ERFAHRUNGSBERICHT · MAI – SEPTEMBER 2026

Hundert Sitzungen mit Claude Code

Ein Erfahrungsbericht aus der Praxis: Joomla-Erweiterungen, Vereins-Tools und Büro-Software – mit allen Umwegen, Fallstricken und Lehren.

Joomla 5 & 6 | 68 Fallstricke | Fallstudien | Arbeitsregeln | Checklisten

Roland Löffler · JoomION

Ausgabe 2026 · Stand: September 2026 · Begleitband zu „Joomla-Erweiterungen entwickeln mit Claude Code“

Inhalt

Teil I · Rückblick

1 Einleitung: Hundert Sitzungen später
2 Zahlen & Zeitleiste
3 Die Projektlandkarte

Teil II · Die Zusammenarbeit

4 Was Claude Code wirklich gut kann
5 Wo es danebenging: zwölf Fehlermuster

Teil III · Joomla-Praxis: Fallstricke

6 Paketbau & Manifest
7 Installer, Updates & Versionen
8 Formulare, Felder & Backend
9 JavaScript, CSS & Assets
10 Laufzeit, Sitzungen & Sicherheit
11 JED-Checker, Audits & Auslieferung

Teil IV · Die Lizenz-Geschichte

12 Vom Lizenzsystem zur freien Software

Teil V · Fallstudien jenseits von Joomla

13 Werkzeuge für einen Fotoverein
14 Büro-Software
15 Abseits vom Code

Teil VI · Lehren

16 Sechzehn Arbeitsregeln
17 Gedächtnis, Regeln, Generatoren
18 Bilanz & Checklisten

Anhang

A Katalog der Fallstricke
B Prompt-Bibliothek aus der Praxis

TEIL I

Rückblick

1 · Einleitung: Hundert Sitzungen später

2 · Zahlen & Zeitleiste

3 · Die Projektlandkarte

KAPITEL 1

Einleitung: Hundert Sitzungen später

> *Dieses Buch ist kein Lehrbuch, sondern ein Protokoll aus der Praxis. Es wertet rund hundert Arbeitssitzungen mit Claude Code aus – mit allem, was gelang, und allem, was schiefging.*

Worum es in diesem Bericht geht

Zwischen Ende Mai und Ende September 2026 ist mit Claude Code eine ganze Werkstatt entstanden: gut zwei Dutzend Joomla-Erweiterungen unter dem Namen **JoomION**, eine Reihe eigenständiger Web-Anwendungen für einen Fotoverein, Büro-Werkzeuge für Rechnungen und Steuer sowie etliche Einzelaufgaben von der Präsentation bis zur Entzerrung eines Kirchenbuchs aus dem Jahr 1864.

Der Begleitband *„Joomla-Erweiterungen entwickeln mit Claude Code“* erklärt, wie man vorgehen sollte. Dieser Bericht zeigt, wie es tatsächlich gelaufen ist. Er beantwortet drei Fragen:

1. **Was ist entstanden?** – Zahlen, Zeitleiste und eine Projektlandkarte (Teil I).
2. **Was kann die KI – und wo lag sie daneben?** – Stärken und typische Fehlermuster (Teil II).
3. **Welche Fallen gibt es?** – 68 dokumentierte Fallstricke mit Symptom, Ursache und Lösung, dazu Fallstudien und Arbeitsregeln (Teile III bis VI).

Die Datengrundlage

Claude Code führt über die Sitzungen hinweg ein **Gedächtnis**: kurze Notizen zu Projekten, Rückmeldungen, technischen Erkenntnissen und Verweisen. Am Ende des Zeitraums umfasste es rund 115 Einträge. Sie bilden – zusammen mit der Sitzungsliste und den tatsächlich erzeugten Dateien im Ausgabeordner – die Grundlage dieses Berichts.

Was dieser Bericht nicht ist

Er ist keine vollständige Mitschrift. Die Notizen halten fest, was sich als wichtig erwiesen hat – meist, weil etwas schiefgelaufen war. Versionsnummern entsprechen dem Stand der Notizen. Zugangsdaten, Kundendaten und Namen Dritter sind bewusst weggelassen.

