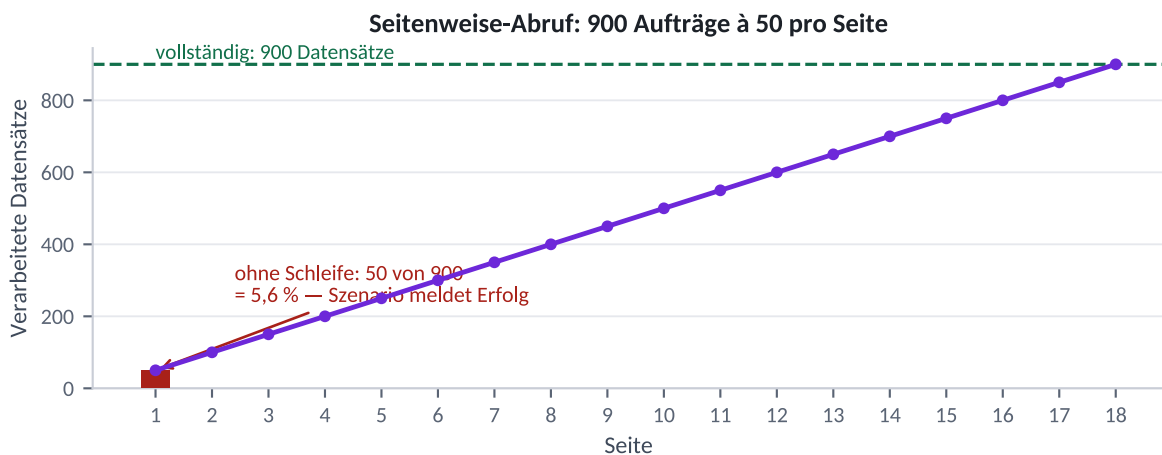


Abb. 3.7 Der Unterschied zwischen einer Seite und allen Seiten. Das Tückische: Ohne Schleife meldet das Szenario Erfolg — es gibt keinen Fehler, nur ein unvollständiges Ergebnis.

VERFAHREN	WIE DU WEITERBLÄTTERST	WORAUF ZU ACHTEN IST
Seitennummer	?page=2 bis die Antwort leer ist	Verschiebt sich, wenn während des Abrufs Datensätze hinzukommen
Versatz und Menge	?offset=100&limit=50	Gleiches Problem; zusätzlich langsam bei großen Versätzen
Zeiger (Cursor)	Kennung aus der Antwort in die nächste Anfrage	Robust — bevorzugen, wo angeboten
Verweis auf die nächste Seite	Adresse aus dem Feld next	Adresse unverändert übernehmen, nicht selbst zusammenbauen

BEISPIEL 3.1 — EINE DOKUMENTATION LESEN

```
POST /v2/contacts
Authorization: Bearer <token>
Content-Type: application/json
Rate limit: 60 requests/minute

{
  "email":      "string, required, unique",
  "first_name": "string, optional",
  "tags":       ["string"],
  "external_id": "string, optional, unique"
}

201 Created  -> {"id":"c_8842","email":"..."}
409 Conflict -> {"error":"email already exists","id":"c_1201"}
429          -> Retry-After: 30
```

Was hier drinsteht, wenn man es liest:

`external_id` ist optional *und eindeutig* — das ist dein Deduplikationsschlüssel aus Modul 2, frei Haus. Setze ihn immer.

409 liefert die vorhandene `id` mit. Damit ist der Konflikt kein Fehler, sondern ein verwertbares Ergebnis: „existiert schon, hier ist die Kennung“ — verzweige darauf, statt abzurechnen.

60 Aufrufe je Minute heißt: Ein Iterator über 500 Kontakte braucht mindestens neun Minuten. Plane eine Pause zwischen den Durchläufen ein, statt in die Sperre zu laufen.

Retry-After: 30 nennt die Wartezeit. Nimm diese Zahl, statt selbst zu raten.

3.6 Eine Dokumentation lesen — in acht Schritten

Die folgende Reihenfolge beantwortet alle Fragen, die du beim Bauen brauchst, und überspringt alles andere. Sie dauert bei einer unbekannten API rund zwanzig Minuten.

SCHRITT	SUCHE NACH	WAS DU DAMIT ENTSCHIEDEST
1	Authentifizierung	Welches Verfahren, welcher Kontotyp, wie erneuert wird
2	Basisadresse und Version	Ob es eine Testumgebung gibt
3	Kontingentgrenzen	Wie viele Vorgänge je Zeitraum möglich sind (2.5)
4	Der Aufruf, den du brauchst — Methode und Pflichtfelder	Ob POST oder PUT, und damit die Wiederholungsstrategie
5	Ein Feld für eine Fremdkennung	Ob Idempotenz frei Haus kommt
6	Fehlerantworten	Welche Codes fachlich und welche technisch sind
7	Seitenweise-Abruf	Ob und wie eine Schleife gebraucht wird
8	Webhooks: Signatur, Wiederholungsverhalten, Zeitgrenze	Wie der Empfang abgesichert wird (3.4)

DIE FRAGE, DIE IN KEINER DOKUMENTATION STEHT

„Was passiert, wenn ich denselben Aufruf zweimal sende?“ — diese Frage beantwortet fast keine Dokumentation ausdrücklich. Sie lässt sich aber in fünf Minuten in der Testumgebung beantworten, und die Antwort bestimmt den ganzen Aufbau. Probiere es aus, bevor du baust.

ÜBUNG 3 — AUFRUFE UND FEHLERREAKTIONEN

Teil A: Ordne jeder Situation den wahrscheinlichsten Statuscode zu und entscheide: wiederholen oder nicht.

1. Das Token ist abgelaufen.
2. Ein Pflichtfeld fehlt im Rumpf.
3. Der Anbieter hat eine Störung.
4. Der Datensatz mit dieser E-Mail existiert bereits.
5. Das Szenario hat in dieser Minute 200 Aufrufe gemacht.
6. Die E-Mail-Adresse ist syntaktisch gültig, aber der Anbieter lehnt Wegwerfadressen ab.

Teil B: Ein Szenario liest jede Stunde neue Aufträge aus einer API und legt sie im ERP an. Die API liefert 50 Datensätze pro Seite und begrenzt auf 100 Aufrufe pro Stunde. An einem Montagmorgen liegen 900 neue Aufträge vor.

1. Wie viele Aufrufe braucht allein das Lesen? Reicht das Kontingent?
2. Was passiert, wenn du die Seitenweise-Schleife vergisst?
3. Entwirf die Wiederholungsstrategie für den Fall, dass mitten im Durchlauf ein 429 kommt.
4. Welche Kennung nutzt du, damit ein Wiederanlauf keine Aufträge doppelt anlegt?
5. **Zusatz:** Das ERP antwortet auf das Anlegen gelegentlich mit 504, legt den Auftrag aber trotzdem an. Wie verhinderst du Doppelanlagen, obwohl du keine verlässliche Antwort bekommst?

KERNAUSSAGEN

- POST ist nicht wiederholbar — das entscheidet über deine Wiederholungsstrategie.
- 4xx wird durch Wiederholen nicht besser; 429 und 5xx schon.
- 409 ist oft ein Erfolg deiner Idempotenz, kein Fehler.
- Zugangsdaten in den Tresor, eigenes technisches Konto, kleinstmögliche Rechte.
- Webhooks: schnell bestätigen, Signatur prüfen, Inhalte validieren.
- Kontingent und Seitenweise-Abruf gehören in jede Planung — gegen die Spitzenmenge gerechnet.
- Acht Schritte durch jede Dokumentation; die neunte Frage beantwortest du selbst in der Testumgebung.

Selbsttest Modul 3

1. Bei welchem Statuscode ist ein automatischer Wiederholungsversuch sinnvoll?

- a) 400
- b) 403
- c) 429
- d) 422

2. Warum reicht eine geheime Webhook-Adresse als Schutz nicht aus?

- a) Weil sie zu lang ist.
- b) Weil sie in Protokollen und Exporten auftaucht und dann jeder Daten einschicken kann.
- c) Weil Plattformen sie regelmäßig ändern.
- d) Weil sie nur mit OAuth funktioniert.

MODUL 4

Datenzuordnung

Fehler in der Datenzuordnung brechen selten etwas ab. Sie schreiben etwas Falsches ins Zielsystem, und das fällt Wochen später auf — beim Jahresabschluss, bei einer Auswertung, bei einer Kundenbeschwerde. Deshalb verdient dieses Modul mehr Aufmerksamkeit, als sein Thema vermuten lässt.

ABSCHNITT	INHALT	ZEIT
4.1	Warum Zuordnungsfehler nicht abbrechen	10 Min
4.2	Die fünf Fallen: Typen, Datum, Zahlen, Leerwerte, Zeichen	35 Min
4.3	Verschachtelte Strukturen	20 Min
4.4	Die Zuordnungstabelle als Arbeitsmittel	20 Min
4.5	Abgleich statt Anlegen — Datensätze zusammenführen	25 Min
Übung 4	Eine vollständige Zuordnung entwerfen	50 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du erkennst Typkonflikte zwischen Systemen, bevor sie im Zielsystem landen.
- Du behandelst Datum, Zeitzone, Zahlenformat und Leerwerte bewusst statt zufällig.
- Du zerlegst und baust verschachtelte Strukturen zuverlässig.
- Du legst für jedes Feld eine Regel fest — auch für den Fall, dass es fehlt.
- Du unterscheidest Anlegen von Abgleichen und wählst den passenden Schlüssel.

4.1 Der stille Fehler

Fehler bei der Datenzuordnung brechen selten den Ablauf ab. Sie schreiben etwas Falsches ins Zielsystem, und das fällt Wochen später auf. Damit gehören sie zur Fehlerklasse mit der niedrigsten Entdeckungsquote — und nach der Rechnung aus 1.3 zur teuersten.

Der Grund liegt in der Arbeitsweise der Werkzeuge: Eine Zuordnung ist in den meisten Plattformen eine Linie zwischen zwei Feldern. Die Linie prüft nichts. Sie überträgt, was da ist — auch wenn es das Falsche ist.

JEDES FELD DURCHLÄUFT DIESELBEN FÜNF FRAGEN



Die Zuordnungstabelle ist genau diese Pipeline, einmal je Feld aufgeschrieben — vor dem ersten Klick.

Abb. 4.1 Datenzuordnung ist keine Verbindung zweier Felder, sondern eine Kette von fünf Entscheidungen je Feld.

4.2 Die fünf Fallen

Falle 1 — Typen

Eine Zeichenkette `"42"` ist keine Zahl `42`. Manche Systeme wandeln stillschweigend um, andere lehnen ab, dritte speichern die Zeichenkette und du wunderst dich, warum die Summe nicht stimmt. Besonders heikel sind Wahrheitswerte: `true`, `"true"`, `1`, `"ja"`, `"Y"` — jedes System bevorzugt eine andere Schreibweise.

QUELLWERT	WIRD OFT GELESEN ALS	SICHERE ZUORDNUNG
"42"	Zahl 42 oder Text „42“	Ausdrücklich in Zahl wandeln
"0042"	Zahl 42 — führende Null weg	Als Text behalten, wenn es eine Kennung ist
"ja" / "Y"	Text, nicht Wahrheitswert	Auf <code>true</code> / <code>false</code> abbilden, Rest in die Prüfliste
"1,5"	1 (nach dem Komma abgeschnitten)	Trennzeichen vorher normalisieren
"+41 79 ..."	Zahl oder Text, je nach System	Immer Text; Telefonnummern sind keine Zahlen

Falle 2 — Datum und Zeitzone

Das teuerste Feld in jeder Integration. Drei Regeln, die alles abdecken:

- **Übertrage in ISO 8601 mit Zeitzone:** 2026-08-30T14:05:00+02:00. Nicht 30.08.2026 14:05 — bei diesem Format weiß niemand, welche Zeitzone gemeint ist.
- **Rechne in UTC,** zeige lokal. Ein Szenario, das um Mitternacht läuft und lokale Zeitstempel vergleicht, liefert zweimal im Jahr falsche Ergebnisse.
- **Nur-Datum-Felder haben keine Zeitzone.** Ein Geburtstag ist der 3. Mai, nicht der 3. Mai 00:00 UTC — sonst wird er in manchen Zeitzonen zum 2. Mai.

EIN GEBURTSTAG, DREI SYSTEME		
Quelle (Formular)	03.05.1988	keine Zeitzone genannt
Übertragen als	1988-05-03T00:00:00Z	still zu UTC ergänzt
Ziel in Sao Paulo	02.05.1988 21:00	ein Tag zu früh
Richtig übertragen	1988-05-03 (Nur-Datum)	keine Zeitzone, kein Fehler
DREI REGELN, DIE ALLES ABDECKEN		
1 · ISO 8601 mit Zeitzone übertragen 2 · in UTC rechnen, lokal anzeigen 3 · Nur-Datum-Felder ohne Zeitzone		

Abb. 4.2 Das teuerste Feld jeder Integration. Der Fehler entsteht nicht beim Übertragen, sondern beim stillen Ergänzen einer Zeitangabe.

Falle 3 — Zahlenformate

1.234,56 und 1,234.56 sind dieselbe Zahl in zwei Schreibweisen — und werden von einem System, das die andere erwartet, als 1,23456 oder als 1234 gelesen. Übertrage Zahlen immer als reine Zahl ohne Trennzeichen, formatiere erst bei der Anzeige. Bei Geldbeträgen kommt hinzu: Zwei Nachkommastellen wollen exakt sein, weshalb viele Systeme intern in Cent rechnen.

RUNDUNG IST EINE FACHLICHE ENTSCHEIDUNG

Wenn du einen Betrag in Cent umrechnest, entscheidest du über Rundung: kaufmännisch, immer abwärts, oder Bankrundung. Bei einzelnen Beträgen ist der Unterschied ein Cent. Bei 20.000 Positionen im Jahr ist er eine Abweichung, die in der Buchhaltung auffällt und dann dir zugeordnet wird. Halte die gewählte Regel in der Zuordnungstabelle fest.

Falle 4 — Leerwerte



Abb. 4.3 Vier Zustände, vier Regeln. „Automatisch“ ist keine Regel — je Feld ist festzulegen, was bei leer geschieht.

Der klassische Schaden: Ein fehlendes Rabattpfeld wird als 0 übertragen und überschreibt im Zielsystem einen bestehenden Rabatt von 15 Prozent. Lege für jedes Feld fest: *Wenn leer, dann nicht übertragen* oder *wenn leer, dann diesen Vorgabewert*. Nie „automatisch“.

Falle 5 — Zeichen und Länge

Umlaute, Emojis und Anführungszeichen brechen Integrationen, die davon ausgehen, dass jedes Zeichen ein Byte ist. Und Zielsysteme kürzen zu lange Werte oft ohne Fehlermeldung: Aus einem 300 Zeichen langen Notizfeld werden 255, und die letzten 45 sind weg. Prüfe die Feldlängen der Zielsysteme und kürze bewusst, mit einem sichtbaren Hinweis.

FALLE	WORAN DU SIE IM BETRIEB ERKENNST
Typen	Summen stimmen nicht; Filter greifen nicht, obwohl der Wert stimmt
Datum/Zeitzone	Datensätze mit Datum einen Tag daneben; zweimal im Jahr Häufung von Fehlern
Zahlenformate	Beträge um Faktor 100 oder 1.000 daneben
Leerwerte	Bestehende Werte im Ziel verschwinden nach einem Lauf
Zeichen/Länge	Texte enden mitten im Satz; Sonderzeichen als Fragezeichen

4.3 Verschachtelte Strukturen

JSON-Antworten sind selten flach. Zwei Muster reichen für fast alles:

```
{
  "order_id": "A-2026-1188",
  "customer": { "name": "Beispiel GmbH", "vat": "DE123456789" },
  "items": [
    { "sku": "X-100", "qty": 2, "price": 49.90 },
    { "sku": "Y-200", "qty": 1, "price": 129.00 }
  ]
}
```

Objekt: Direkter Zugriff auf das Unterfeld — `customer.name`. Vorsicht, wenn `customer` fehlen kann; dann bricht der Zugriff ab. Prüfe erst die Existenz.

Liste: Hier brauchst du einen Iterator. Und die Entscheidung: Willst du je Position eine Zeile im Zielsystem, oder eine Zeile je Auftrag mit zusammengefassten Positionen? Das ist keine technische, sondern eine fachliche Frage — und sie gehört in die Prozessbeschreibung aus Modul 1.

ENTSCHEIDUNG	WANN RICHTIG	FOLGE FÜR DAS SZENARIO
Eine Zeile je Position	Wenn Positionen einzeln ausgewertet, bepreist oder kommissioniert werden	Iterator; Vorgangszahl × Positionen; Auftragsnummer in jeder Zeile mitführen
Eine Zeile je Auftrag	Wenn nur Summen und Status zählen	Aggregator; eine Zeile; Positionsdetails als Text oder Anhang
Beides	Wenn ein System Kopfdaten und ein anderes Positionen führt	Kopf zuerst anlegen, Kennung merken, dann Positionen mit dieser Kennung

DIE REIHENFOLGE BEI KOPF UND POSITIONEN

Wenn Kopf und Positionen getrennt angelegt werden, entsteht zwischen beiden Schritten ein Teilerfolg-Zustand aus 2.3: Der Auftrag existiert ohne Positionen. Zwei Absicherungen: Erstens den Kopf mit einem Status „unvollständig“ anlegen und erst nach den Positionen auf „offen“ setzen. Zweitens die Positionen idempotent schreiben, damit ein Wiederanlauf sie nicht verdoppelt — Schlüssel ist die Kombination aus Auftragsnummer und Positionsnummer, nicht die Artikelnummer.

4.4 Die Zuordnungstabelle

BEISPIEL 4.1 — EINE ZUORDNUNGSTABELLE

Vor jeder Integration entsteht diese Tabelle. Sie kostet zwanzig Minuten und spart Tage.