Wie Sie diesen Bericht lesen

Die Teile bauen nicht aufeinander auf. Wer selbst Joomla-Erweiterungen entwickelt, findet in Teil III und im Anhang A ein Nachschlagewerk. Wer wissen will, wie sich die Zusammenarbeit mit einer KI anfühlt und was man dabei lernen muss, beginnt mit Teil II und Teil VI. Die Fallstudien in Teil V zeigen, dass dieselben Muster weit über Joomla hinaus gelten.

Das Wichtigste in einem Satz

Claude Code hat die Arbeit vervielfacht – aber die meisten Fehler entstanden nicht aus fehlendem Wissen, sondern aus Raten, Übereifer und fehlender Prüfung. Genau dagegen helfen feste Regeln.

KAPITEL 2

Zahlen & Zeitleiste

> *Vier Monate in Zahlen – und wie sich die Arbeit von Monat zu Monat verschoben hat: von großen Joomla-Komponenten über freie Software hin zu Werkzeugen für Verein und Büro.*

Auf einen Blick

≈ 100 Sitzungen mit Claude Code	**4 Monate** 27. Mai – 29. September 2026	**37** Installationspakete im Ausgabeordner	**16** Erweiterungen mit eigenem Update-Feed
20 Audit-Berichte als PDF	**68** dokumentierte Fallstricke	**89** Python-Hilfsskripte	**63** Anleitungs- und FAQ-Dateien (DE/EN)

Die Zahlen stammen aus dem Ausgabeordner zum Ende des Zeitraums. Ältere Pakete, die vor der Einführung der festen Ordnerstruktur entstanden, sind darin nicht enthalten – die tatsächliche Zahl gebauter Versionen liegt deutlich höher. Allein der Security Scanner durchlief in der Reihe 1.00 über 90 Versionen, das Template in der Reihe 1.9 über 130.

Die Zeitleiste

Ende Mai 2026

Der Einstieg

Responsive Fotowettbewerbsseite, Upload-Modul, Schema-Plugin, erste Gespräche zum LicManager, Positionen und Layout des JoomION-Templates.

Juni 2026

Die großen Komponenten

Framework-Builder, Security Scanner (ab 15.06.), Neuaufbau des LicManagers (ab 17.06.), Lizenz- und Update-System (ab 22.06.), JoomION Projekte, Katalog, Updater, erste PSR-12-Audits – und am Monatsende die Entscheidung zur Entlizenzierung.

Juli 2026

Freie Software

Rollout der Entlizenzierung, Fork des Downloaders, Widerruf als eigenständiges Paket (fertig 13.07.), Admin-URL, ScrollBox, JED-Checker-Runden, Dalli-Klick.

August 2026

Werkzeuge für Büro und Verein

Bilderabstimmung mit fairer Rundenformel, Arbeitsnachweis (seitdem live), Rechnungstool in Excel/VBA, Rechnungsauslese mit OCR, die Upload-Diagnose, Countdown-Modul, Präsentationen, der Tagesgruß-Timing-Fehler.

September 2026

Breite und Tiefe

Badge, der Countdowner-Totalausfall und seine Lehren, Besucherzähler, Mitgliedsantrag und Fragebogen, REÜR, Redirect-Plugin, Fotoquiz, Fotowettbewerb-Admin, Kirchenbuch, Offline-Kopie – und am 29.09. Fotowettbewerb und Jury in einer Anwendung.

Was die Zahlen verraten

- **Hohe Versionszahlen sind kein Makel:** Die Regel „bei jeder Änderung die Versionsnummer erhöhen“ macht jeden Zwischenstand nachvollziehbar. Jede der über 130 Template-Versionen ist ein installierbarer, unterscheidbarer Stand.
- **Dokumentation skaliert:** 63 Anleitungs- und FAQ-Dateien in zwei Sprachen und vier Formaten wären von Hand kaum zu pflegen. Ein Generator macht daraus einen Befehl pro Erweiterung.
- **Qualitätssicherung wurde zur Routine:** 20 Audit-Berichte zeigen, dass Prüfen ein fester Arbeitsschritt geworden ist – nicht erst, wenn etwas brennt.