QUELLE	ZIEL	UMWANDLUNG	WENN LEER
order_id	external_ref	unverändert	Abbruch, Pflichtfeld
customer.vat	tax_id	Leerzeichen entfernen, Großschreibung	nicht übertragen
created	order_date	ISO 8601, nach UTC	Zeitpunkt des Laufs, mit Vermerk
total	amount_cents	× 100, kaufmännisch runden	Abbruch, Pflichtfeld
discount	discount_pct	unverändert	nicht übertragen
note	comment	auf 255 Zeichen kürzen	leerer Text

Die wichtigste Zeile ist `discount`: „nicht übertragen“ statt „0“. Genau dieser Unterschied hat in realen Projekten wiederholt bestehende Kundenrabatte gelöscht — ein Fehler, der erst bei der nächsten Rechnung auffällt und dann Kundengespräche kostet.

Die Tabelle hat eine zweite Funktion, die oft unterschätzt wird: Sie ist die einzige Unterlage, mit der eine andere Person das Szenario in einem Jahr nachvollziehen kann. Ein Screenshot der Zuordnungsoberfläche ist keine Dokumentation — er zeigt, *was* verbunden ist, nicht *warum*.

Drei Spalten, die oft fehlen und sich lohnen

ZUSATZSPALTE	INHALT	WOFÜR
Quelle der Regel	Wer hat entschieden, dass gekürzt wird — und wann	Damit die Regel später geändert werden kann, ohne zu raten
Verhalten bei Konflikt	Ziel hat bereits einen Wert: überschreiben, behalten, in die Prüfliste	Der häufigste Streitfall beim Abgleich (4.5)
Personenbezug	Ja/Nein je Feld	Grundlage für Löschkonzept und Auskunft (Anhang D)

4.5 Abgleich statt Anlegen

Die meisten Integrationen legen nicht nur an, sie gleichen ab: Kunde vorhanden? Dann aktualisieren. Nicht vorhanden? Dann anlegen. Dieser Ablauf hat zwei Entscheidungen, die vor dem Bauen fallen müssen.

Erstens: Welcher Schlüssel entscheidet über „vorhanden“?

ABGLEICHSSCHLÜSSEL	STÄRKE	SCHWÄCHE
Fremdkennung aus dem Quellsystem	eindeutig, stabil	nur vorhanden, wenn das Ziel ein solches Feld hat
E-Mail-Adresse	in vielen Systemen eindeutig	ändert sich; geteilte Postfächer erzeugen falsche Treffer
Name + Postleitzahl	immer verfügbar	unscharf; Schreibweisen, Umzüge, Namensänderungen
Steuernummer / Handelsregisternummer	sehr zuverlässig bei Firmen	oft nicht erfasst; nicht bei Privatpersonen

Zweitens: Was gilt bei Konflikt?

Wenn beide Seiten einen Wert haben und die Werte sich unterscheiden, gibt es drei Möglichkeiten — und die Antwort ist je Feld verschieden, nicht je System:

1. **Quelle gewinnt** — richtig für Felder, die im Quellsystem gepflegt werden (Bestellstatus, Liefermenge).
2. **Ziel gewinnt** — richtig für Felder, die im Zielsystem gepflegt werden (Vertriebsnotiz, Kundenbetreuer, Rabattstufe).
3. **Prüfliste** — richtig, wenn die Abweichung selbst die Information ist (andere Adresse, anderer Firmenname).

DER HÄUFIGSTE ZUORDNUNGSFEHLER IN BESTEHENDEN SYSTEMEN

„Quelle gewinnt“ für alle Felder ist die Voreinstellung, die am schnellsten gebaut ist — und die stillste Datenzerstörung, die es gibt. Sie überschreibt jede Pflege, die im Zielsystem stattgefunden hat, bei jedem Lauf aufs Neue. Wer nach der Einführung hört „unsere Notizen verschwinden immer wieder“, sucht genau hier.

ÜBUNG 4 — ZUORDNUNG ENTWERFEN

Aus einem Webformular kommen diese Daten in ein CRM:

```
{
  "submitted":    "30.08.2026 14:05",
  "full_name":    "Müller-Schmidt, Anna",
  "company":      "",
  "budget":       "10.000 - 25.000 EUR",
  "employees":    "50",
  "consent_news": "on",
  "message":      "Wir suchen eine Lösung für ... (620 Zeichen)",
  "utm_source":   null
}
```

Das CRM erwartet: `first_name`, `last_name`, `company` (max. 100), `budget_min` (Zahl), `budget_max` (Zahl), `company_size` (Zahl), `newsletter` (Wahrheitswert), `notes` (max. 500), `source`, `created_at` (ISO 8601 UTC).

1. Erstelle die vollständige Zuordnungstabelle nach dem Muster aus 4.1 — einschließlich der drei Zusatzspalten aus 4.4.
2. Bei drei Feldern ist die Umwandlung nicht eindeutig. Welche, und wie entscheidest du?
3. `full_name` in Vor- und Nachname zu zerlegen ist eine Falle. Nenne zwei Fälle, in denen deine Regel scheitert, und was du stattdessen tust.
4. Was passiert mit `message`, und wie stellst du sicher, dass die Kürzung auffällt?
5. `consent_news` fehlt, wenn der Haken nicht gesetzt war. Warum ist das gefährlicher als "off", und wie behandelst du es?
6. **Zusatz:** Der Kontakt existiert bereits im CRM, mit einer gepflegten Vertriebsnotiz und einer anderen Firmenangabe. Lege je Feld fest, wer gewinnt.

KERNAUSSAGEN

- Zuordnungsfehler brechen nicht ab, sie schreiben still Falsches — niedrigste Entdeckungsquote, höchste Folgekosten.
- Datum in ISO 8601 mit Zeitzone, rechnen in UTC, Nur-Datum ohne Zeitzone.
- Zahlen ohne Trennzeichen übertragen, Geld in kleinster Einheit, Rundungsregel festhalten.
- `null`, `""`, fehlend und `0` sind vier verschiedene Dinge.
- Je Feld eine Regel für den Leerfall — nie „automatisch“.
- Beim Abgleich je Feld festlegen, wer bei Konflikt gewinnt; „Quelle gewinnt“ für alles zerstört gepflegte Daten.
- Die Zuordnungstabelle vor dem Bauen kostet zwanzig Minuten und spart Tage.

Selbsttest Modul 4

1. Ein fehlendes Rabattfeld wird als `0` übertragen. Was ist die Folge?

- a) Nichts, `0` und leer sind gleichwertig.
- b) Ein bestehender Rabatt im Zielsystem wird überschrieben.
- c) Das Zielsystem lehnt den Datensatz ab.
- d) Der Wert wird ignoriert.

2. Welches Datumsformat gehört in eine Integration?

- a) 30.08.2026 14:05
- b) 2026-08-30T14:05:00+02:00
- c) 08/30/2026
- d) Das Format des Quellsystems, unverändert.

MODUL 5

Fehlerbehandlung und der Mensch im Ablauf

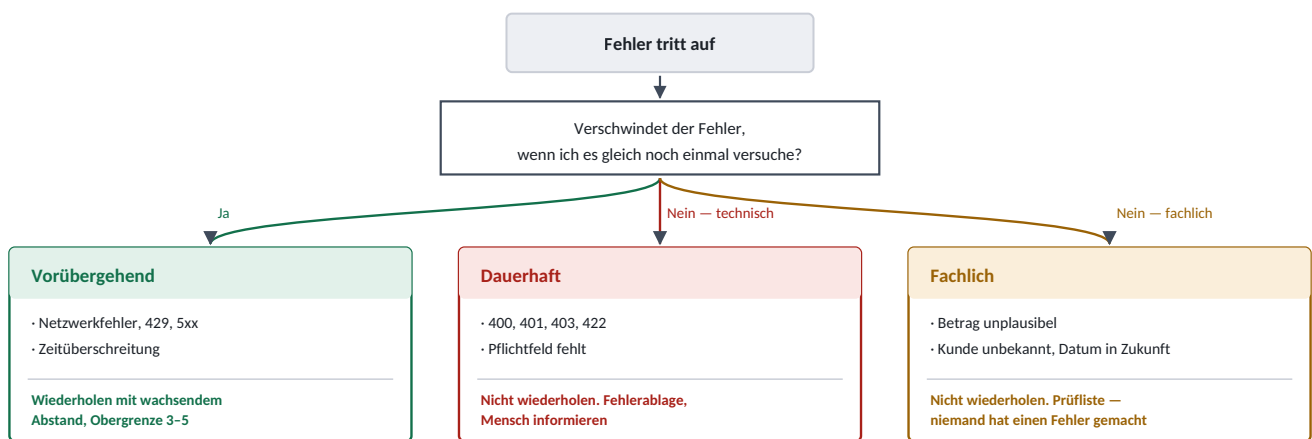
Bis hierher ging es darum, dass ein Ablauf funktioniert. Ab hier geht es darum, was geschieht, wenn er das nicht tut — und wer es merkt. Dieses Modul ist der wirtschaftlich wertvollste Teil des Tracks, weil es die Entdeckungsquote aus Modul 1 in die Höhe treibt.

ABSCHNITT	INHALT	ZEIT
5.1	Drei Fehlerklassen und drei verschiedene Reaktionen	25 Min
5.2	Wiederholung: Abstand, Obergrenze, Voraussetzung	20 Min
5.3	Fehlerablage und Alarmierung, die jemanden erreicht	25 Min
5.4	Fünf Kriterien für den Menschen im Ablauf	30 Min
5.5	Der stille Ausfall und der Wächter	15 Min
Übung 5	Fehlerbehandlungsplan für ein Preispflege-Szenario	50 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du unterscheidest vorübergehende, dauerhafte und fachliche Fehler und reagierst je Klasse richtig.
- Du baust Wiederholung mit wachsendem Abstand und Obergrenze statt endloser Versuche.
- Du richtest eine Fehlerablage und eine Alarmierung ein, die jemanden erreicht.
- Du entscheidest begründet, an welcher Stelle ein Mensch im Ablauf steht.
- Du erkennst den stillen Ausfall und baust den Wächter dagegen.

5.1 Drei Fehlerklassen



Die dritte Klasse wird von Plattformen selten unterstützt und ist trotzdem die häufigste im Betrieb.

Abb. 5.1 Eine einzige Frage trennt die drei Klassen — und die Antwort bestimmt, ob Wiederholung hilft, schadet oder am Problem vorbeigeht.

KLASSE	BEISPIELE	REAKTION
Vorübergehend	Netzwerkfehler, 429, 5xx, Zeitüberschreitung	Wiederholen mit wachsendem Abstand
Dauerhaft	400, 401, 403, 422, Pflichtfeld fehlt	Nicht wiederholen. In die Fehlerablage, Mensch informieren.
Fachlich	Betrag unplausibel, Kunde unbekannt, Datum in der Zukunft	Nicht wiederholen. In die Prüfliste — hier hat niemand einen Fehler gemacht, die Daten passen nur nicht.

Die dritte Klasse wird in Plattformen selten unterstützt und ist trotzdem die häufigste im Betrieb. Sie entsteht nicht durch einen technischen Defekt, sondern weil die Wirklichkeit nicht ins Muster passt — und genau dort braucht es einen Menschen.

DIE FACHLICHE KLASSE IST KEIN FEHLER, SONDERN EINE AUFGABE

Der Unterschied ist nicht akademisch, er ändert das Verhalten: Ein technischer Fehler geht an die Person, die das Szenario betreut. Ein fachlicher Fall geht an die Person, die den Prozess verantwortet — die Preispflege, die Buchhaltung, den Vertrieb. Wer beides in dieselbe Ablage schreibt, bekommt eine Liste, die niemand abarbeitet, weil jeder den anderen zuständig glaubt.

5.2 Wiederholung richtig einstellen

Drei Zahlen und eine Regel:

- **Wachsender Abstand:** 1, 2, 4, 8, 16 Sekunden statt fünfmal sofort. Ein überlastetes System erholt sich nicht, während du es weiter anrufst.
- **Obergrenze:** drei bis fünf Versuche. Danach ist es kein vorübergehender Fehler mehr.
- **Nur bei vorübergehenden Fehlern.** Eine pauschale Wiederholung über alle Fehler ist der verbreitetste Konfigurationsfehler — sie verbrennt Kontingent und verzögert die Meldung des eigentlichen Problems.

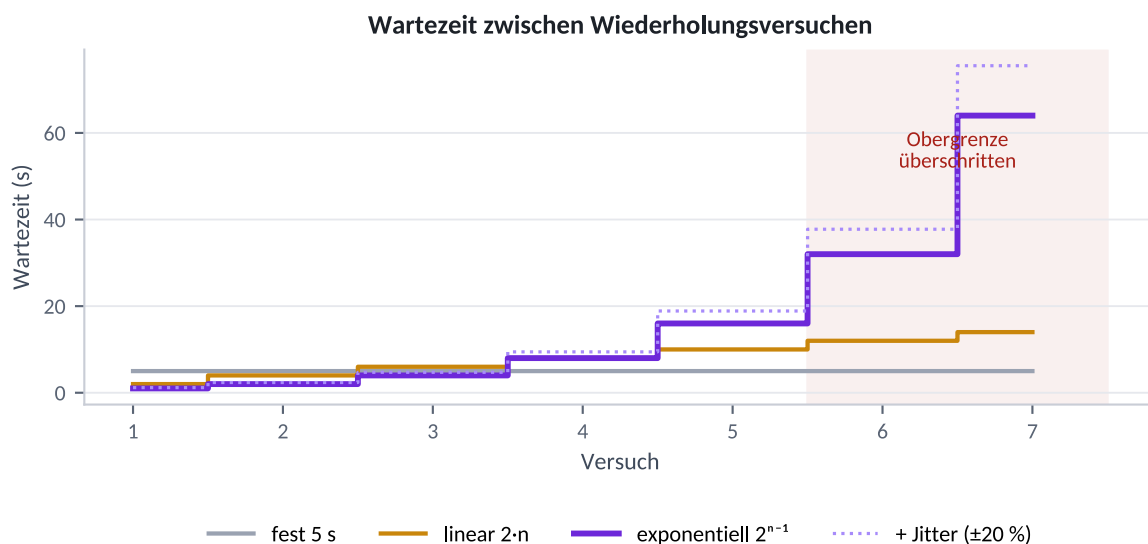


Abb. 5.4 Wartestrategien im Vergleich. Der Zufallsanteil verhindert, dass viele gleichzeitig blockierte Abläufe im selben Moment erneut anfragen — ohne ihn entsteht nach jeder Störung eine zweite Störung.

Und die Regel dahinter: **Wiederholung setzt Idempotenz voraus.** Wer Wiederholung auf einem nicht idempotenten Ablauf einschaltet, hat die Zahl der Doppelbuchungen mit fünf multipliziert. Modul 2 ist die Voraussetzung für dieses Modul.

SITUATION	WIEDERHOLEN?	WIE OFT	ABSTAND
Netzwerkfehler beim Lesen	ja	5	1-16 s, exponentiell
429 mit Retry-After	ja	5	die genannte Zeit, nicht kürzer
5xx beim Schreiben, Ablauf idempotent	ja	3	2-8 s, exponentiell
5xx beim Schreiben, Ablauf nicht idempotent	nein	—	in die Fehlerablage, Zustand prüfen lassen
401 / 403	nein	—	Zugangsdaten erneuern, dann neu starten
400 / 422	nein	—	Fehlerablage bzw. Prüfliste

DIE ZEILE, ÜBER DIE MAN ZWEIMAL NACHDENKEN SOLLTE

Zeile 4 ist der unbequeme Fall: Ein 5xx beim Schreiben bedeutet nicht, dass nichts passiert ist — es bedeutet, dass du keine verlässliche Antwort bekommen hast. Der Datensatz kann angelegt sein. Ohne Idempotenz ist die einzig richtige Reaktion: nicht wiederholen, sondern nachsehen. Mit Idempotenz ist die Wiederholung unbedenklich — das ist der praktische Wert des ganzen Aufwands aus Modul 2.

5.3 Fehlerablage und Alarmierung

Jeder Fehler, der nicht durch Wiederholung verschwindet, gehört in eine **Fehlerablage** — eine Tabelle oder ein Datenspeicher mit sechs Spalten: Zeitpunkt, Szenario, Auslöserdaten, Schritt, Fehlermeldung, Status. Diese Ablage ist das, was aus einem Ausfall einen bearbeitbaren Vorgang macht.

SPALTE	INHALT	WARUM SIE GEBRAUCHT WIRD
Zeitpunkt	ISO 8601 mit Zeitzone	Zuordnung zu Störungen beim Anbieter
Szenario	Name und Version	Bei mehreren Szenarien je Prozess
Auslöserdaten	Kennung des Vorgangs, nicht der ganze Datensatz	Wiederanlauf ohne erneutes Suchen; Datensparsamkeit
Schritt	Fachlicher Name des gescheiterten Schritts	Grundlage der Wiederaufnahme (2.3)
Fehlermeldung	Statuscode und Text, gekürzt	Klassenzuordnung nach 5.1
Status	offen / in Arbeit / erledigt / nicht behebbar	Sonst wird die Ablage zur Halde

Dazu eine **Alarmierung**, die tatsächlich jemanden erreicht. Nicht eine Mail an ein Postfach, das niemand liest, sondern an eine Person oder einen Kanal mit klarer Zuständigkeit. Und mit Inhalt: Welches Szenario, welcher Datensatz, welcher Schritt, was ist zu tun.

ALARM, DER IGNORIERT WIRD

Betreff: Fehler in Szenario 14

Ein Fehler ist aufgetreten.
Details im Ausführungsprotokoll.