KAPITEL 3

Die Projektlandkarte

> *Zwei Welten, ein Werkzeug: Joomla-Erweiterungen für den eigenen Vertrieb auf der einen Seite, eigenständige Anwendungen für Verein, Büro und Freizeit auf der anderen.*

Joomla-Erweiterungen (JoomION)

Den größten Block bilden Erweiterungen für Joomla 5 und 6 – von kleinen Modulen bis zu Komponenten mit eigenem Datenmodell, Dashboard, Hilfeseite, Update-Feed und Dokumentation in zwei Sprachen.

Projekt	Art	Technik	Stand	Status
JoomION Template + Framework-Builder	Template + System-Plugin	Joomla 5/6	1.9.133	laufend gepflegt
JoomION Security Scanner	Paket (Komponente + Task-Plugin)	Joomla 5/6	1.00.93	fertig
JoomION LicManager	Komponente (Verkauf + Lizenzen)	Joomla 6, PayPal REST	1.0.86	eingefroren
JoomION Erweiterungs-Downloader	Komponente (Fork des LicManagers)	Joomla 5/6	1.0.29+	fertig
JoomION Updater	Komponente	Joomla 5/6	1.2.7	fertig
JoomION Katalog	Komponente + JSON-Feed	Joomla 6	1.0.1	fertig
JoomION Projekte	Paket (Projekte, Zeiten, Rechnungen)	Joomla 5/6	Phase 1–3	Phase 4 offen
JoomION Pager	Komponente (OnePager-Builder)	Joomla 5/6	Phase 1	fertig
JoomION Gallery	Paket (Galerie + Lightbox)	Joomla 5/6	1.0.x	in Arbeit
JoomION Widerruf	Paket (Modul + System-Plugin)	Joomla 5/6	1.0.28	fertig, Audit bestanden
JoomION Admin-URL	System-Plugin	Joomla 5/6	1.0.22	fertig, Audit bestanden
JoomION Schema	System-Plugin (JSON-LD, OpenGraph)	Joomla 5/6	–	fertig
JoomION Redirect	System-Plugin (301 auf kanonische URL)	Joomla 5/6	1.0.0	fertig
JoomION Badge	System-Plugin (Schwebe-Badges)	Joomla 5/6	–	fertig
JoomION Besucherzähler	Paket (2 Module)	Joomla 5/6	1.4.2	fertig
JoomION Tagesgruß	Modul	Joomla 5/6	2.2.28	fertig

Projekt	Art	Technik	Stand	Status
JoomION Countdowner	Modul	Joomla 5/6	1.6.2	fertig
JoomION ScrollBox	Modul	Joomla 5/6	1.0.0	fertig
JoomION SecCheck	Task-Plugin	Joomla 5/6	1.0.4	fertig
Bilder- und Dateienupload	Pakete	Joomla 5/6	5.0.x / 6.1.21	fertig
FlipMe	Modul + Komponente (Blätterbuch)	Joomla 5/6	2.1.15	fertig
JoomION Fleet	Fernwartung (Server + Client)	Joomla 5/6	Plan	geplant
Mitgliederverwaltung	Paket (Konzept)	Joomla 6	Konzept	wartet auf Entscheidungen

Eigenständige Anwendungen

Parallel entstanden Werkzeuge ohne Joomla: schlanke PHP-Anwendungen für den Fotoverein, Büro-Software und einige Einzelstücke.

Projekt	Art	Technik	Stand	Status
Fotowettbewerb mit integrierter Jury	Web-App	PHP, MySQL	Variante C	im Servertest
Bilderabstimmung	Web-App (Jury-Abstimmung)	PHP, JSON-Dateien	–	fertig
Fotoquiz	Web-App (200 Fragen, Ranglisten)	PHP, signierte Tokens	1.0.9	fertig
Mitgliedsantrag-Popup	Formular-Wizard	PHP, Bootstrap 5	–	fertig
Fragebogen-Popup	Selbsttest	HTML, JavaScript	–	fertig
Arbeitsnachweis / Baustellen	Web-App	PHP, MySQL, FPDF	–	live im Einsatz
REÜR	Rechnungen + EÜR	PHP, MySQL, tFPDF	1.0	fertig
Rechnungstool	Arbeitsmappe mit Makros	Excel, VBA	1.0	fertig
Rechnungsauslese	Browser-Werkzeug mit OCR	pdf.js, Tesseract.js	–	fertig
Dalli-Klick	Partyspiel	PowerPoint, VBA	–	privat
Präsentationen	Foliensätze	python-pptx, LibreOffice	–	fertig
Kirchenbuch 1864/65	Bildentzerrung + Transkription	Python, Bildverarbeitung	–	fertig
Offline-Kopie einer Vereinswebsite	Statischer Spiegel	HTML, JavaScript	–	repariert