ALARM, DER BEARBEITET WIRD

Betreff: [Preispflege] 3 Artikel nicht aktualisiert – Prüfung nötig

Szenario: Lieferantenpreise täglich
Zeitpunkt: 11.09.2026 06:04 +02:00
Betroffen: Art. 11-482, 11-509, 12-003
Schritt: Preis schreiben
Grund: Abweichung > 50 % zum Vorpreis
Zu tun: Preise in der Prüfliste bestätigen oder ablehnen (Link)

Alarmmüdigkeit vermeiden

Eine Alarmierung, die täglich grundlos auslöst, wird nach zwei Wochen gefiltert — und dann fehlt sie in dem Moment, in dem sie zählt. Drei Maßnahmen:

- **Bündeln:** Ein Alarm je Lauf mit allen betroffenen Fällen, nicht einer je Fall.
- **Schwellen:** Einzelne fachliche Fälle in die Prüfliste, Alarm erst ab einem Anteil (etwa 5 Prozent eines Laufs) oder bei technischen Fehlern sofort.
- **Stilllegen mit Ablaufdatum:** Wer einen Alarm abschaltet, setzt ein Datum, an dem er wieder angeht. Dauerhaft stillgelegte Alarmer sind unsichtbare Ausfälle.

5.4 Wann ein Ablauf einen Menschen braucht

Fünf Kriterien. Trifft eines zu, gehört ein Freigabeschritt in den Ablauf:

1. **Unumkehrbare Wirkung nach außen** — Versand an Kunden, Veröffentlichung, Löschung.
2. **Geld** — Zahlungen, Gutschriften, Bestellungen, Preisänderungen.
3. **Entscheidungen über Personen** — Auswahl, Bewertung, Zugang, Kündigung.
4. **Niedrige Konfidenz** — insbesondere hinter einem KI-Schritt, wenn das Ergebnis unsicher ist.
5. **Ausnahme vom Muster** — alles, was in die fachliche Fehlerklasse fällt.

Wie ein guter Freigabeschritt aussieht

Ein Freigabeschritt, der nur „Bestätigen?“ fragt, wird nach zwei Wochen im Reflex geklickt — und ist dann wertlos. Drei Anforderungen:

- **Entscheidungsgrundlage mitliefern:** die drei bis fünf Angaben, die für die Entscheidung zählen, direkt in der Anfrage. Nicht ein Link, der zum Anmelden zwingt.
- **Ablehnen muss genauso leicht sein wie zustimmen** — und einen sinnvollen Folgeweg haben.
- **Zeitüberschreitung ist ein definierter Fall:** Was passiert nach 24 Stunden ohne Antwort? Erinnerung, Eskalation oder Abbruch — aber nicht stillschweigend zustimmen.

SCHWACHE FREIGABE

Neue Rechnung zur Freigabe: RE-2026-4412.

Freigeben
Ablehnen

Der Freigebende weiß nichts.

Nach zwei Wochen wird im Reflex geklickt —
die Freigabe ist dann Dokumentation, keine
Kontrolle.

TRAGFÄHIGE FREIGABE

Rechnung RE-2026-4412 · Musterlieferant GmbH
 Betrag: 4.812,00 € (Vormonat 1.240,00 € · +288 %)
 Bestellbezug: keiner gefunden
 Fälligkeit: 14.09.2026
 Prüfung: Betrag = Positionen ✓ Lieferant bekannt ✓
 Bestellbezug ✗

Freigeben
Ablehnen: kein Bestellbezug
An Einkauf

Drei Angaben, zehn Sekunden, benannter Ablehnungsgrund —
und ein Folgeweg, der nicht in einer Sackgasse endet.

Abb. 5.3 Ein Freigabeschritt ohne Entscheidungsgrundlage ist ein Klick, kein Kontrollpunkt. Der Unterschied kostet zwanzig Minuten beim Bauen.

Wie viele Freigaben verträgt ein Prozess?

Freigaben sind teuer: Sie kosten Aufmerksamkeit, und Aufmerksamkeit ist die knappste Ressource im Betrieb. Deshalb gilt die Regel, Freigaben nicht auf Vorgänge, sondern auf *Ausreißer* zu legen — auf die Fälle, die eine Plausibilitätsprüfung als ungewöhnlich markiert hat.

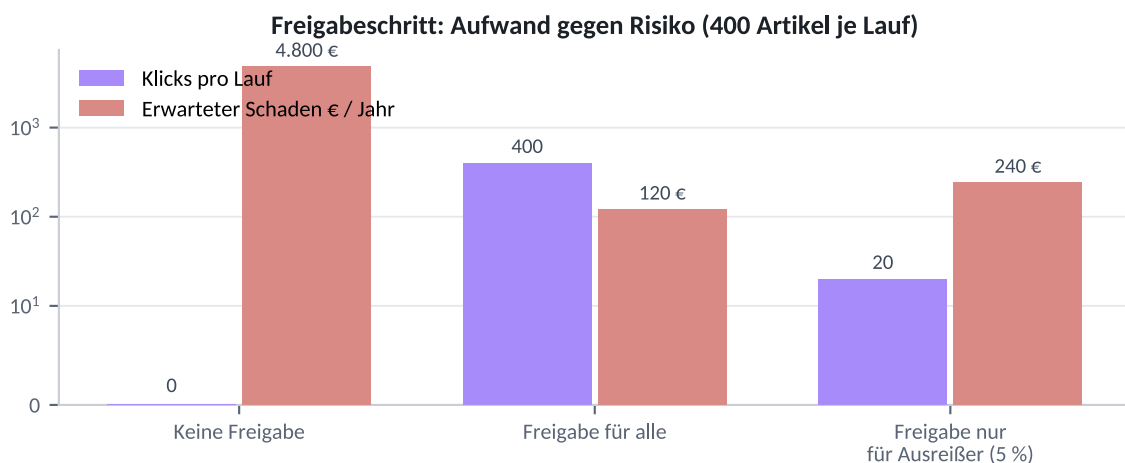


Abb. 5.5 Drei Entwürfe für dasselbe Szenario mit 400 Artikeln je Lauf. Die Freigabe für alle senkt das Risiko kaum weiter als die Freigabe für Ausreißer — kostet aber das Zwanzigfache an Klicks. Nach zwei Wochen klickt niemand mehr hin.

BEISPIEL 5.1 — FREIGABE, DIE TRÄGT

Szenario: Eingehende Lieferantenrechnungen werden ausgelesen, geprüft und zur Zahlung vorbereitet.

Schwache Freigabe: „Neue Rechnung zur Freigabe: RE-2026-4412. [Freigeben] [Ablehnen]" — Der Freigebende weiß nichts und klickt auf Freigeben.

Tragfähige Freigabe:

Rechnung RE-2026-4412 · Musterlieferant GmbH
Betrag: 4.812,00 € (Vormonat: 1.240,00 € – Abweichung +288 %)
Bestellbezug: keiner gefunden
Fälligkeit: 14.09.2026
Prüfung: Betrag stimmt mit Positionen überein ✓
Lieferant bekannt ✓ Bestellbezug x

[Freigeben] [Ablehnen: kein Bestellbezug] [An Einkauf]

Der Unterschied: Die Abweichung zum Vormonat und der fehlende Bestellbezug sind genau die zwei Angaben, die eine Entscheidung in zehn Sekunden erlauben. Und „Ablehnen" hat einen benannten Grund, der direkt in die Fehlerablage geht — statt einer Sackgasse.

5.5 Der stille Ausfall und der Wächter

Die schlimmste Fehlerart ist die, bei der nichts passiert. Ein Szenario ist deaktiviert, weil die Verbindung abgelaufen ist. Ein Auslöser feuert nicht mehr, weil im Quellsystem ein Feld umbenannt wurde. Es gibt keine Fehlermeldung — es gibt nur keine Ausführungen. Drei Wochen später fragt jemand, warum keine Leads mehr ankommen.

DER STILLE AUSFALL UND SEIN GEGENMITTEL



Abb. 5.2 Die teuerste Fehlerklasse im Betrieb ist die, bei der nichts passiert. Ein Wächter ist das einzige Mittel, das sie sichtbar macht.

Gegenmaßnahme: ein Wächter. Ein zweites, einfaches Szenario, das täglich prüft, ob das erste seit einer erwarteten Zeitspanne gelaufen ist — und Alarm schlägt, wenn nicht. Zehn Minuten Aufwand für die Fehlerklasse, die im Betrieb am teuersten ist.

WAS DER WÄCHTER PRÜFT	WORAN ER ES ERKENNT	SCHWELLE
Läuft das Szenario überhaupt?	Letzter erfolgreicher Lauf in der Protokoll- oder Ergebnistabelle	abhängig vom Takt, meist 24 h
Läuft es in normaler Menge?	Zahl der verarbeiteten Vorgänge gegen den Wochentagsdurchschnitt	Abweichung > 50 % nach unten
Wächst die Fehlerablage?	Zahl offener Einträge	mehr als 10 offen oder älter als 3 Tage
Laufen Zugangsdaten bald ab?	Ablaufdatum der Verbindung	14 Tage vorher

WER WACHT ÜBER DEN WÄCHTER?

Ein Wächter, der selbst ausfällt, ist schlimmer als keiner — er erzeugt das Gefühl von Sicherheit. Zwei einfache Gegenmittel: Erstens den Wächter so bauen, dass er auch bei Erfolg eine Spur hinterlässt (eine Zeile „geprüft, alles in Ordnung“), die selbst alt werden kann. Zweitens eine wöchentliche Nachricht „Wächter aktiv, n Szenarien geprüft“ — eine Nachricht, deren Ausbleiben auffällt.

ÜBUNG 5 — FEHLERBEHANDLUNGSPLAN

Szenario: Täglich um 6 Uhr werden aus einer Lieferanten-API Preisänderungen gelesen und im Shop aktualisiert. Betroffen sind je Lauf 50 bis 400 Artikel.

1. Nenne für jede der drei Fehlerklassen zwei konkrete Fehler, die hier auftreten können.
2. Lege die Wiederholungsstrategie fest: bei welchen Fehlern, wie oft, mit welchen Abständen.
3. Entwirf die Fehlerablage: welche sechs Spalten, und wer sieht sie wann?
4. Ein Artikel bekommt einen Preis von 0,01 € statt 129 €. Welche Fehlerklasse ist das, und welche Prüfung fängt es ab?
5. Braucht dieses Szenario einen Freigabeschritt? Begründe mit den fünf Kriterien aus 5.4 — und wenn ja: an welcher Stelle, mit welcher Entscheidungsgrundlage?
6. Beschreibe den Wächter, der einen stillen Ausfall bemerkt.
7. **Zusatz:** Entwirf den Alarmtext nach dem Muster aus 5.3 — vollständig, so dass die Person, die ihn erhält, ohne Rückfrage handeln kann.

KERNAUSSAGEN

- Drei Fehlerklassen mit drei verschiedenen Reaktionen; die fachliche wird am häufigsten übersehen und geht an den Fachbereich, nicht an die IT.
- Wiederholung nur bei vorübergehenden Fehlern, mit wachsendem Abstand und Obergrenze.
- Wiederholung ohne Idempotenz vervielfacht den Schaden.
- Fehlerablage mit sechs Spalten und Status; Alarmierung mit Inhalt und Zuständigkeit.
- Alarmmüdigkeit ist ein Entwurfsfehler: bündeln, Schwellen setzen, Stilllegung befristen.
- Fünf Kriterien für den Menschen im Ablauf; Freigaben auf Ausreißer legen, nicht auf alle Vorgänge.
- Ein Freigabeschritt braucht Entscheidungsgrundlage, gleichwertiges Ablehnen und eine Regel für Zeitüberschreitung.
- Ein Wächter gegen den stillen Ausfall — und eine Spur, deren Ausbleiben auffällt.

Selbsttest Modul 5

1. Was ist die Voraussetzung dafür, Wiederholung überhaupt einzuschalten?

- a) Ein schnelles Netzwerk.
- b) Idempotenz des Ablaufs.
- c) Ein kostenpflichtiger Tarif.
- d) Ein Freigabeschritt.

2. Welche Fehlerart bleibt am längsten unbemerkt?

- a) Ein 500 beim Zielsystem.
- b) Ein abgelaufenes Token.
- c) Der stille Ausfall — das Szenario läuft gar nicht mehr.
- d) Ein fehlendes Pflichtfeld.

MODUL 6

KI-Agenten und Wirtschaftlichkeit

Ein KI-Schritt ist der einzige Schritt in deinem Szenario, der bei gleicher Eingabe nicht dasselbe tut. Und eine Wirtschaftlichkeitsrechnung ohne Wartungsposten ist der einzige Grund, aus dem gute Automatisierungen nach einem Jahr abgeschaltet werden. Beides behandelt dieses Modul.

ABSCHNITT	INHALT	ZEIT
6.1	Der KI-Schritt und die Pflicht zur Validierung	30 Min
6.2	Automatisierung oder Agent — und welche Werkzeuge er bekommt	25 Min
6.3	Wirtschaftlichkeit vollständig rechnen	35 Min
6.4	Kipppunkt und Sensitivität	20 Min
6.5	Wenn sich eine Automatisierung nicht lohnt	15 Min
Übung 6	Wirtschaftlichkeit rechnen, Agentengrenzen ziehen	50 Min
Selbsttest	Zwei Fragen im Prüfungsformat	10 Min

LERNZIELE

- Du baust einen KI-Schritt so ein, dass er das Szenario nicht unberechenbar macht.
- Du unterscheidest eine Automatisierung von einem Agenten und kennst die Folgen.
- Du entscheidest, welche Werkzeuge ein Agent bekommt — und welche nicht.
- Du rechnest die Wirtschaftlichkeit vollständig, einschließlich Wartung und Fehlerkosten.
- Du bestimmst den Kipppunkt und erkennst die Fälle, in denen sich eine Automatisierung nicht lohnt.

6.1 Ein KI-Schritt ist ein unzuverlässiger Schritt

Jeder andere Schritt in deinem Szenario ist berechenbar: Dieselbe Eingabe erzeugt dieselbe Ausgabe. Ein KI-Schritt nicht. Er liefert bei gleicher Eingabe unterschiedliche Formulierungen, gelegentlich ein anderes Format, und im Fehlerfall etwas, das nach einer Antwort aussieht.

Daraus folgt eine feste Bauweise. Ein KI-Schritt steht nie allein:

1 KI-Schritt

- Ausgabeformat erzwingen (JSON, feste Felder)
- Erlaubte Werte als Aufzählung vorgeben
- Feld "confidence" mitliefern lassen
- Regel für "weiß ich nicht" im Prompt

2 Validierung — kein KI-Schritt

- Ist es gültiges JSON?
- Sind alle Pflichtfelder da?
- Liegen die Werte in der erlaubten Aufzählung?
- Plausibilität: Betrag > 0, Datum nicht in der Zukunft

3 Verzweigung

- gültig und sicher → weiter im Ablauf
- gültig und unsicher → Freigabeschritt (Modul 5)
- ungültig → Fehlerablage, Mensch informieren

Block 2 ist der, den man weglässt — weil „das Modell das doch richtig macht“. Er ist der Unterschied zwischen betreiben und überwachen.

Abb. 6.1 Ein KI-Schritt steht nie allein. Die Validierung darf selbst kein KI-Schritt sein — sonst prüft die Unsicherheit sich selbst.

Der zweite Block ist der, den man weglässt, weil „das Modell das doch richtig macht“. Er ist der Unterschied zwischen einem Szenario, das man betreiben kann, und einem, das man überwachen muss.

Was in den KI-Schritt hineingehört

ELEMENT	WARUM	BEISPIEL
Festes Ausgabeformat	Ohne Format ist jede Antwort ein Sonderfall	JSON mit den Feldern <code>kat</code> egorie, <code>betrag</code> , <code>confidence</code>
Aufzählung erlaubter Werte	Verhindert erfundene Kategorien	<code>"kat</code> egorie": "Miete Reise Material Sonstiges"
Konfidenzfeld	Macht Unsicherheit zur auswertbaren Größe	<code>"confidence": 0.0–1.0</code>
Regel für Unwissen	Sonst wird geraten	„Wenn nicht eindeutig, setze <code>kat</code> egorie = "unklar" "
Keine Anweisungen aus Fremddaten	Freitext aus einer externen Quelle kann Anweisungen enthalten	Eingabe klar als Daten kennzeichnen und abgrenzen

DIE VALIDIERUNG DARF KEIN ZWEITER KI-SCHRITT SEIN

Wenn die Prüfung selbst ein Modellaufruf ist, prüft die Unsicherheit sich selbst. Die Validierung ist regelbasiert: Ist es gültiges JSON? Sind alle Pflichtfelder da? Liegt der Wert in der Aufzählung? Ist der Betrag größer null und das Datum nicht in der Zukunft? Diese vier Prüfungen kosten vier Bedingungen und fangen den größten Teil aller Fehlerfälle ab.

Konfidenz ist eine Selbstauskunft, kein Messwert

Ein Modell, das `0.95` ausgibt, hat nicht gemessen, dass es zu 95 Prozent richtig liegt — es hat eine Zahl erzeugt. Der Wert ist trotzdem nützlich, weil er innerhalb desselben Aufbaus *relativ* verlässlich ist: Fälle mit niedriger Selbsteinschätzung sind tatsächlich häufiger falsch. Nutze ihn als Sortierkriterium für den Freigabeschritt, nicht als Wahrscheinlichkeit.

6.2 Automatisierung oder Agent

	AUTOMATISIERUNG	AGENT
Ablauf	Du legst die Schritte fest	Das Modell wählt die Schritte
Vorhersagbarkeit	Vollständig	Begrenzt
Fehlersuche	Schritt für Schritt nachvollziehbar	Nur über Protokolle rekonstruierbar
Geeignet für	Bekannte, wiederkehrende Abläufe	Aufgaben mit vielen Varianten
Kosten je Vorgang	konstant und im Voraus bekannt	schwankend — abhängig von der Zahl der Modellaufrufe
Braucht	Fehlerbehandlung	Fehlerbehandlung <i>und</i> harte Grenzen

In No-Code-Plattformen gibt es zunehmend Agentenbausteine: ein Modell, dem mehrere Werkzeuge zur Verfügung stehen und das selbst entscheidet, welches es benutzt. Das ist mächtig und verlagert die Frage von „funktioniert der Ablauf?“ zu „was kann dieser Agent im schlimmsten Fall anrichten?“.

WELCHE WERKZEUGE BEKOMMT EIN AGENT?



Abb. 6.2 Werkzeugvergabe an einen Agenten als Zwei-Achsen-Entscheidung. Die Frage lautet nicht „funktioniert der Ablauf?“, sondern „was kann er im schlimmsten Fall anrichten?“.