Ein Muster zieht sich durch

Fast jedes eigenständige Projekt übernahm Bausteine eines Vorgängers: der Fragebogen den Aufbau des Mitgliedsantrags, das Fotoquiz beide, REÜR den Berichts-Editor des Arbeitsnachweises. Wiederverwendung war der größte Beschleuniger – und, wie Teil II zeigt, gelegentlich auch der Weg, auf dem alte Fehler in neue Projekte wanderten.

TEIL II

Die Zusammenarbeit

4 · Was Claude Code wirklich gut kann
5 · Wo es danebenging: zwölf Fehlermuster

KAPITEL 4

Was Claude Code wirklich gut kann

> *Einige Aufgaben hat Claude Code nicht nur beschleunigt, sondern überhaupt erst wirtschaftlich möglich gemacht. Sieben Beobachtungen aus der Praxis.*

1 – Komplette, installierbare Pakete

Aus einer Beschreibung entstand regelmäßig ein vollständiges Joomla-Paket: Manifest, Service-Provider, Model-View-Controller, Sprachdateien in zwei Sprachen, SQL-Installation, Install-Skript, Build-Skript und Update-Feed. Das JoomION-Redirect-Plugin etwa wurde in einer Sitzung konzipiert, gebaut, auditiert und dokumentiert – einschließlich eines selbst gefundenen Fehlers, der vor dem ersten Build behoben war.

2 – Dokumentation in Serie

Ein eigener Generator erzeugt aus einer Inhaltsdatei je Erweiterung Anleitung und FAQ als HTML, Markdown, PDF und Word – jeweils auf Deutsch und Englisch. Was von Hand Tage dauern würde, ist ein Befehl.

3 – Gründliche Prüfberichte

Für jede Erweiterung entsteht auf Wunsch ein PSR-12- und Sicherheitsaudit als PDF: Joomla-5/6-Konformität, SQL-Injection, CSRF, Zugriffskontrolle, Upload-Prüfungen, Schadcode-Muster. Die Befunde sind nach Schwere sortiert und mit Datei und Zeile belegt.

4 – Belegen statt behaupten

Die besten Diagnosen entstanden, wenn Claude Code im Joomla-Quellcode nachsah, statt zu vermuten: wie der Installer den Skript-Klassennamen bildet, wie die Namespace-Map aufgebaut wird, warum die Datenbank-Seite eine leere Version meldet. Diese Erkenntnisse sind heute Teil der Fallstricke in Teil III.

5 – Diagnose-Werkzeuge bauen

Beim Upload-Fehler, der nur große Dateien traf, war im Log nichts zu sehen. Statt weiter zu raten, baute Claude Code eine kleine clientseitige Protokollierung ein. Der erste echte Fehlversuch lieferte den HTTP-Status 403 und die Blockseite einer Sicherheitserweiterung – Ursache gefunden, kein Fehler im eigenen Paket.

6 – Mathematik und Fachlogik

Für die Bilderabstimmung wurde eine faire, von der Bildanzahl abhängige Rundenformel hergeleitet und sofort umgesetzt: Die Zahl der voraussichtlichen Runden folgt aus `r = ⌈ln(N / Schwelle) / ln(1 / Quote)⌉`. Ein Quotenrechner empfiehlt seitdem Behaltequote und Rundenzahl – und reproduzierte die Rechenbeispiele des Nutzers exakt.

7 – Weit über den eigenen Tellerrand

Excel-Arbeitsmappen mit VBA per COM-Automatisierung, ein OCR-Werkzeug für gescannte Rechnungen, PowerPoint-Makros für ein Partyspiel, entzerrte und transkribierte Kirchenbuchseiten, die Reparatur einer defekten Offline-Kopie mit 523 nachgeladenen Fotos: Die Bandbreite der erledigten Aufgaben war größer als die jedes einzelnen Fachgebiets.

> **Die gemeinsame Zutat**
> In allen sieben Fällen war das Ziel klar umrissen, und das Ergebnis ließ sich prüfen – durch Installation, Test, Rechenbeispiel oder Log. Wo eine dieser Zutaten fehlte, wurde es schwierig.

KAPITEL 5

Wo es danebenging: zwölf Fehlermuster

> *Die lehrreichsten Stunden waren die, in denen etwas schiefging. Fast alle Fehler fallen in wiederkehrende Muster – und fast alle lassen sich mit einer einfachen Regel verhindern.*

Die folgenden Muster sind nach Häufigkeit und Folgen ausgewählt. Zu jedem gehört eine Regel, die seitdem fest im Gedächtnis von Claude Code steht (Kapitel 16).

1 – Raten statt fragen

Nach einem Update fiel eine komplette Website aus. Claude Code lieferte zwei plausible Diagnosen und zwei Korrekturen – beide behoben echte Nebenprobleme, keine die Ursache. Erst die exakte PHP-Fehlermeldung zeigte: Beim Neuschreiben einer Datei war eine Methode verloren gegangen.

Regel

Spätestens nach dem ersten erfolglosen Versuch die exakte Fehlermeldung anfordern – Stacktrace, Server-Log, Browser-Konsole.

2 – Mit der falschen Quelle arbeiten

Ein Bugfix für fünf Erweiterungen wurde gegen lokale Arbeitsordner geschrieben, die deutlich älter waren als die ausgelieferten Pakete. Das Ergebnis sah plausibel aus – mit höherer Versionsnummer, aber auf veraltetem Code.

ANWEISUNG AUS DER PRAXIS

Nimm diese, die ich dir gebe, und keine alten.

3 – Den Auftrag ausweiten

Eine neue Option „Fett & Kursiv“ sollte nur in bestimmten Boxen erscheinen. Claude Code baute sie in alle fünfzehn Schriftstil-Auswahlen ein, auch in die Überschriften. Die Arbeitskopie war nach dem Teilrückbau so unklar, dass aus dem letzten sauberen Paket neu begonnen werden musste.

4 – Ungefragt mit erledigen

Beim Aktualisieren der Dokumentation einer Erweiterung wurden über ein Sammelskript die Dokumente dreier anderer Erweiterungen mit neu erzeugt – mit neuen Zeitstempeln, ohne Auftrag.

5 – Fehlerhafte Vorlagen kopieren

Ein bekannter und längst dokumentierter Joomla-6-Fehler (Ausgabe per `echo` im Install-Skript) tauchte wieder auf, weil ein neues Modul sein Skript aus einer älteren, ungeprüften Vorlage übernahm. Die Folge war eine Kette bis zum Totalausfall.

6 – Audit-Empfehlungen blind umsetzen

Ein Audit empfahl, CSS über den WebAssetManager statt inline zu laden. Im konkreten Fall wurde das Modul aber erst nach dem Rendern des Seitenkopfs erzeugt – der Button funktionierte danach gar nicht mehr. Die Empfehlung war richtig, nur nicht hier.

7 – Ganze Dateien neu schreiben

Beim vollständigen Neuschreiben einer Klasse ging eine aufgerufene Methode verloren. `php -l` meldete „keine Syntaxfehler“ – die Website lieferte trotzdem bei jedem Aufruf einen Fatal Error.

8 – Pflichtschritte vergessen

Acht Versionen des Security Scanners wurden ohne aktualisierten Update-Feed gebaut. Beim Fotowettbewerb wurde eine lokale Konfiguration mit Testzugangsdaten mitgeliefert, die nach einer späteren Testrunde erneut entstanden war.

9 – Funktionierendes zerstören

Für eine Live-Vorschau auf der Optionsseite des Templates entstand ein zusätzliches Feld, das das bestehende Layout umbaute. Über sechs Versionen verschlechterte sich die Seite, bis von der letzten guten Version neu begonnen wurde.