Die Regel dafür ist dieselbe wie in Modul 5, nur strenger: Ein Agent bekommt nur Werkzeuge, deren Missbrauch du hinnehmen könntest. Leserechte großzügig, Schreibrechte sparsam, alles mit Außenwirkung hinter einem Freigabeschritt. Wer einem Agenten Versand- und Löschrechte gibt und auf gute Anweisungen vertraut, hat die Grenze im falschen Medium gezogen.

Drei Grenzen, die jeder Agent braucht

- **Schrittobergrenze:** Nach n Modellaufrufen bricht der Agent ab und übergibt an einen Menschen. Ohne sie kostet ein unleserlicher Beleg mehr als der Beleg wert ist.
- **Kostenobergrenze je Vorgang:** Ein harter Betrag, bei dessen Überschreitung abgebrochen wird.
- **Werkzeugliste statt Werkzeugzugang:** Der Agent bekommt genau die benannten Werkzeuge — nicht „Zugriff auf das CRM“, sondern „Kunde suchen“ und „Notiz anlegen“.

6.3 Wirtschaftlichkeit rechnen

Die übliche Rechnung lautet: *Zeitersparnis mal Stundensatz, minus Plattformkosten*. Sie ist zu optimistisch, weil sie den größten Posten auslässt.

Nutzen pro Jahr
= Vorgänge/Jahr × eingesparte Minuten/Vorgang × Minutensatz

Kosten pro Jahr
= Plattformkosten
+ KI-Kosten (Aufrufe × Preis je Aufruf)
+ Aufbau (einmalig, auf 3 Jahre verteilt)
+ WARTUNG (das vergessene Drittel)
+ Fehlerkosten (Häufigkeit × Schaden × Entdeckungsquote)

Zur Wartung: Eine belastbare Faustregel aus der Praxis lautet, dass der Aufbau etwa ein Drittel und der laufende Betrieb zwei Drittel des Gesamtaufwands ausmachen — über die Lebensdauer gerechnet. Quellsysteme ändern ihre APIs, Verbindungen laufen ab, Sonderfälle kommen dazu, jemand muss die Fehlerablage abarbeiten. Wer nur den Aufbau rechnet, rechnet ein Drittel.

Abb. 6.3 Links das Verhältnis von Aufbau zu Betrieb über die Lebensdauer, rechts die Kostenposten des Rechenbeispiels 6.1. Die Wartung ist der größte Einzelposten und fehlt in fast jeder ersten Präsentation.

Zur Entdeckungsquote: Der Faktor, den fast niemand ansetzt. Ein Fehler, der zu 100 Prozent auffällt, kostet die Korrektur. Ein Fehler, der zu 20 Prozent auffällt, kostet die Korrektur *plus* den Schaden der 80 Prozent, die durchgehen. Deshalb ist ein Szenario mit guter Fehlerablage nicht nur ordentlicher, sondern messbar billiger.

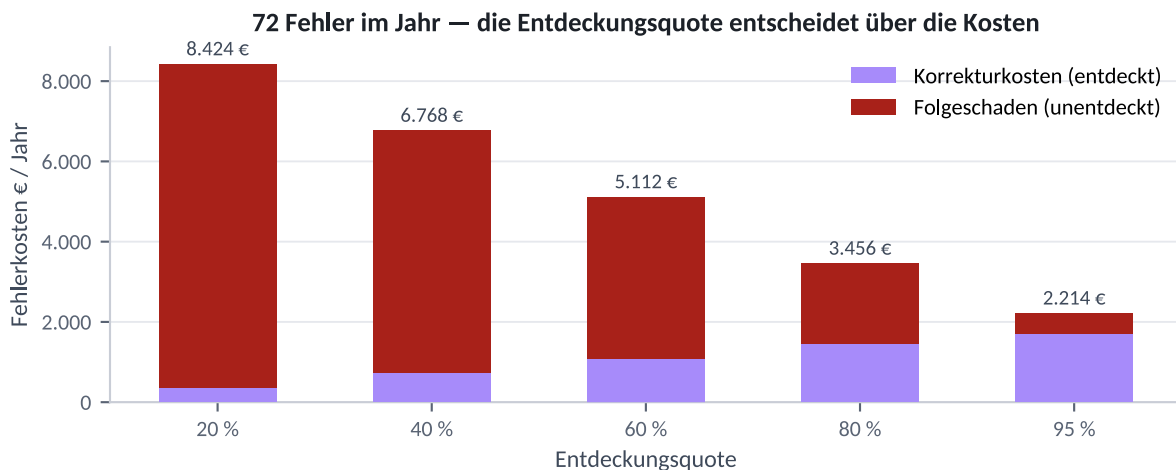


Abb. 6.4 Dieselbe Fehlerzahl, fünf Entdeckungsquoten. Von 60 auf 95 Prozent zu kommen senkt die Fehlerkosten um mehr als ein Drittel — allein durch Fehlerablage und Plausibilitätsprüfung.

BEISPIEL 6.1 — EINE RECHNUNG, DIE TRÄGT

Vorhaben: Eingehende Bestellungen aus dem Shop automatisch ins ERP übertragen.
Annahmen: 1.800 Bestellungen im Jahr, 6 Minuten Handarbeit je Bestellung, Minutensatz 0,60 €, Aufbau 22 Stunden à 65 €.

POSITION	JAHR
1.800 Bestellungen × 6 Minuten × 0,60 €/Min	+ 6.480 €
Plattform (mittlerer Tarif)	– 420 €
KI-Schritt zur Adressprüfung: 1.800 × 0,004 €	– 7 €
Aufbau 22 Std. × 65 € = 1.430 €, auf 3 Jahre	– 477 €
Wartung: 2 Std./Monat × 65 €	– 1.560 €
Fehlerkosten: 1 % falsch × 1.800 × 25 € Korrektur ÷ 0,8 Entdeckung	– 563 €
Ergebnis	+ 3.453 €

Was die Rechnung zeigt: Der Nutzen ist real, aber die Wartung frisst ein Viertel davon — und sie ist der Posten, der in der ersten Präsentation nie auftaucht. Die Amortisation liegt bei etwa fünf Monaten.

Und was sie noch zeigt: Bei 300 statt 1.800 Bestellungen wäre der Nutzen 1.080 € und die Rechnung deutlich negativ. Dieselbe Automatisierung, dieselbe Qualität — und trotzdem die falsche Entscheidung. Die Frage lautet nie „ist das automatisierbar?“, sondern immer „bei welcher Menge lohnt es sich?“.

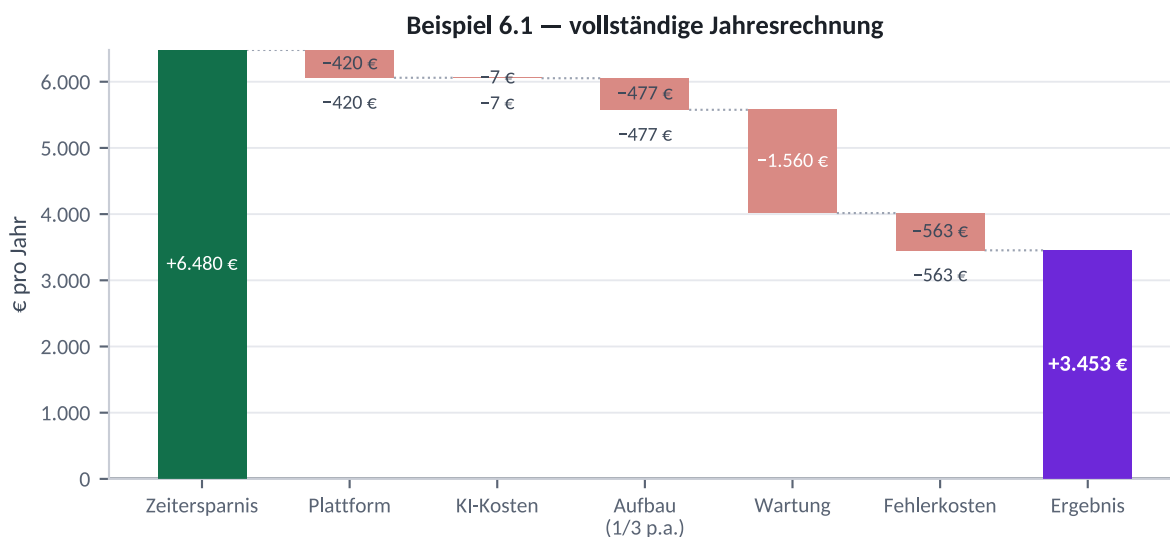


Abb. 6.5 Die Rechnung als Wasserfall: Der Nutzen links, die fünf Kostenposten in der Mitte, das Ergebnis rechts. Wer nur die ersten beiden Balken zeigt, verspricht 6.060 € statt 3.453 €.

6.4 Kipppunkt und Sensitivität

Eine einzelne Zahl am Ende der Rechnung ist wenig wert, weil jede Annahme geschätzt ist. Zwei Auswertungen machen daraus eine belastbare Entscheidung: der Kipppunkt und die Sensitivität.

Der Kippunkt

Bei welcher Menge wird das Ergebnis null? Diese Zahl ist wichtiger als das Ergebnis selbst, weil sie sagt, wie viel Luft die Entscheidung hat. Liegt die erwartete Menge knapp über dem Kippunkt, ist das Vorhaben eine Wette auf die Mengenprognose.

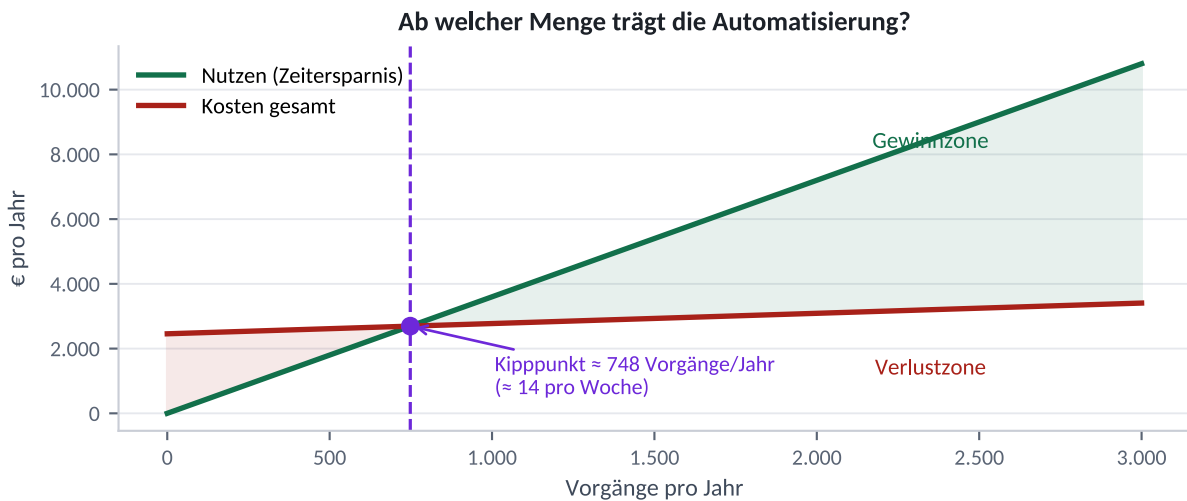


Abb. 6.6 Nutzen und Kosten über der Menge. Der Kippunkt ist die Zahl, die in die Entscheidungsvorlage gehört — zusammen mit der Frage, wie sicher die erwartete Menge ist.

Die Sensitivität

Welche Annahme kippt das Ergebnis, wenn sie um ein Viertel danebenliegt? Rechne jede Annahme einmal mit –25 Prozent durch und notiere die Wirkung auf das Ergebnis. Das dauert zehn Minuten und beantwortet die einzige Frage, die in der Entscheidungsrunde wirklich gestellt wird.

ANNAHME	BASIS	BEI –25 %	ERGEBNIS	WIRKUNG
Menge (Vorgänge/Jahr)	1.800	1.350	+ 1.833 €	stark
Eingesparte Minuten je Vorgang	6	4,5	+ 1.833 €	stark
Minutensatz	0,60 €	0,45 €	+ 1.833 €	stark
Aufbauaufwand	1.430 €	1.073 €	+ 3.572 €	gering
Wartungsaufwand	1.560 €	1.170 €	+ 3.843 €	mittel
Entdeckungsquote (bei –25 %)	80 %	60 %	+ 3.266 €	gering

WIE DAS ERGEBNIS ZU LESEN IST

Die ersten drei Zeilen sind dieselbe Zahl, weil sie im Nutzen multipliziert werden — ein Viertel weniger Menge wirkt genauso wie ein Viertel weniger Zeitersparnis. Das ist die eigentliche Erkenntnis der Tabelle: Die Nutzenseite entscheidet, die Kostenseite nuanciert. Wer eine Wirtschaftlichkeitsrechnung verteidigen muss, verteidigt die Mengenschätzung und die gemessene Bearbeitungszeit — nicht den Plattformtarif.

Und die letzte Zeile zeigt, warum eine gute Fehlerbehandlung sich rechnet, ohne dass sie das Ergebnis dominiert: Sie schützt gegen den Fall, der in dieser Tabelle gar nicht steht — den seltenen, teuren, unentdeckten Fehler.

6.5 Wenn sich eine Automatisierung nicht lohnt

Es gehört zum Beruf, ein Vorhaben abzulehnen. Fünf Muster, bei denen die Antwort „nicht automatisieren“ die richtige ist — und was stattdessen hilft:

MUSTER	WARUM ES NICHT TRÄGT	WAS STATTDESSEN
Zu selten	Zweimal im Jahr amortisiert keinen Aufbau — und in einem Jahr weiß niemand mehr, wie das Szenario funktioniert	Checkliste und eine halbe Stunde Handarbeit
Regel instabil	Monatlicher Umbau kostet mehr als der Prozess	Erst die Regel klären lassen, dann in sechs Monaten neu fragen
Ausnahmen überwiegen	Der Normalfall ist der Sonderfall	Prozess bereinigen; oft ist das der eigentliche Gewinn
Quelldaten unbrauchbar	Keine stabile Kennung, keine Struktur, Freitextfelder	Erfassung ändern — ein Formularfeld ist billiger als jede Zerlegungslogik
Verantwortung ungeklärt	Niemand bearbeitet die Fehlerablage, niemand entscheidet über Ausnahmen	Betrieb klären (Anhang D), dann bauen

DER SATZ, MIT DEM MAN EIN VORHABEN ABLEHNT

„Das lässt sich automatisieren, aber es lohnt sich erst ab etwa 35 Vorgängen pro Woche — aktuell sind es zwölf. Wenn die Menge steigt, ist der Aufbau in zwei Tagen gemacht.“ Diese Formulierung nennt die Zahl, die Bedingung und den Weg zurück. Sie ist besser als jedes Nein, weil sie eine Entscheidung ermöglicht statt sie zu verhindern.

ÜBUNG 6 – WIRTSCHAFTLICHKEIT UND GRENZEN

Teil A: Ein Unternehmen will die Erfassung von Reisekostenbelegen automatisieren. 40 Belege pro Woche, je 4 Minuten Handarbeit, Minutensatz 0,55 €. Aufbau geschätzt 30 Stunden à 70 €. Plattform 30 €/Monat. KI-Schritt zur Belegauslesung 0,01 € je Beleg. Fehlerquote geschätzt 4 %, Korrektur je Fehler 12 €, Entdeckungsquote 60 %.

1. Rechne den Jahreswert nach dem Muster aus 6.1, mit Wartung nach der Faustregel.
2. Lohnt es sich? Ab welcher Belegzahl pro Woche kippt die Rechnung?
3. Welche einzelne Maßnahme verbessert das Ergebnis am stärksten — und warum ist es nicht die Fehlerquote?
4. Führe die Sensitivitätsrechnung aus 6.4 für die drei Nutzenannahmen durch.

Teil B: Der Fachbereich schlägt vor, den KI-Schritt zu einem Agenten auszubauen, der bei unklaren Belegen selbst im Buchhaltungssystem nachschlägt, Rückfragen per Mail stellt und die Buchung anlegt.

1. Welche Werkzeuge bekäme dieser Agent, und welche davon würdest du ihm verweigern? Ordne sie in die Matrix aus 6.2 ein.
2. An welcher Stelle setzt du den Freigabeschritt, und welche Angaben enthält er?
3. Nenne zwei Fälle, in denen der Agent etwas tut, das ein festes Szenario nicht getan hätte — einen guten und einen schlechten.
4. **Zusatz:** Welche zwei harten Grenzen aus 6.2 legst du fest, und mit welchen Zahlen?

KERNAUSSAGEN

- Ein KI-Schritt steht nie allein: Format erzwingen, validieren, verzweigen.
- Die Validierung ist kein KI-Schritt — sonst prüft die Unsicherheit sich selbst.
- Konfidenz ist eine Selbstauskunft, nützlich als Sortierkriterium, nicht als Wahrscheinlichkeit.
- Ein Agent bekommt nur Werkzeuge, deren Missbrauch hinnehmbar wäre — plus Schritt- und Kostenobergrenze.
- Wartung ist rund zwei Drittel des Gesamtaufwands und fehlt in jeder ersten Rechnung.
- Fehlerkosten mal Entdeckungsquote — unentdeckte Fehler sind die teuren.
- Kippunkt und Sensitivität machen aus einer Zahl eine Entscheidung.
- Die Frage ist nicht „automatisierbar?“, sondern „ab welcher Menge lohnt es sich?“.

Selbsttest Modul 6

1. Warum darf die Validierung eines KI-Schritts kein weiterer KI-Schritt sein?
 - a) Weil es doppelte Kosten verursacht.
 - b) Weil dann Unsicherheit die Unsicherheit prüft.
 - c) Weil Plattformen das nicht erlauben.
 - d) Weil es langsamer ist.

2. Welcher Kostenposten fehlt in der Regel bei der ersten Wirtschaftlichkeitsrechnung?

- a) Die Plattformkosten.
- b) Die Aufbaukosten.
- c) Die laufende Wartung.
- d) Die KI-Kosten.