10 – Vorauswahl statt Kontrolle

Für die Größe eines Badges auf Mobilgeräten lieferte Claude Code vier Voreinstellungen („Automatisch“, „Leicht“, „Stark“, „Aus“). Gewünscht waren je Gerätegröße eigene Prozentwerte und eigene Positionen – der Umbau kostete eine Version mit geänderten Feldnamen.

11 – Am falschen Ort ablegen

Pakete landeten versehentlich im Ordner für veröffentlichte Versionen, ein neues Projekt außerhalb des vereinbarten Arbeitsbereichs, ein Audit als Chat-Anzeige statt als PDF im Audit-Ordner. Jedes Mal eine Kleinigkeit – aber jedes Mal Aufräumarbeit.

12 – „Sieht plausibel aus“ für „funktioniert“ halten

Vier Anläufe brauchte der Erweiterungs-Downloader, bis er im richtigen Ordner installiert wurde. Jede Version sah für sich korrekt aus; die Ursache lag in einer Joomla-Eigenheit, die erst ein Blick in den Quellcode offenlegte.

> **Überzeugend heißt nicht richtig**
> Keiner dieser Fehler war auf den ersten Blick erkennbar. Die Antworten klangen jedes Mal sicher. Die Gegenmittel sind unspektakulär: Belege verlangen, exakt eingrenzen, nach jeder Änderung prüfen – und das, was sich bewährt, als Regel festschreiben.

TEIL III

Joomla-Praxis: Fallstricke

6 · Paketbau & Manifest
7 · Installer, Updates & Versionen
8 · Formulare, Felder & Backend
9 · JavaScript, CSS & Assets
10 · Laufzeit, Sitzungen & Sicherheit
11 · JED-Checker, Audits & Auslieferung

KAPITEL 6

Paketbau & Manifest

> *Die meisten Installationsfehler entstehen, bevor eine einzige Zeile Programmlogik läuft: im Manifest, in der Ordnerstruktur und im ZIP selbst.*

Joomla leitet aus dem Manifest weit mehr ab, als man vermutet: den Ordnernamen, den Element-Namen, den erwarteten Dateinamen des Manifests, den Namespace und die Menü-Links. Viele dieser Ableitungen passieren still – ein Fehler zeigt sich erst beim erneuten Einlesen, beim Update oder nie.

So lesen Sie die Fallstricke

Jede Karte nennt das beobachtbare **Symptom**, die tatsächliche **Ursache** und die bewährte **Lösung**. Die Nummern gelten im ganzen Buch und im Katalog (Anhang A).

F-01 ZIP mit Backslash-Pfaden

SYMPTOM	Joomla meldet „Manifest-Datei konnte nicht gefunden werden“, obwohl die XML-Datei korrekt im ZIP-Wurzelverzeichnis liegt.
URSACHE	PowerShell `Compress-Archive` speichert Einträge wie `language\de-DE\datei.ini` mit Backslash. Der ZIP-Standard verlangt den Schrägstrich; der PHP-Installer erkennt die Ordnerstruktur nicht.
LÖSUNG	ZIP über `System.IO.Compression.ZipArchive` bauen und jeden Eintragsnamen auf `/` normalisieren. Nach dem Build die Eintragsnamen stichprobenartig auslesen.

Aufgetreten in: mod_countdown 1.0.0, Scanner-Build

F-02 CRLF statt LF in allen Paketdateien

SYMPTOM	Das Prüfwerkzeug meldet „Incorrect end-of-line character found“ für Sprach- und PHP-Dateien.
URSACHE	Unter Windows schreiben sowohl Editor-Werkzeuge als auch Python im Textmodus `\r\n`. Joomla läuft damit, der Coding-Standard verlangt aber LF.
LÖSUNG	Beim Schreiben `newline='\n'` erzwingen und vor dem Zippen alle Textdateien normalisieren; danach per `grep -c $'\r'` gegenprüfen.

Aufgetreten in: Admin-URL-Plugin 1.0.15

F-03 Plugin-Manifest mit falschem Dateinamen

SYMPTOM	Installation bricht ab: „Die Datei <element>.xml konnte nicht gefunden werden“.
URSACHE	Das Manifest hieß `plg_task_joomionseccheck.xml`. Joomla leitet den erwarteten Namen aus dem Element ab und sucht `joomionseccheck.xml`.
LÖSUNG	Plugin-Manifest exakt wie das Element benennen. Das Präfix `plg_<gruppe>_` gehört nur an Sprachdateien und den ZIP-Namen.