Musterprüfung A

20 Fragen · 40 Minuten · Bestehensgrenze 70 % (14 von 20)

Zwei vollständige Prüfungssätze im Format der Zertifizierungsprüfung. Musterprüfung A deckt die Module in ihrer Reihenfolge ab; Musterprüfung B mischt die Themen und enthält mehr Szenariofragen — sie ist die schwierigere von beiden und der bessere Gradmesser.

	MUSTERPRÜFUNG A	MUSTERPRÜFUNG B
Fragen	20	20
Dauer	40 Minuten	40 Minuten
Bestehensgrenze	70 % (14 von 20)	70 % (14 von 20)
Charakter	themenweise, nah am Lehrtext	gemischt, mehr Anwendung
Empfohlener Zeitpunkt	nach Modul 6	zwei Tage später

Unter realen Bedingungen bearbeiten: ohne Hilfsmittel, ohne Unterbrechung, Zeit nehmen. Genau eine Antwort ist richtig. Lösungen mit Begründung in Anhang A.

1. Welcher Auslösertyp verbraucht Aufrufe auch dann, wenn nichts passiert?

- a) Webhook
- b) Abfrage (Polling)
- c) Manueller Start
- d) Keiner

2. Ein Prozess braucht bei jedem dritten Vorgang eine Sonderbehandlung. Was folgt?

- a) Sofort automatisieren, die Zeitersparnis ist groß.
- b) Zuerst den Prozess bereinigen, dann automatisieren.
- c) Nur den Sonderfall automatisieren.
- d) Der Ausnahmeanteil ist ohne Bedeutung.

3. Was bedeutet Idempotenz?

- a) Der Ablauf läuft besonders schnell.
- b) Zweimal ausgeführt hat er dasselbe Ergebnis wie einmal.
- c) Er kann nicht scheitern.
- d) Er braucht keine Fehlerbehandlung.

4. Woher stammt ein guter Deduplikationsschlüssel?

- a) Aus dem Zeitstempel des Laufs.
- b) Stabil aus dem Quellsystem, eindeutig im Zielsystem.
- c) Aus einer Zufallszahl.
- d) Aus der Zeilennummer der Tabelle.

5. Warum gehören nach außen wirkende Schritte ans Ende eines Ablaufs?

- a) Weil sie mehr Kontingent kosten.
- b) Weil ein Abbruch davor keinen sichtbaren Zwischenzustand hinterlässt.
- c) Weil sie langsamer sind.
- d) Weil sie sonst nicht protokolliert werden.

6. Welche HTTP-Methode ist bei Wiederholung am gefährlichsten?

- a) GET
- b) PUT
- c) POST
- d) DELETE

7. Bei welchem Code ist Wiederholen sinnvoll?

- a) 401
- b) 422
- c) 429
- d) 400

8. Ein 409 Conflict beim Anlegen eines Kontakts bedeutet meist:

- a) Ein schwerer Fehler, Ablauf abbrechen.
- b) Deine Idempotenzprüfung greift — verzweige darauf.
- c) Die Zugangsdaten sind falsch.
- d) Das Kontingent ist erschöpft.

9. Warum reicht eine geheime Webhook-Adresse nicht als Schutz?

- a) Sie ist zu kurz.
- b) Sie taucht in Protokollen und Exporten auf; wer sie kennt, kann Daten einschicken.
- c) Sie läuft nach 30 Tagen ab.
- d) Sie funktioniert nur mit OAuth.

10. Warum sollte ein Webhook-Empfang sofort bestätigt werden?

- a) Damit das Protokoll kürzer wird.
- b) Weil der Absender sonst wiederholt und das Szenario mehrfach läuft.
- c) Weil die Plattform es vorschreibt.
- d) Um Kontingent zu sparen.

11. Welches Datumsformat gehört in eine Integration?

- a) 30.08.2026 14:05
- b) 2026-08-30T14:05:00+02:00
- c) 30/08/26
- d) Unverändert aus der Quelle

12. Ein Rabattfeld fehlt in der Quelle und wird als 0 übertragen. Die Folge ist:

- a) nichts, 0 und leer sind gleich.
- b) ein bestehender Rabatt im Ziel wird überschrieben.
- c) der Datensatz wird abgelehnt.
- d) das Feld wird ignoriert.

13. Wie überträgt man Geldbeträge zuverlässig?

- a) Als formatierte Zeichenkette mit Tausendertrennzeichen.
- b) Als reine Zahl ohne Trennzeichen, oft in der kleinsten Einheit.
- c) Immer als Fließkommazahl mit vier Nachkommastellen.
- d) Im Format des Quellsystems.

14. Welche Fehlerklasse rechtfertigt einen automatischen Wiederholungsversuch?

- a) Dauerhafte Fehler.
- b) Fachliche Fehler.
- c) Vorübergehende Fehler.
- d) Alle drei.

15. Was ist die Voraussetzung dafür, Wiederholung einzuschalten?

- a) Ein Freigabeschritt.
- b) Idempotenz.
- c) Ein Webhook-Auslöser.
- d) Ein KI-Schritt.

16. Was ist ein „stiller Ausfall“?

- a) Ein Fehler ohne Meldung im Protokoll.
- b) Das Szenario läuft gar nicht mehr, ohne dass jemand es merkt.
- c) Eine Zeitüberschreitung.
- d) Ein abgelehnter Datensatz.

17. Welches Kriterium verlangt einen Menschen im Ablauf?

- a) Der Ablauf hat mehr als zehn Schritte.
- b) Der Ablauf löst eine Zahlung aus.
- c) Der Ablauf läuft nachts.
- d) Der Ablauf nutzt einen Webhook.

18. Ein guter Freigabeschritt zeichnet sich vor allem dadurch aus, dass er

- a) möglichst schnell zu bestätigen ist.
- b) die Entscheidungsgrundlage direkt mitliefert.
- c) an mehrere Personen gleichzeitig geht.
- d) nach 24 Stunden automatisch zustimmt.

19. Wie wird ein KI-Schritt im Szenario abgesichert?

- a) Durch einen zweiten KI-Schritt, der prüft.
- b) Durch erzwungenes Format, regelbasierte Validierung und Verzweigung.
- c) Durch niedrigere Temperatur.
- d) Durch mehr Beispiele im Prompt.

20. Welcher Posten fehlt am häufigsten in der Wirtschaftlichkeitsrechnung?

- a) Plattformkosten
- b) Aufbaukosten
- c) Laufende Wartung
- d) KI-Kosten

Musterprüfung B

20 Fragen · 40 Minuten · Bestehensgrenze 70 % (14 von 20)

Frühestens zwei Tage nach Musterprüfung A bearbeiten. Diese Fassung enthält mehr Fragen, die eine Entscheidung verlangen statt einer Definition — das entspricht dem Schwierigkeitsgrad der echten Prüfung.

1. Ein Webhook-Auslöser wird mit einem täglichen Abgleichslauf kombiniert. Wozu dient der Abgleichslauf?

- a) Er beschleunigt die Verarbeitung.
- b) Er holt Ereignisse nach, die der Webhook verpasst hat.
- c) Er ersetzt die Fehlerablage.
- d) Er senkt die Kontingentkosten.

2. Welche Voraussetzung muss erfüllt sein, damit ein nachholender Abgleichslauf keinen Schaden anrichtet?

- a) Er muss nachts laufen.
- b) Der Ablauf muss idempotent sein.
- c) Er darf nur lesen.
- d) Er braucht einen eigenen Zugang.

3. Ein Filter sortiert 70 % der Vorgänge aus. Wo sollte er stehen?

- a) Möglichst spät, damit alle Daten vorliegen.
- b) So früh wie möglich im Ablauf.
- c) Direkt vor dem Aggregator.
- d) Die Position ist ohne Wirkung.

4. Zwei Zweige einer Verzweigung haben überlappende Bedingungen. Was passiert?

- a) Der erste passende Zweig gewinnt.
- b) Beide Zweige laufen und schreiben ins Ziel.
- c) Das Szenario bricht ab.
- d) Die Plattform verhindert das.

5. Ein Iterator läuft im Test über 3 Positionen, in der Produktion über 400. Welches Risiko entsteht zuerst?

- a) Die Datenqualität sinkt.
- b) Die Kontingentgrenze wird erreicht.
- c) Die Signaturprüfung schlägt fehl.
- d) Der Aggregator fehlt.

6. Ein Ablauf scheitert in Schritt 4 von 7. Die Schritte 1 bis 3 waren rein lesend und haben im eigenen System gespeichert. Was ist richtig?

- a) Kompensation, weil ein Zwischenzustand besteht.
- b) Wiederaufnahme ab Schritt 4.
- c) Neustart von Schritt 1 ohne weitere Vorkehrung.
- d) Der Vorgang muss verworfen werden.

7. Wozu dient ein Präfix wie `TEST-` im Deduplikationsschlüssel einer Testauslösung?

- a) Zur besseren Lesbarkeit der Protokolle.
- b) Damit der Testlauf den späteren echten Lauf nicht blockiert.
- c) Um Kontingent zu sparen.
- d) Um die Signatur zu erfüllen.

8. Welche Angabe brauchst du, um den Vorgangsverbrauch eines Szenarios abzuschätzen?

- a) Die durchschnittliche Listenlänge.
- b) Die maximale Listenlänge.
- c) Die Zahl der Verzweigungen.
- d) Die Laufzeit je Schritt.

9. Welche HTTP-Methode ist typischerweise wiederholbar, weil sie einen Zielzustand setzt?

- a) POST
- b) PUT
- c) PATCH auf eine Liste
- d) Keine

10. Eine API antwortet mit `202 Accepted`. Was bedeutet das für dein Szenario?

- a) Der Vorgang ist abgeschlossen.
- b) Der Vorgang wurde angenommen, das Ergebnis muss später abgeholt werden.
- c) Die Anfrage war fehlerhaft.
- d) Das Kontingent ist erschöpft.

11. Welches Authentifizierungsverfahren erlaubt es, Rechte auf das Nötige zu begrenzen und einzeln zu entziehen?

- a) API-Schlüssel
- b) Bearer-Token ohne Erneuerung
- c) OAuth
- d) Gemeinsames Geheimnis

12. Ein Szenario liest 900 Datensätze bei 50 je Seite. Was passiert ohne Schleife?

- a) Das Szenario bricht mit einem Fehler ab.
- b) Es verarbeitet 50 Datensätze und meldet Erfolg.
- c) Die API liefert automatisch alles.
- d) Das Kontingent wird überschritten.

13. Was ist an einem Zeitstempel `30.08.2026 14:05` problematisch?

- a) Er ist zu ungenau.
- b) Die Zeitzone fehlt.
- c) Er ist zu lang.
- d) Er enthält keine Sekunden.

14. Ein Nur-Datum-Feld wird mit Zeitzone übertragen. Welcher Fehler droht?

- a) Der Wert wird abgelehnt.
- b) Das Datum verschiebt sich je nach Zeitzone um einen Tag.
- c) Die Sortierung kehrt sich um.
- d) Kein Fehler.

15. Beim Abgleich gilt für ein Feld, das im Zielsystem gepflegt wird (etwa eine Vertriebsnotiz):

- a) Quelle gewinnt.
- b) Ziel gewinnt.
- c) Immer in die Prüfliste.
- d) Feld leeren.

16. Ein 5xx tritt beim Schreiben auf, der Ablauf ist **nicht** idempotent. Was ist richtig?

- a) Fünfmal wiederholen mit wachsendem Abstand.
- b) Nicht wiederholen, sondern den Zustand im Zielsystem prüfen lassen.
- c) Sofort wiederholen.
- d) Den Vorgang verwerfen.

17. Welcher Eintrag gehört **nicht** in eine Fehlerablage?

- a) Zeitpunkt mit Zeitzone
- b) Kennung des Vorgangs
- c) Der vollständige Datensatz
- d) Status der Bearbeitung

18. Eine Alarmierung löst täglich grundlos aus. Welche Folge ist am wahrscheinlichsten?

- a) Die Fehlerquote sinkt.
- b) Der Alarm wird gefiltert und fehlt im Ernstfall.
- c) Das Kontingent wird überschritten.
- d) Die Plattform sperrt den Kanal.

19. Welche Grenze braucht ein Agent zwingend, damit ein Sonderfall nicht mehr kostet als der Vorgang wert ist?

- a) Eine Zeichenbegrenzung im Prompt.
- b) Eine Schritt- und Kostenobergrenze.
- c) Ein zweites Modell zur Kontrolle.
- d) Einen Webhook-Auslöser.

20. Eine Wirtschaftlichkeitsrechnung ergibt +430 € bei einem Kippunkt von 36 Vorgängen pro Woche; aktuell sind es 40. Wie ist das zu bewerten?

- a) Klar wirtschaftlich, umsetzen.
- b) Knapp — das Ergebnis hängt an der Mengenprognose.
- c) Unwirtschaftlich, ablehnen.
- d) Der Kippunkt ist ohne Bedeutung.

Lösungen

Übungen, Selbsttests und beide Musterprüfungen. Die Übungslösungen sind ausführlicher als nötig — sie enthalten die Begründung, weil die Begründung das ist, was in der Prüfung abgefragt wird.

Übungen

Übung 1 — Automatisieren oder nicht

1 Umsatzauswertung: vollständig. Monatlich ist selten, aber der Vorgang ist lang und rein lesend — keine Außenwirkung, jeder Fehler beim nächsten Blick sichtbar.

2 Verträge beenden: teilweise. Statusänderung mit Folgen für Abrechnung und Zugang; die Erkennung automatisch, die Änderung nach Freigabe.

3 Leads ins CRM: vollständig. Häufig, stabil, geringe Fehlerkosten — der Klassiker.

4 Nachbestellung: teilweise. Löst Geld aus. Vorschlag automatisch, Bestellung nach Freigabe.

5 Krankmeldungen: teilweise, und heikel. Gesundheitsbezogene Daten, hoher Ausnahmeannteil (freie Formulierung), Folgen für die Entgeltabrechnung. Erfassung und Vorschlag automatisch, Übernahme durch die Personalabteilung. Zusätzlich: Zugriffsrechte und Aufbewahrung gehören hier ausdrücklich geklärt (Anhang D).

6 Preisliste zweimal jährlich: nicht automatisieren. Zwei Ausführungen im Jahr amortisieren keinen Aufbau, und in einem Jahr weiß niemand mehr, wie das Szenario funktioniert.

Teil B — die Freigabestellen: Bei 4 vor dem Absenden der Bestellung, mit Artikel, Menge, Preis, Vergleichspreis der letzten Bestellung und aktuellem Bestand — fünf Angaben, zehn Sekunden. Bei 2 vor der Statusänderung, mit Kundenname, Vertragsende, offenen Posten und laufenden Leistungen.

Teil C: Punkt 5 ist der Prüfstein. Eine gute Antwort nennt Ausnahmen, die tatsächlich auftreten — nicht „Systemausfall“, sondern etwa: Lieferant hat den Artikel abgekündigt; Mindestbestellmenge liegt über dem Bedarf; der Artikel ist bereits bestellt, aber noch nicht geliefert. Zu jeder Ausnahme gehört eine Festlegung, nicht nur eine Nennung.

Übung 2 — Idempotenz herstellen

1. Schaden je Schritt: Schritt 3 sendet der Führungskraft eine zweite Genehmigungsmail — sie genehmigt womöglich zweimal. Schritt 4 zieht die Urlaubstage *ein zweites Mal* ab; das ist der teuerste, weil er still passiert und erst beim Jahresabschluss auffällt. Schritt 5 erzeugt einen doppelten Kalendereintrag. Schritt 6 verwirrt den Antragsteller.

2. Schlüssel: die Zeilen-ID des Antrags aus der Tabelle — stabil und eindeutig. Gespeichert wird sie im Kalendereintrag (Beschreibungsfeld oder Fremdkennung) oder, sauberer, in einer eigenen Spalte „verarbeitet_am“ in derselben Antragstabelle.

3. Wo die Prüfung sitzt: ganz am Anfang, als Schritt 0, vor dem Lesen der Personaldatenbank. Nicht später, weil jeder Schritt davor bereits Kontingent kostet — und nicht erst vor Schritt 4, weil dann die Genehmigungsmail schon draußen ist.

4. Riskante Reihenfolge: Schritt 3 sendet die Genehmigungsmail, *bevor* in Schritt 4 die Tage abgezogen werden. Scheitert Schritt 4, ist die Genehmigung erteilt und nicht verbucht. Richtig: prüfen (1, 2), abziehen (4), Kalender (5), dann die beiden Mails — alle Außenwirkung ans Ende, wie in 2.3.

5. Zustand nach Fehlschlag in Schritt 4 (in der ungeordneten Fassung): Es ist noch nichts nach außen gewirkt, nur gelesen und geprüft. Wiederaufnahme ist richtig — nichts ist zu kompensieren. Genau deshalb war die Umordnung die eigentliche Lösung dieser Übung.

6. Testauslösung: Umzuleiten sind die Genehmigungsmail, die Bestätigungsmail und der Kalendereintrag (in einen Testkalender); der Abzug in der Personaldatenbank wird übersprungen und protokolliert. Präfix `TEST-` vor der Zeilen-ID, sonst blockiert der Testlauf den echten Antrag.

Übung 3 — Aufrufe und Fehlerreaktionen

Teil A: 1 → 401, nicht wiederholen (Token erneuern, dann neu starten). 2 → 400, nicht wiederholen. 3 → 5xx, wiederholen mit wachsendem Abstand. 4 → 409, nicht wiederholen — verzweigen und die mitgelieferte Kennung verwenden. 5 → 429, wiederholen nach `Retry-After`. 6 → 422, nicht wiederholen; fachlicher Fehler in die Prüfliste.

Teil B: 900 Aufträge zu 50 je Seite sind **18 Aufrufe** zum Lesen — plus 900 Aufrufe zum Anlegen im ERP. Das Kontingent von 100 pro Stunde bezieht sich auf die Quell-API und reicht fürs Lesen bequem; der Engpass liegt beim ERP, dessen Grenze separat zu prüfen ist.