Aufgetreten in: SecCheck-Task-Plugin 1.0.1 → 1.0.3

F-04 Leeres „element“ beim Plugin-Install

SYMPTOM	Datenbankfehler „Field 'element' doesn't have a default value“.
URSACHE	Bei einem namespaced Plugin ohne klassische Haupt-PHP-Datei fehlte das Attribut, aus dem Joomla den Element-Namen liest.
LÖSUNG	Im Manifest `<folder plugin="name">services</folder>` setzen.

Aufgetreten in: JoomION Pager 1.1.1

F-05 Komponente landet im falschen Ordner

SYMPTOM	„Class … not found“, im Menü erscheint der rohe Sprachschlüssel.
URSACHE	Joomla bildet das Element aus dem `<name>`-Tag, nicht aus dem Dateinamen. Aus „JoomION Erweiterungs Downloader“ wurde `com_joomionerweiterungsdownloader`.
LÖSUNG	`<name>` immer als Sprachschlüssel in Element-Form setzen (`COM_JOOMIONDOWNLOADER`); den Anzeigenamen liefert die `.sys.ini`.

Aufgetreten in: Erweiterungs-Downloader – vier Fehlversuche (1.0.0–1.0.3)

F-06 Namespace mit Administrator-Suffix

SYMPTOM	Autoload schlägt fehl oder funktioniert nur, weil `provider.php` Namespaces zur Laufzeit nachregistriert.
URSACHE	Joomla hängt an den Manifest-Namespace selbst `\Administrator\` bzw. `\Site\` an. Steht das Suffix schon im Manifest, entstehen doppelte Präfixe.
LÖSUNG	Im Manifest nur die Basisform (`Joomla\Component\Bilderupload`), dann den Workaround entfernen.

Aufgetreten in: Audit von com_bilderupload

F-07 packagename passt nicht zum Dateinamen

SYMPTOM	Die Versionsspalte wird nach „Cache aktualisieren“ leer, der native Updater zeigt nie ein Update – obwohl Feed und ZIP korrekt sind.
URSACHE	Joomla sucht bei jedem erneuten Lesen `pkg_<packagename>.xml`. `joomionwiderruf` passte nicht zu `pkg_joomion_widerruf.xml`.
LÖSUNG	`<packagename>` exakt passend zum Dateinamen setzen; verwaiste Paket-Zeile danach aufräumen.

Aufgetreten in: JoomION Widerruf (bis 1.0.14)

F-08 Leerer media-Knoten

SYMPTOM	JED-Checker: „The node extension/media is empty“ – obwohl die Installation funktioniert.
URSACHE	Ein selbstschließendes `<media … />` ohne Kind-Elemente ist gültiges XML, wird vom Prüfwerkzeug aber abgelehnt.
LÖSUNG	Explizite `<folder>css</folder>` / `<folder>js</folder>`-Kinder eintragen.

Aufgetreten in: JoomION Widerruf

F-09 Komponenten-Optionen öffnen nicht

SYMPTOM	„Form::loadForm could not load file“ beim Klick auf „Optionen“.
URSACHE	Die Optionsfelder standen nur im `<config>`-Block des Manifests.
LÖSUNG	Eigene `admin/config.xml` anlegen und im `<files>`-Abschnitt als `<filename>` registrieren.

Aufgetreten in: Security Scanner 1.00.26

F-10 SQL-Ordner wird nicht gefunden

SYMPTOM	Installationsfehler „administrator/.../sql existiert nicht“.
URSACHE	Der `sql/`-Ordner lag im ZIP-Wurzelverzeichnis. Der Pfad im Manifest ist relativ zum Admin-Komponentenordner.
LÖSUNG	`sql/` in den Admin-Ordner legen und dort in `<files>` auflisten.