Ohne Seitenweise-Schleife verarbeitest du 50 von 900 Aufträgen — und das Szenario meldet Erfolg. Genau die Fehlerklasse aus Modul 5, die niemand bemerkt.

Bei 429 mitten im Durchlauf: Wartezeit aus `Retry-After` nehmen, wachsender Abstand, höchstens fünf Versuche, danach in die Fehlerablage *mit der Seitennummer* — damit der Wiederanlauf dort fortsetzt statt von vorn zu beginnen.

Als Kennung dient die Auftragsnummer aus der Quelle, im ERP als Fremdkennung gespeichert.

Zusatz: Bei 504 ohne verlässliche Antwort hilft nur Nachsehen: vor jedem Wiederholungsversuch per `GET` prüfen, ob die Auftragsnummer im ERP bereits existiert. Besser noch: das Anlegen als `PUT` auf die Auftragsnummer ausführen, falls die Schnittstelle das anbietet — dann ist die Wiederholung unbedenklich.

Übung 4 — Zuordnung entwerfen

Nicht eindeutige Felder: `budget` muss aus „10.000 - 25.000 EUR“ in zwei Zahlen zerlegt werden — Trennzeichen und Währung entfernen, an „-“ teilen, beide als Zahl. `consent_news`: „on“ → `true`, *fehlend* → `false`. `submitted`: das Format ist mehrdeutig ohne Zeitzone — die Zeitzone des Formularservers muss man wissen, nicht raten, und dann nach UTC umrechnen.

Namenszerlegung: „Müller-Schmidt, Anna“ ist im Format *Nachname, Vorname* — die naive Regel „vor dem ersten Leerzeichen ist der Vorname“ liefert hier „Müller-Schmidt,“ als Vornamen. Zwei Fälle, in denen jede Regel scheitert: mehrteilige Nachnamen mit Präposition („van der Berg“) und Namen ohne Trennkomma. Bessere Lösung: *das Formular ändern* und zwei Felder abfragen. Wenn das nicht geht: am Komma teilen, wenn eines da ist, sonst den vollständigen Namen ins Nachnamensfeld und den Vornamen leer lassen — falsch, aber sichtbar falsch statt still falsch.

`message`: 620 Zeichen in ein 500-Zeichen-Feld. Auf 497 kürzen und „...“ anhängen, den vollständigen Text in ein Notizfeld oder als Anhang mitgeben. Die Kürzung muss sichtbar sein — sonst glaubt der Vertrieb, das sei die ganze Anfrage.

`consent_news` **fehlend:** Gefährlicher als `"off"`, weil ein fehlendes Feld je nach Plattform als „nicht ändern“ interpretiert werden kann — bei einem bestehenden Kontakt bliebe eine frühere Einwilligung stehen. Bei einer Einwilligung ist das rechtlich heikel. Regel: fehlend wird ausdrücklich zu `false` gemappt, nie ausgelassen.

Zusatz — Konfliktregeln: `company` und `budget` aus der Quelle (aktueller); Vertriebsnotiz und Kundenbetreuer aus dem Ziel (dort gepflegt); ein abweichender Firmenname geht in die Prüfliste, weil die Abweichung selbst die Information ist (4.5).

Übung 5 — Fehlerbehandlungsplan

Fehlerklassen: Vorübergehend — API nicht erreichbar, 429 bei 400 Artikeln. Dauerhaft — Token abgelaufen, Artikelnummer im Shop unbekannt (404). Fachlich — Preis 0,01 €, Preissprung +900 %, Währung plötzlich USD statt EUR.

Wiederholung: nur bei der ersten Klasse, 1/2/4/8 Sekunden, höchstens vier Versuche, dann Fehlerablage mit Artikelnummer. Die Preisaktualisierung ist ein `PUT` auf denselben Zielzustand und damit idempotent — Wiederholung ist hier unbedenklich.

Fehlerablage: Zeitpunkt, Szenario, Artikelnummer, Schritt, Fehlermeldung, Status. Gesehen wird sie täglich um 8 Uhr von der Person, die die Preispflege verantwortet — nicht von der IT, weil die Entscheidungen fachlich sind.

0,01 € statt 129 €: fachlicher Fehler. Abgefangen durch eine Plausibilitätsprüfung vor dem Schreiben: Abweichung zum bisherigen Preis über einem Schwellenwert — etwa $\pm 50\%$ — führt nicht zur Aktualisierung, sondern in die Prüfliste. Diese eine Regel ist wertvoller als die gesamte übrige Fehlerbehandlung, weil sie den teuersten Fall abfängt.

Freigabeschritt: Ja, aber nicht für alle Artikel — das wären 400 Klicks. Nur für die Ausreißer über dem Schwellenwert. Entscheidungsgrundlage: Artikel, alter Preis, neuer Preis, Abweichung in Prozent, Datum der letzten Änderung.

Wächter: Ein zweites Szenario, täglich um 8 Uhr: Gab es seit 6 Uhr einen erfolgreichen Lauf, und lag die Zahl der verarbeiteten Artikel im üblichen Bereich? Wenn nein, Nachricht an den Verantwortlichen. Zehn Minuten Aufwand.

Zusatz — Alarmtext: Betreff und Inhalt nach dem Muster aus 5.3: Szenario, Zeitpunkt mit Zeitzone, betroffene Artikelnummern, gescheiterter Schritt, Grund und die konkrete Handlung mit Link — nicht „Details im Protokoll“.

Übung 6 — Wirtschaftlichkeit und Grenzen

Teil A. Nutzen: $40 \times 52 = 2.080$ Belege $\times 4$ Min $\times 0,55$ € = **4.576 €**.

Plattform	360 €		
KI-Schritt	$2.080 \times 0,01$ €	=	21 €
Aufbau	30×70 € = 2.100 €, /3	=	700 €
Wartung	Faustregel: 2 $\times$ Aufbau- anteil pro Jahr	=	1.400 €
Fehlerkosten	$2.080 \times 4\% \times 12$ € / 0,6	=	1.664 €

Summe Kosten			4.145 €
Ergebnis	4.576 - 4.145	=	+ 431 €

Es lohnt sich — aber nur knapp, und jede Schätzung kann die Rechnung kippen. **Kipppunkt:** Bei etwa 36 Belegen pro Woche wird das Ergebnis negativ, weil die Fixkosten aus Aufbau und Wartung nicht mit der Menge sinken.

Die wirksamste Maßnahme ist nicht die Fehlerquote, sondern die *Entdeckungsquote*. Von 60 % auf 95 % zu kommen senkt die Fehlerkosten von 1.664 € auf 1.051 € — ein Gewinn von über 600 €, allein durch eine gute Fehlerablage und eine Plausibilitätsprüfung. Die Fehlerquote von 4 % auf 2 % zu halbieren bringt ähnlich viel, ist aber deutlich schwerer: Sie hängt an der Belegqualität, die du nicht kontrollierst. Man verbessert das, worauf man Zugriff hat.

Sensitivität: Menge, Minuten und Minutensatz wirken identisch, weil sie im Nutzen multipliziert werden — je –25 % ergibt einen Nutzen von 3.432 € und damit ein Ergebnis von rund –713 €. Das Vorhaben trägt also keine 25 Prozent Abweichung auf der Nutzenseite. Genau das gehört in die Entscheidungsvorlage.

Teil B. Werkzeuge, die der Agent bekommt: Belegablage lesen, Buchhaltungssystem lesen, Stammdaten lesen. Verweigert: Buchung anlegen und Mail versenden — beides mit Außenwirkung beziehungsweise finanzieller Wirkung. Der Agent bereitet vor, ein Mensch bucht.

Freigabeschritt vor der Buchung, mit: Beleg als Bild, erkannter Betrag, Kostenstelle, Vorsteuer, Konfidenz des Agenten und — die wichtigste Angabe — welche Belege er *nicht* zuordnen konnte.

Guter Fall: Ein Beleg ohne Kostenstelle; der Agent schaut die letzten drei Reisen derselben Person nach und schlägt die dort verwendete vor. Ein festes Szenario hätte abgebrochen. Schlechter Fall: Ein unleserlicher Beleg über 4,80 €; der Agent schlägt sich mehrfach damit herum, macht acht Modellaufrufe und kostet mehr, als der Beleg wert ist — ein festes Szenario hätte ihn nach einem Versuch in die Prüfliste gelegt.

Zusatz — harte Grenzen: höchstens fünf Modellaufrufe je Beleg, und höchstens 0,10 € Modellkosten je Beleg; bei Überschreitung Abbruch und Prüfliste. Beide Zahlen ergeben sich aus dem Belegwert, nicht aus dem Bauchgefühl.

Selbsttests

Modul 1: 1 b · 2 b **Modul 2:** 1 b · 2 b **Modul 3:** 1 c · 2 b

Modul 4: 1 b · 2 b **Modul 5:** 1 b · 2 c **Modul 6:** 1 b · 2 c

Musterprüfung A

NR.	LÖSUNG	BEGRÜNDUNG
1	b	Abfrage läuft im Intervall, unabhängig davon, ob es Neues gibt (1.1).
2	b	Ab etwa 20 % Ausnahmen zuerst den Prozess bereinigen (1.2).
3	b	Zweimal ausgeführt = einmal ausgeführt im Ergebnis (2.2).
4	b	Stabil aus der Quelle, eindeutig im Ziel (2.2).
5	b	Kein sichtbarer Zwischenzustand bei Abbruch (2.3).
6	c	POST legt an — zweimal heißt zweimal (3.1).
7	c	429 ist vorübergehend; 4xx sonst nicht (3.2).
8	b	409 liefert meist die vorhandene Kennung mit (3.2, Beispiel 3.1).
9	b	Die Adresse ist öffentlich erreichbar; Signaturprüfung ist nötig (3.4).
10	b	Späte Bestätigung führt zu Wiederholungen des Absenders (3.4, Beispiel 2.1).
11	b	ISO 8601 mit Zeitzone (4.2).
12	b	0 ist ein Wert und überschreibt (4.2).
13	b	Ohne Trennzeichen, oft in Cent (4.2).
14	c	Nur vorübergehende Fehler (5.1, 5.2).
15	b	Ohne Idempotenz vervielfacht Wiederholung den Schaden (5.2).
16	b	Keine Fehlermeldung, nur keine Ausführungen (5.5).
17	b	Geld ist eines der fünf Kriterien (5.4).
18	b	Ohne Entscheidungsgrundlage wird im Reflex geklickt (5.4).
19	b	Format erzwingen, regelbasiert validieren, verzweigen (6.1).
20	c	Wartung ist rund zwei Drittel und fehlt fast immer (6.3).

Musterprüfung B

NR.	LÖSUNG	BEGRÜNDUNG
1	b	Der Webhook ist schnell, aber nicht zuverlässig; der Abgleich holt Verpasstes nach (1.1).
2	b	Ohne Idempotenz verdoppelt der Abgleich, was er reparieren soll (1.1, 2.2).
3	b	Jeder Schritt hinter einem Filter, der ohnehin abbricht, kostet Geld (2.1).
4	b	Überlappende Bedingungen lassen beide Wege laufen (2.1).
5	b	Alles hinter dem Iterator läuft je Position — 400 statt 3 (2.1, 2.5).
6	b	Nichts hat nach außen gewirkt; Wiederaufnahme genügt (2.3).
7	b	Sonst hält die Idempotenzprüfung den echten Vorgang für erledigt (2.4).
8	b	Gegen die Spitzenmenge rechnen, nicht gegen den Durchschnitt (2.5).
9	b	PUT setzt einen Zielzustand und ist deshalb wiederholbar (3.1).
10	b	202 heißt angenommen, nicht verarbeitet — Ergebnis später abholen (3.2).
11	c	OAuth erlaubt begrenzten Umfang und einzelnen Entzug (3.3).
12	b	Ohne Schleife werden 50 von 900 verarbeitet — und Erfolg gemeldet (3.5).
13	b	Ohne Zeitzone ist die Angabe mehrdeutig (4.2).
14	b	Ein Geburtstag ist ein Datum, kein Zeitpunkt (4.2).
15	b	Für im Ziel gepflegte Felder gewinnt das Ziel (4.5).
16	b	Ein 5xx heißt: keine verlässliche Antwort — nachsehen statt wiederholen (5.2).
17	c	Nur die Kennung, nicht der ganze Datensatz — Wiederanlauf und Datensparsamkeit (5.3).
18	b	Alarmmüdigkeit: gefilterte Alarmer fehlen im Ernstfall (5.3).
19	b	Schritt- und Kostenobergrenze begrenzen den Sonderfall (6.2).
20	b	Vier gegen 36 ist zu wenig Abstand; das Ergebnis hängt an der Menge (6.4).

KERNAUSSAGEN

- **18-20 richtig** — sehr sicher, tritt zur Prüfung an.
- **14-17 richtig** — bestanden, aber knapp. Die echte Prüfung enthält Szenariofragen mit Ablaufskizzen; arbeite die betroffenen Module nach.
- **unter 14** — noch nicht ausreichend. Wiederhole die Module einschließlich der Übungen an einer echten Plattform und die Musterprüfung frühestens nach zwei Tagen.
- Bei beiden Musterprüfungen gilt: Wer die *Begründung* zu jeder Antwort geben kann, besteht auch Fragen, die anders formuliert sind. Wer nur die Buchstaben kennt, besteht sie nicht.

Auswertung nach Modulen

Trage deine Fehler aus beiden Musterprüfungen in diese Tabelle ein. Ab zwei Fehlern in einem Modul lohnt sich das Nacharbeiten mehr als eine dritte Musterprüfung.

MODUL	FRAGEN IN A	FRAGEN IN B	FEHLER A	FEHLER B	NACHARBEITEN?
1 · Grundlagen	1, 2	1, 2	□□	□□	ab 2
2 · Szenarien	3, 4, 5	3-8	□□□	□□□□□□	ab 2
3 · Webhooks/API	6-10	9-12	□□□□□	□□□□	ab 2
4 · Zuordnung	11, 12, 13	13, 14, 15	□□□	□□□	ab 2
5 · Fehler/Mensch	14-18	16, 17, 18	□□□□□	□□□	ab 2
6 · KI/Wirtschaft	19, 20	19, 20	□□	□□	ab 2

Fallstudien

Drei Vorhaben von der Idee bis zum laufenden Betrieb

Drei Vorhaben mit den Entscheidungen, die unterwegs getroffen wurden, und den Fehlern, die dabei gemacht wurden. Die Fälle sind zusammengesetzt und anonymisiert; die Zahlen sind Modellwerte.

Fallstudie 1 — Bestellungen vom Shop ins ERP

Ausgangslage	Ein Handelsunternehmen überträgt 150 Bestellungen pro Woche von Hand aus dem Shop ins ERP. Zwei Mitarbeitende, je rund sechs Minuten pro Bestellung.
Auslöser	Webhook des Shops bei Bestellabschluss, ergänzt um einen nächtlichen Abgleichslauf über die letzten 48 Stunden.
Mengengerüst	7.800 Bestellungen im Jahr, je 1 bis 12 Positionen, Spitzenlast am Montagmorgen und vor Feiertagen.
Kritische Felder	Bestellnummer (Deduplikationsschlüssel), Betrag in Cent, Lieferadresse, Rabatt.

Was in der ersten Fassung schiefging

1. **Doppelte Bestellungen.** Der Shop wiederholte Webhooks, weil die Bestätigung erst nach der ERP-Anlage kam (Beispiel 2.1). 3 Wochen unbemerkt, 214 Doppelungen.
2. **Gelöschte Rabatte.** Fehlendes Rabattfeld wurde als 0 übertragen und überschrieb Kundenrabatte im ERP (4.2).
3. **Unvollständige Aufträge.** Bei Positionen wurde der Auftragskopf angelegt und der Positionslauf scheiterte — Aufträge ohne Positionen, die in der Kommissionierung auftauchten (4.3).
4. **Stiller Ausfall.** Nach einem Tokenwechsel lief das Szenario zwölf Tage nicht; aufgefallen ist es der Buchhaltung (5.5).

Die zweite Fassung

MASSNAHME	MODULBEZUG	AUFWAND
Webhook sofort bestätigen, Verarbeitung danach	3.4	1 Std.
Bestellnummer als Fremdkennung im ERP, Prüfung als Schritt 0	2.2	3 Std.
Zuordnungstabelle mit Leerwertregeln je Feld	4.4	2 Std.
Auftragskopf mit Status „unvollständig“, erst nach Positionen freigeben	4.3	2 Std.
Fehlerablage mit sechs Spalten, Alarm an die Auftragsannahme	5.3	3 Std.
Wächter: täglicher Lauf und Mengenabweichung > 50 %	5.5	1 Std.
Testauslösung mit Präfix <code>TEST-</code>	2.4	1 Std.

Ergebnis nach zwölf Monaten

KENNZAHL	VORHER (HANDARBEIT)	ERSTE FASSUNG	ZWEITE FASSUNG
Bearbeitungszeit je Bestellung	6,0 Min	0,4 Min	0,5 Min
Fehlerhafte Übertragungen	ca. 1,5 %	3,1 %	0,7 %
davon unentdeckt	ca. 15 %	ca. 80 %	ca. 10 %
Nacharbeit pro Monat	—	9 Std.	1,5 Std.
Jahresergebnis	—	– 2.100 €	+ 14.800 €

DIE LEHRE AUS FALLSTUDIE 1

Die erste Fassung war nicht langsamer als die zweite — sie war *schneller* gebaut und im Ergebnis teurer als Handarbeit. Der Unterschied liegt vollständig in den sieben Maßnahmen der zweiten Fassung, die zusammen dreizehn Stunden gekostet haben. Dreizehn Stunden gegen 16.900 € Ergebnisunterschied im ersten Jahr.

Fallstudie 2 — Bewerbungen ins Bewerbermanagement

Ausgangslage	Ein mittelständisches Unternehmen erhält 15 Bewerbungen pro Woche an ein Sammelpostfach; die Personalabteilung legt sie von Hand im Bewerbermanagement an.
Besonderheit	Personenbezogene Daten, teils besondere Kategorien (Schwerbehindertenausweis in Anlagen). Betriebsrat ist zu beteiligen.
Auslöser	Neue Nachricht im Postfach; Filter auf Betreff und Absender.
KI-Schritt	Auslesen von Name, Position und Anschreiben aus dem Text; Ausgabe als JSON mit Konfidenzfeld (6.1).

Entscheidungen, die den Fall geprägt haben

- **Keine automatische Bewertung.** Der KI-Schritt liest aus, er bewertet nicht — Entscheidungen über Personen fallen unter Kriterium 3 aus 5.4 und gehören zu einem Menschen. Diese Festlegung stand vor dem ersten Klick und war die Grundlage der Zustimmung des Betriebsrats.
- **Anlagen werden nicht durch die Plattform geschleust.** Statt der Datei wird ein Verweis übergeben; die Datei wandert direkt vom Postfach in die Ablage. Das begrenzt die Datenverarbeitung auf das Nötige und umgeht die Größenbegrenzung (2.5).
- **Konfidenz unter 0,8 → Freigabeschritt.** Rund 12 % der Bewerbungen landen dort — genau die, bei denen der Text ungewöhnlich aufgebaut ist.
- **Löschfrist von Anfang an.** Die Fehlerablage enthält nur die Nachrichten-ID, nie den Inhalt; Einträge werden nach 30 Tagen gelöscht.

Zahlen

KENNZAHL	WERT
Bewerbungen pro Jahr	780
Zeit je Bewerbung vorher / nachher	9 Min / 2 Min
Anteil im Freigabeschritt	12 %
Anteil in der Prüfliste (kein Anhang, Fließtext)	9 %
Aufbau	26 Std.
Wartung pro Monat	1,5 Std.
Jahresergebnis	+ 2.900 €

WARUM DAS ERGEBNIS NIEDRIGER IST ALS BEI FALLSTUDIE 1 — UND TROTZDEM RICHTIG

780 Vorgänge sind wenig; die Rechnung trägt nur knapp. Der eigentliche Gewinn lag woanders: Die Durchlaufzeit von der Bewerbung bis zur Rückmeldung sank von durchschnittlich vier Tagen auf einen. In einem angespannten Bewerbermarkt ist das der Wert, der zählt — er steht aber in keiner Wirtschaftlichkeitsrechnung. Halte solche Effekte getrennt fest, statt sie in die Geldrechnung zu pressen.

Fallstudie 3 — Lieferantenpreise im Shop

Ausgangslage	Täglicher Abruf von Preisänderungen aus einer Lieferanten-API, 50 bis 400 Artikel je Lauf.
Auslöser	Zeitplan, täglich 06:00 — bewusst vor Geschäftsbeginn, damit Ausreißer bis 08:00 geprüft werden können.
Besonderheit	Unmittelbare Außenwirkung: Ein falscher Preis ist im Shop sofort öffentlich und kann bestellt werden.

Der Vorfall

Im vierten Betriebsmonat lieferte die Lieferanten-API für 38 Artikel Preise in einer anderen Einheit — Cent statt Euro. Die Preise im Shop fielen auf ein Hundertstel. Zwischen 06:04 und 08:15 gingen elf Bestellungen ein, deren Warenwert bei korrektem Preis rund 4.100 € betragen hätte.

Die Plausibilitätsprüfung war zu diesem Zeitpunkt bereits eingebaut — aber mit einem Schwellenwert von ± 80 Prozent und nur nach oben. Ein Preissturz auf ein Hundertstel lag außerhalb der Prüfung, weil niemand ihn für möglich gehalten hatte.

Was danach geändert wurde

1. Schwellenwert symmetrisch: Abweichung über ± 50 Prozent in *beide* Richtungen führt in die Prüfliste, nicht zur Aktualisierung.
2. Zusätzliche Prüfung gegen den Einkaufspreis: Ein Verkaufspreis unter dem Einkaufspreis wird nie geschrieben, unabhängig von der prozentualen Abweichung.

3. Mengenprüfung: Ändern sich in einem Lauf mehr als 20 Prozent aller Artikel um mehr als 20 Prozent, wird der gesamte Lauf angehalten und gemeldet — ein Einheitenwechsel betrifft immer viele Artikel gleichzeitig.
4. Freigabeschritt nur für Ausreißer, mit Artikel, altem Preis, neuem Preis, Abweichung und Datum der letzten Änderung.
5. Wächter auf Mengenabweichung nach unten *und* oben.

DIE ALLGEMEINE LEHRE

Die dritte Maßnahme ist die interessanteste, weil sie nicht den einzelnen Datensatz prüft, sondern den *Lauf*. Fast alle Massenfehler — Einheitenwechsel, Zeichensatzwechsel, vertauschte Spalten, leerer Export — haben dieselbe Signatur: Sie betreffen viele Datensätze gleichzeitig und auf ähnliche Weise. Eine einzige Prüfung auf Laufebene fängt eine ganze Klasse von Fehlern ab, für die man sonst zwanzig Einzelprüfungen bräuchte.

Vorlagen und Arbeitsblätter

Sechs Arbeitsblätter zum Übernehmen

Bewusst kurz gehalten — eine Vorlage, die eine Seite überschreitet, wird nicht ausgefüllt.

C.1 Prozessbeschreibung

Prozess: _____
 Verantwortlich: _____ Datum: _____

1 AUSLÖSER
 Was startet den Ablauf? _____
 Wie oft? ____ / Woche Spitzenmenge: ____ / Tag
 Typ: ☐ Webhook ☐ Abfrage ☐ Zeitplan

2 EINGABEN
 Quellsystem: _____ Format: _____
 Pflichtfelder: _____

3 REGELN
 Entscheidung 1: _____
 Entscheidung 2: _____
 Stabil seit: _____ Änderung erwartet: ☐ ja ☐ nein

4 WIRKUNG
 Zielsystem(e): _____
 Außenwirkung: ☐ keine ☐ Mail ☐ Geld ☐ Person

5 AUSNAHMEN (mindestens drei, je mit Festlegung)
 a) _____ -> _____
 b) _____ -> _____
 c) _____ -> _____

BEWERTUNG (0-3 je Kriterium, Produkt bilden)
 Häufigkeit ____ x Stabilität ____ x Ausnahmen ____ x Fehler ____ = ____

C.2 Zuordnungstabelle

QUELLFELD	ZIELFELD	TYP	UMWANDLUNG	WENN LEER	BEI KONFLIKT	PERSONENBEZUG

Je Feld eine Zeile. „Wenn leer“ darf nie „automatisch“ lauten. „Bei Konflikt“: Quelle / Ziel / Prüfliste.

C.3 Fehlerbehandlungsplan

Szenario: _____ Version: _____

KLASSE	FEHLER (konkret)	REAKTION
Vorübergehend	_____	Wdh. ____ x, Abstand _____
	_____	Wdh. ____ x, Abstand _____
Dauerhaft	_____	Fehlerablage + Alarm an _____
	_____	Fehlerablage + Alarm an _____
Fachlich	_____	Prüfliste, gesehen von _____
	_____	Prüfliste, gesehen von _____

IDEMPOTENZ

Schlüssel: _____ Quelle: _____
Gespeichert in: _____ Prüfung an Position: _____

PLAUSIBILITÄT (die Prüfung, die den teuersten Fall abfängt)

Feld _____ Bereich _____ sonst -> Prüfliste
Auf Lauebene: mehr als ____ % der Sätze um mehr als ____ % -> Stopp

WÄCHTER

Prüft: ☐ Lauf erfolgt ☐ Menge plausibel ☐ Ablage wächst
☐ Zugangsdaten laufen ab in ____ Tagen
Läuft: _____ Meldet an: _____

C.4 Freigabeschritt

Anlass der Freigabe (Kriterium aus 5.4): _____

ENTSCHEIDUNGSGRUNDLAGE (3 bis 5 Angaben, keine Links zum Anmelden)

1 _____ 2 _____
3 _____ 4 _____

AUSWAHLMÖGLICHKEITEN

☐ Zustimmung] -> _____
☐ Ablehnen: Grund] -> _____
☐ Weiterleiten an _____] -> _____

ZEITÜBERSCHREITUNG

Nach ____ Stunden: ☐ Erinnerung ☐ Eskalation an _____
☐ Abbruch ☐ NIE: automatisch zustimmen

MENGE

Erwartete Freigaben je Lauf: ____ (bei > 20: auf Ausreißer beschränken)

C.5 Wirtschaftlichkeitsrechnung

NUTZEN

Vorgänge / Jahr _____
× eingesparte Minuten je Vorgang _____
× Minutensatz _____ €
= Nutzen je Jahr _____ €

KOSTEN

Plattform je Jahr _____ €
KI-Aufrufe: _____ × _____ € _____ €
Aufbau: _____ Std × _____ € = _____ € , / 3 _____ €
Wartung je Jahr (Faustregel: 2 × Aufbauanteil) _____ €
Fehlerkosten:
Vorgänge _____ × Fehlerquote _____ %
× Korrektur _____ € ÷ Entdeckungsquote _____ = _____ €
= Kosten je Jahr _____ €

ERGEBNIS

_____ €
KIPPPUNKT (Menge, bei der das Ergebnis 0 wird) _____
ABSTAND zur erwarteten Menge _____ %

SENSITIVITÄT (je Annahme -25 %)

Menge _____ € Minuten _____ € Satz _____ €
Aufbau _____ € Wartung _____ € Entdeckung _____ €

C.6 Übergabeblatt für den Betrieb

Szenario: _____ Seit: _____

Fachlich verantwortlich: _____

Technisch verantwortlich: _____

Vertretung: _____

LÄUFT Takt: _____ Übliche Menge je Lauf: _____ bis _____

VERBINDUNGEN System _____ Konto _____ läuft ab _____
System _____ Konto _____ läuft ab _____

FEHLERABLAGER Ort: _____ Gesehen von: _____ täglich _____ Uhr

PRÜFLISTE Ort: _____ Gesehen von: _____ täglich _____ Uhr

ALARM Kanal: _____ Erreicht: _____

WAS ZU TUN IST, WENN ...

... der Wächter meldet: _____
... die Fehlerablage wächst: _____
... ein Zugang abgelaufen ist: _____
... das Quellsystem sich ändert: _____

ABSCHALTEN Wer darf: _____ Wie: _____
Was danach von Hand geschieht: _____

Betrieb, Übergabe, Governance

Was nach der Inbetriebnahme geschieht — und wer es tut

Dieser Anhang ist nicht prüfungsrelevant und im Alltag der wichtigste Teil dieses Handbuchs.

D.1 Der Betriebszyklus

Eine Automatisierungslandschaft braucht wöchentlich etwa zwanzig Minuten Aufmerksamkeit. Weniger führt dazu, dass sich Fehler ansammeln; mehr deutet darauf hin, dass eine der Maßnahmen aus Modul 5 fehlt.

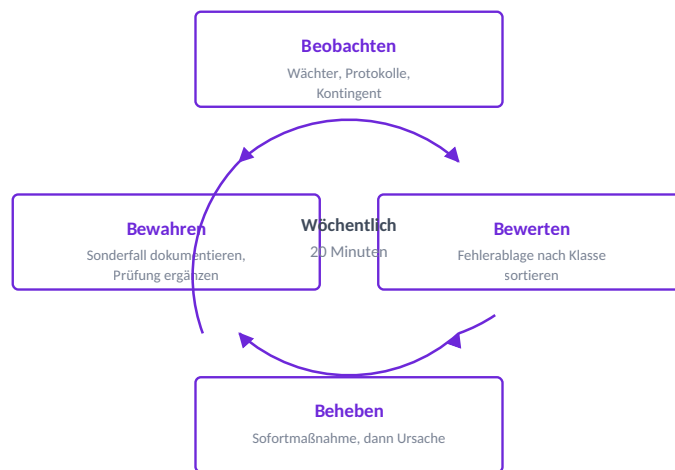


Abb. A.1 Der Betriebszyklus. Ohne Schritt 4 wiederholt sich Schritt 3 — und die Wartungskosten aus Modul 6 steigen genau dort.

Der vierte Schritt — **Bewahren** — ist der, der weggelassen wird, und der einzige, der die Wartungskosten senkt. Wer einen Sonderfall behebt, ohne die Prüfung zu ergänzen und den Anlass zu dokumentieren, behebt ihn in drei Monaten erneut.

D.2 Rollen und Zuständigkeiten

AUFGABE	FACHBEREICH	BAUENDE PERSON	IT / BETRIEB
Prozessbeschreibung und Ausnahmen festlegen	V	M	—
Szenario bauen und testen	M	V	—
Zugangsdaten und technische Konten	—	M	V
Fehlerablage (technische Fälle) abarbeiten	—	V	M
Prüfliste (fachliche Fälle) abarbeiten	V	—	—
Freigaben erteilen	V	—	—
Wächter überwachen	—	V	M
Änderung des Quellsystems melden	M	M	V
Abschaltung entscheiden	V	M	M

V = verantwortlich, M = mitwirkend. Die entscheidende Zeile ist die fünfte: Fachliche Fälle gehören nicht in die IT.

DIE ZUSTÄNDIGKEIT, DIE AM HÄUFIGSTEN FEHLT

In den meisten gescheiterten Automatisierungen war nicht die Technik das Problem, sondern dass niemand benannt war, der die Prüfliste ansieht. Eine Automatisierung ohne benannte fachliche Zuständigkeit ist unvollständig — sie erzeugt Arbeit, die niemandem gehört, und die niemandem gehörende Arbeit bleibt liegen.

D.3 Änderungen an laufenden Szenarien

1. **Vorher sichern.** Export oder Version des laufenden Stands, mit Datum.
2. **Änderung beschreiben.** Ein Satz: was, warum, wer hat es beauftragt.
3. **Mit Testauslösung prüfen** (2.4), nicht in der Produktion.
4. **In einem ruhigen Zeitfenster aktivieren** — nicht Montagmorgen, nicht am Monatsende.
5. **Ersten Lauf beobachten** und die Menge mit dem Üblichen vergleichen.
6. **Übergabebblatt aktualisieren** (C.6), wenn sich Takt, Menge oder Zuständigkeit geändert haben.

D.4 Reifegrad einer Automatisierungslandschaft

Das folgende Modell hilft bei der Frage, wo als Nächstes investiert werden sollte. Es ist kein Zertifizierungsschema, sondern eine Bestandsaufnahme.

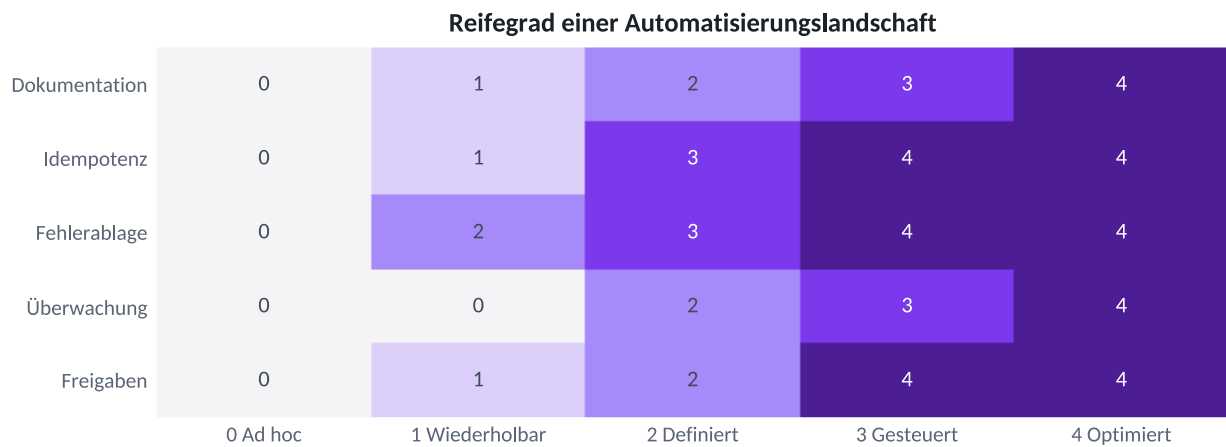


Abb. D.2 Fünf Dimensionen, fünf Stufen. Die Erfahrung ist eindeutig: Überwachung hinkt fast immer hinterher — sie wird erst gebaut, nachdem ein stiller Ausfall wehgetan hat.

STUFE	WORAN MAN SIE ERKENNT	NÄCHSTER SCHRITT
0 Ad hoc	Einzelne Szenarien, gebaut von einzelnen Personen, keine Dokumentation	Prozessbeschreibungen nachziehen (C. 1)
1 Wiederholbar	Benennungskonventionen, Zuordnungstabellen vorhanden	Idempotenz in allen schreibenden Szenarien (2.2)
2 Definiert	Fehlerablage und Prüfliste existieren und werden abgearbeitet	Wächter und Alarmierung (5.5)
3 Gesteuert	Überwachung läuft, Zuständigkeiten sind benannt	Kennzahlen: Menge, Fehlerquote, Entdeckungsquote je Szenario
4 Optimiert	Entscheidungen über neue Vorhaben werden mit Kippunkt und Sensitivität begründet	Bestand regelmäßig auf Abschaltkandidaten prüfen

D.5 Datenschutz und rechtlicher Rahmen

Dieser Abschnitt ersetzt keine Rechtsberatung. Er benennt die Punkte, die bei Automatisierungen mit personenbezogenen Daten regelmäßig zu klären sind, und ordnet sie den Modulen zu.

PUNKT	WAS ZU KLÄREN IST	BEZUG
Rechtsgrundlage	Auf welcher Grundlage werden die Daten verarbeitet — Vertrag, berechtigtes Interesse, Einwilligung?	vor dem Bauen
Datensparsamkeit	Welche Felder braucht das Szenario wirklich? Die Zuordnungstabelle mit Spalte „Personenbezug“ ist die Grundlage.	4.4
Auftragsverarbeitung	Vertrag mit dem Plattformanbieter; Verarbeitungsort	Plattform-Landkarte
Protokolle	Ausführungsprotokolle enthalten oft vollständige Datensätze. Aufbewahrungsdauer prüfen und begrenzen.	2.5
Fehlerablage	Nur Kennungen speichern, nicht die Inhalte; Löschfrist festlegen	5.3
Automatisierte Entscheidungen	Entscheidungen mit rechtlicher Wirkung gegenüber Personen brauchen menschliche Beteiligung	5.4
Auskunft und Löschung	Wo überall liegt eine Kopie — Zielsystem, Fehlerablage, Protokolle, Zwischenspeicher?	D.6
KI-Einsatz	Welche Daten gehen an das Modell, wo wird verarbeitet, werden sie zum Training verwendet?	6.1

DIE FRAGE, DIE EINE AUTOMATISIERUNG DATENSCHUTZRECHTLICH AM HÄUFIGSTEN AUFFÄLLIG MACHT

Nicht das Zielsystem, sondern die *Nebenablagen*: Ausführungsprotokolle, Fehlerablagen, Zwischentabellen, Testdaten. Sie entstehen beiläufig, enthalten oft vollständige Datensätze und werden bei Auskunfts- und Löschbegehren vergessen. Wer beim Bauen eine Liste aller Orte führt, an denen Daten liegen bleiben, erspart sich die Suche später.

D.6 Abschaltung

Automatisierungen werden gebaut und selten abgeschaltet. Ein jährlicher Blick auf den Bestand lohnt sich: Szenarien, deren Prozess es nicht mehr gibt, laufen weiter, verbrauchen Kontingent und erzeugen Fehler, die niemand zuordnen kann.

1. Läuft der zugrunde liegende Prozess noch? Wenn nein: abschalten.
2. Stimmt die Menge noch mit der Annahme überein? Wenn sie deutlich gefallen ist: Kippunkt neu rechnen (6.4).
3. Gibt es eine benannte fachliche Zuständigkeit? Wenn nein: benennen oder abschalten.
4. Beim Abschalten: Zugänge entziehen, Fehlerablage archivieren, Daten nach Löschfrist entfernen, Übergabeblatt als „stillgelegt“ kennzeichnen — mit Datum und Grund.

Glossar

Vierzig Begriffe, die in diesem Handbuch und in der Prüfung in genau dieser Bedeutung verwendet werden

Abfrage (Polling)

Auslöserart, bei der regelmäßig nach neuen Daten gefragt wird — unabhängig davon, ob es welche gibt.

Aggregator

Baustein, der iterierte Einzelvorgänge wieder zu einem Ergebnis zusammenfügt.

Alarmierung

Nachricht über einen Fehler an eine Person oder einen Kanal mit klarer Zuständigkeit — mit Inhalt, nicht nur mit Hinweis.

Alarmmüdigkeit

Zustand, in dem Alarme ignoriert oder gefiltert werden, weil sie zu häufig grundlos auslösen.

Auslöser

Das Ereignis, das ein Szenario startet: Webhook, Abfrage oder Zeitplan.

Außenwirkung

Wirkung eines Schritts außerhalb der eigenen Systeme — Mail, Zahlung, Veröffentlichung. Nicht zurückholbar.

Backoff

Wachsender Abstand zwischen Wiederholungsversuchen, meist exponentiell.

Bearer-Token

Zeitlich begrenzter Zugangsschlüssel, der in einer Kopfzeile mitgeschickt wird.

Deduplikationsschlüssel

Stabile Kennung aus dem Quellsystem, im Zielsystem gespeichert, um Doppelverarbeitung zu erkennen.

Entdeckungsquote

Anteil der Fehler, der tatsächlich bemerkt wird. Der Faktor, der die Fehlerkosten am stärksten bestimmt.

Fehlerablage

Tabelle mit allen Fällen, die nicht automatisch verarbeitet werden konnten — die Grundlage jeder Nachbearbeitung.

Fehlerklasse

Einteilung in vorübergehend, dauerhaft und fachlich; bestimmt die Reaktion.

Filter

Baustein, der einen Ablauf nur unter Bedingungen weiterlaufen lässt.

Freigabeschritt

Punkt im Ablauf, an dem ein Mensch entscheidet, bevor es weitergeht.

Idempotenz

Eigenschaft eines Ablaufs, zweimal ausgeführt dasselbe Ergebnis zu liefern wie einmal.

ISO 8601

Datumsformat mit Zeitzone, z. B. 2026-08-30T14:05:00+02:00.

Iterator

Baustein, der eine Liste in Einzelvorgänge zerlegt.

Jitter

Zufälliger Anteil in der Wartezeit zwischen Wiederholungen, damit nicht alle blockierten Abläufe gleichzeitig erneut anfragen.

Kompensation

Rückgängigmachen bereits ausgeführter Schritte nach einem Abbruch.

Konfidenz

Selbsteinschätzung eines Modells zur eigenen Sicherheit. Als Sortierkriterium nützlich, nicht als Wahrscheinlichkeit.

Kontingentgrenze

Begrenzung der Aufrufe je Zeiteinheit; Überschreitung führt zu 429.

Kopfzeile (Header)

Zusatzangabe zu einer HTTP-Anfrage oder -Antwort, etwa Authentifizierung oder Wartezeit.

Laufebenen-Prüfung

Plausibilitätsprüfung, die nicht den einzelnen Datensatz, sondern den gesamten Lauf betrachtet — fängt Massenfehler ab.

Leerwert

Sammelbegriff für null, leere Zeichenkette, fehlendes Feld und die Zahl 0 — vier verschiedene Zustände.

OAuth

Verfahren zur Anmeldung im Namen eines Kontos, mit begrenztem Umfang und Erneuerung.

Plausibilitätsprüfung

Regelbasierte Prüfung, ob ein Wert fachlich sinnvoll ist — die wirksamste einzelne Maßnahme gegen fachliche Fehler.

Prüfliste

Ablage für fachliche Fälle, die ein Mensch aus dem Fachbereich entscheidet — getrennt von der technischen Fehlerablage.

Seitenweise-Abruf

Auslieferung langer Listen in Portionen; ohne Schleife verarbeitet man nur die erste.

Sensitivität

Auswertung, welche Annahme das Ergebnis einer Rechnung kippt, wenn sie danebenliegt.

Signaturprüfung

Verifikation, dass ein Webhook tatsächlich vom erwarteten Absender stammt.

Statuscode

Dreistellige Zahl in der HTTP-Antwort, die sagt, was mit der Anfrage geschehen ist.

Stiller Ausfall

Das Szenario läuft nicht mehr, ohne dass ein Fehler gemeldet wird.

Szenario

Die Gesamtheit aus Auslöser und Aktionen; je nach Plattform Zap, Workflow oder Flow.

Teilerfolg

Zustand, in dem ein Teil der Schritte ausgeführt ist und der Rest nicht.

Testauslösung

Weg, ein Szenario mit Beispieldaten ohne Außenwirkung zu starten — mit eigenem Schlüsselpräfix.

Verzweigung

Baustein, der den Ablauf in mehrere Wege teilt.

Wächter

Zweites Szenario, das prüft, ob das erste noch läuft.

Wiederaufnahme

Fortsetzung eines Ablaufs ab dem gescheiterten Schritt.

Zeitplan

Auslöserart, die zu festen Zeiten läuft.

Zuordnungstabelle

Aufstellung aller Felder mit Quelle, Ziel, Umwandlung, Leerwertregel und Konfliktregel.

Prüfungsinformationen

Format, Bewertung, Zulassung und der Weg danach

Fragen	20, Einfachauswahl und Szenariofragen mit Ablaufskizzen
Dauer	40 Minuten, ohne Unterbrechung
Bestehensgrenze	70 % (14 von 20)
Voraussetzung	AI Foundations Professional (AF) oder anerkannte Gleichwertigkeit
Hilfsmittel	keine
Bewertung	deterministisch gegen einen von Fachexperten freigegebenen Antwortschlüssel; kein KI-Einsatz
Wiederholung	nach 24 Stunden, im Rahmen der erworbenen Versuche
Gültigkeit	befristet; Rezertifizierung nach den Bedingungen des Zertifizierungsschemas

Geprüfte Kompetenzen und ihre Gewichtung

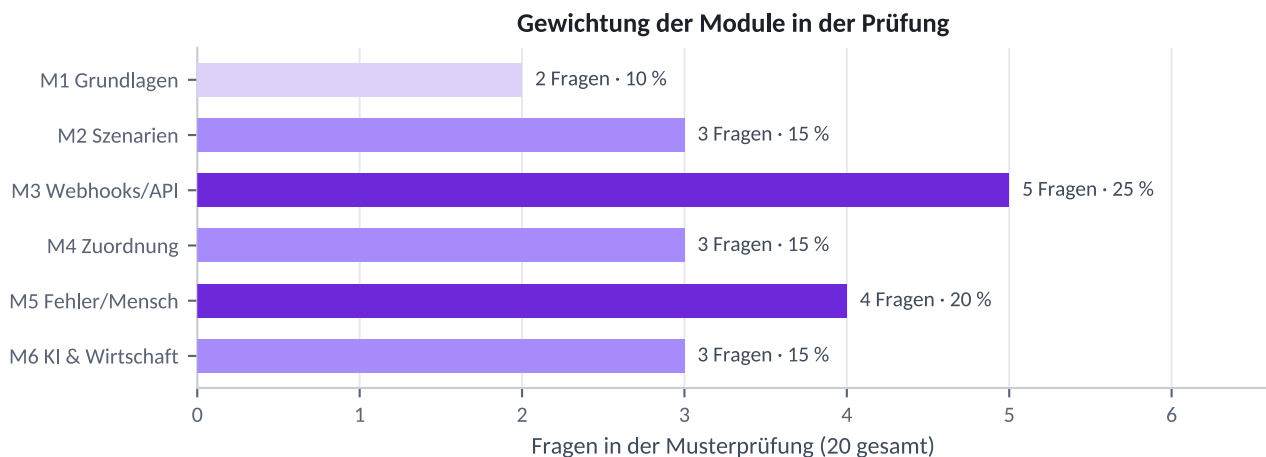


Abb. F.1 Verteilung der Fragen über die Module in der Musterprüfung. Die echte Prüfung folgt derselben Gewichtung; Modul 3 hat das größte Gewicht, Modul 1 das kleinste — obwohl Modul 1 die Entscheidungen enthält, die im Beruf am meisten zählen.

- Entscheidung über Automatisierung — Kriterien, Multiplikatorwirkung, Prozessbeschreibung
- Mehrstufige Szenarien — Verzweigung, Iteration, Idempotenz, Teilerfolg
- Webhooks und API-Grundlagen — Methoden, Statuscodes, Authentifizierung, Kontingente
- Datenzuordnung — Typen, Formate, Leerwerte, Abgleich
- Fehlerbehandlung — Fehlerklassen, Wiederholung, Fehlerablage, Mensch im Ablauf
- KI-Schritte in No-Code-Umgebungen — Absicherung, Werkzeuggrenzen
- Wirtschaftlichkeit — vollständige Rechnung einschließlich Wartung, Kippunkt

Zulassung

Die Teilnahme an diesem Kurs ist **keine** Voraussetzung für die Prüfung. Neben der Kursvorbereitung stehen der direkte Prüfungsweg (Exam-Only) und der Nachweis gleichwertiger Vorkenntnisse offen; die Kriterien sind veröffentlicht. Auch das Foundation-Zertifikat kann durch gleichwertige Nachweise ersetzt werden.

Vorbereitung in der Woche vor der Prüfung

TAG	WAS	DAUER
-7	Musterprüfung A unter realen Bedingungen	1 h
-6	Auswertung, Module mit zwei oder mehr Fehlern nacharbeiten	2 h
-4	Musterprüfung B unter realen Bedingungen	1 h
-3	Auswertung; Übungen 2 und 5 an einer echten Plattform wiederholen	2 h
-1	Kernaussagen aller sechs Module lesen, Glossar durchgehen	45 Min
0	Prüfung	40 Min

Wie es weitergeht

Wer aus konfigurierten Abläufen gebaute Systeme machen will, findet die Fortsetzung in *AI Agent & Automation Architect* (AA). Wer die KI-Schritte in seinen Szenarien zuverlässiger machen will, in *AI Prompt Engineer Specialist* (PE).

CertifAI ist ein unabhängiger Anbieter. Dieses Zertifikat ist ein Industriezertifikat und keine staatlich anerkannte Qualifikation. Es wird von CertifAI ausgestellt und ist nicht behördlich zugelassen.

Teilnehmerhandbuch No-Code AI Automation Expert (NC) · Version 2.0 · Ausgabe DE · Rechtsstand August 2026

Stichwortverzeichnis

Verweise auf Abschnittsnummern, nicht auf Seiten

Abfrage (Polling) 1.1, 0.3	Kompensation 2.3
Abschaltung D.6	Konfidenz 6.1
Aggregator 2.1	Kontingentgrenze 2.5, 3.5
Agent 6.2	Laufzeit 2.5
Alarmierung 5.3	Leerwerte 4.2, 4.4
Alarmmüdigkeit 5.3	Multiplikator 1.3
Anti-Muster 1.5	OAuth 3.3
Auslöser 1.1, C.1	Plausibilitätsprüfung 5.1, B.3, C.3
Außenwirkung 1.1, 2.3, 6.2	Prozessbeschreibung 1.4, C.1
Authentifizierung 3.3	Prüfliste 5.1, D.2
Backoff 3.5, 5.2	Reifegrad D.4
Bearer-Token 3.3	Rollen (V/M) D.2
Bestehensgrenze F	Seitenweise-Abruf 3.5
Datenschutz D.5	Sensitivität 6.4, C.5
Datumsformat 4.2	Signaturprüfung 3.4
Deduplikationsschlüssel 2.2, 2.4, 3.5	Statuscodes 3.2
Entdeckungsquote 1.3, 6.3	Stiller Ausfall 5.5, B.1
Fallstudien B	Szenario 1.1
Fehlerablage 5.3, D.5	Teilerfolg 2.3, 4.3
Fehlerklassen 5.1	Testauslösung 2.4, D.3
Fehlerkosten 1.2, 6.3	Übergabe C.6, D.2
Filter 2.1	Verzweigung 2.1
Freigabeschritt 5.4, C.4	Wächter 5.5, C.3
Geldbeträge 4.2	Wartung 6.3, 6.4
HTTP-Methoden 3.1	Webhook 1.1, 3.4
Idempotenz 2.2, 3.1, 5.2	Wiederaufnahme 2.3
Index G	Wiederholung 5.2
Iterator 2.1, 2.5	Wirtschaftlichkeit 6.3, C.5
Jitter 3.5	Zeitplan 1.1
KI-Schritt 6.1	Zeitzone 4.2
Kipppunkt 6.4, C.5	Zuordnungstabelle 4.4, C.2

Abbildungsverzeichnis

Abb. 0.1 Zertifizierungspfad · 0.2 Abgrenzung der Tracks · 0.3 Aufrufe und Verzögerung je Auslöserart · 0.4 Aufwandsprofile · 0.5 Störungsprofil · 1.1 Anatomie eines Szenarios · 1.2 Auslösertypen im Zeitverlauf · 1.3 Fehlerquote und Entdeckung · 1.4 Entscheidungsbaum · 1.5 Prozessbeschreibung · 2.1 Die vier Bausteine · 2.2 Idempotenz · 2.3 Teilerfolg · 2.4 Reihenfolgeprinzip · 2.5 Zeitbudget eines Webhooks · 3.1 Anatomie einer HTTP-Anfrage · 3.2 Statuscode-Landkarte · 3.3 Authentifizierungsverfahren · 3.4 Webhook-Absicherung · 3.5 Kontingent · 3.6 Wartestrategien · 3.7 Seitenweise-Abruf · 4.1 Zuordnungspipeline · 4.2 Zeitzonenfalle · 4.3 Leerwerte

· 5.1 Fehlerklassen · 5.2 Stiller Ausfall und Wächter · 5.3 Freigabekarten im Vergleich · 5.4 Wartestrategien · 5.5 Freigabe: Aufwand gegen Risiko · 6.1 Aufbau eines KI-Schritts · 6.2 Werkzeug-Risiko-Matrix · 6.3 Kostenstruktur · 6.4 Entdeckungsquote und Fehlerkosten · 6.5 Wirtschaftlichkeit als Wasserfall · 6.6 Kipppunkt · A.1 Betriebszyklus · D.2 Reifegradmodell · F.1 Prüfungsgewichtung

Zu dieser Ausgabe

Titel	Teilnehmerhandbuch No-Code AI Automation Expert (NC)
Ausgabe	Version 2.0 · Deutsch · Rechtsstand August 2026
Umfang	6 Module · 39 Abbildungen · 6 Übungen · 12 Selbsttestfragen · 2 Musterprüfungen · 6 Vorlagen · 3 Fallstudien · 40 Glossarbegriffe
Herausgeber	CertifAI
Zielgruppe	Personen, die Automatisierungen in bestehenden Plattformen konfigurieren und betreiben
Vorausgesetzt	AI Foundations Professional (AF) oder anerkannte Gleichwertigkeit
Fortsetzung	AI Agent & Automation Architect (AA) · AI Prompt Engineer Specialist (PE)

CertifAI ist ein unabhängiger Anbieter. Dieses Zertifikat ist ein Industriezertifikat und keine staatlich anerkannte Qualifikation. Es wird von CertifAI ausgestellt und ist nicht behördlich zugelassen.

Teile dieses Handbuchs wurden KI-gestützt erstellt und anschließend fachlich geprüft und freigegeben. Die Prüfungsfragen werden unabhängig vom Kursmaterial entwickelt; die Bewertung erfolgt deterministisch gegen einen von Fachexperten freigegebenen Antwortschlüssel.

Alle Zahlenbeispiele sind Modellrechnungen mit offengelegten Annahmen und keine Marktdaten. Die Fallstudien in Anhang B sind zusammengesetzt und anonymisiert. Genannte Plattformnamen dienen der Einordnung; es besteht keine Verbindung zu den Herstellern, und keine Plattform wird empfohlen. Anhang D ersetzt keine Rechtsberatung.