Aufgetreten in: JoomION Katalog

F-11 Jeder Menüklick landet auf dem Dashboard

SYMPTOM	Komponente installiert und aktiv, aber jeder Klick im Admin-Menü springt still auf das Joomla-Dashboard.
URSACHE	Der Manifest-Link enthielt `index.php?`. Joomla stellt das selbst voran – es entsteht `index.php?index.php?option=…`, das echte `option` fehlt.
LÖSUNG	Menü-Links nur als Query schreiben: `option=com_x&view=y`. Diagnose-Trick: die Adresszeile ansehen.

Aufgetreten in: Security Scanner

F-12 „Version ()“ auf der Datenbank-Wartungsseite

SYMPTOM	System → Wartung → Datenbank meldet eine nicht passende Update-Version mit leerer Klammer.
URSACHE	Das Komponenten-Manifest hieß `com_x.xml`; Joomla erwartet es ohne Präfix als `x.xml`. Zweite Ursache: höchste SQL-Update-Datei ≠ Manifest-Version.
LÖSUNG	Manifest umbenennen, die alte Datei im `postflight` löschen und den `#__schemas`-Eintrag dort selbst setzen.

Aufgetreten in: Updater 1.2.7, LicManager 1.0.86

Die Build-Kette in drei Zeilen

Dateien mit LF normalisieren → ZIP mit Schrägstrich-Pfaden bauen → Eintragsnamen auslesen und prüfen. Erst dann gilt ein Paket als fertig.

KAPITEL 7

Installer, Updates & Versionen

> *Ein Paket, das sich installieren lässt, ist erst die halbe Miete. Die andere Hälfte: sauber aktualisieren, die richtige Version anzeigen und Updates zuverlässig anbieten.*

Der Wechsel von Joomla 5 auf 6 hat hier die meisten Überraschungen gebracht: Der Installer antwortet mit JSON, alte Dateisystem-Klassen sind verschwunden, und der native Updater ist strenger beim Zuordnen von Updates.

Aufbau eines freien Update-Feeds

Jede frei verteilte Erweiterung bekommt eine statische XML-Datei auf dem Update-Server – mit zwei Blöcken für Joomla 5 und 6 und drei Prüfsummen:

```
<updates>
  <update>
    <name>JoomION Redirect</name>
    <element>joomionredirect</element>
    <type>plugin</type>
    <folder>system</folder>
    <client>site</client>
    <version>1.0.0</version>
    <downloads>
      <downloadurl type="full"
format="zip">https://…/plg_system_joomionredirect_v1.0.0.zip</downloadurl>
    </downloads>
    <sha256>…</sha256><sha384>…</sha384><sha512>…</sha512>
    <targetplatform name="joomla" version="6.[0-9]+"/>
    <php_minimum>8.1.0</php_minimum>
  </update>
  <!-- zweiter Block identisch mit version="5.[0-9]+" -->
</updates>
```

F-13 „Unexpected token '<'“ bei der Installation unter Joomla 6

SYMPTOM	Die Installation bricht im Browser ab; unter Joomla 5 lief dasselbe Paket.
URSACHE	Joomla 6 installiert über eine JSON-Antwort. Ein `echo` im Install-Skript landet mitten im JSON.
LÖSUNG	Ausgabe puffern (`ob_start()`) und per `enqueueMessage()` anzeigen. Jedes kopierte Install-Skript vor der Auslieferung auf `echo` prüfen.

Aufgetreten in: Template, Scanner – und als Rückfall im Countdowner 1.6.0

F-14 Hinweise der Unter-Erweiterungen fehlen

SYMPTOM	Aktivierungs-Buttons aus dem Template-Skript erscheinen nach der Paketinstallation nicht.
URSACHE	Bei Paketen zeigt Joomla nur Ausgaben des Paket-eigenen Skripts.
LÖSUNG	Hinweise in das `script.php` des Pakets verlagern.

Aufgetreten in: Template-Paket 1.9.77

F-15 Rote Leiste ohne Text

SYMPTOM	Die Installation endet mit einer leeren roten Fehlermeldung.
URSACHE	Das Install-Skript nutzte `Joomla\CMS\Filesystem\Folder` / `File`, die es in Joomla 6 nicht mehr gibt – Fatal Error im `postflight`.
LÖSUNG	Natives PHP (`mkdir`, `file_put_contents`) verwenden und das ganze Skript in `try/catch` kapseln.

Aufgetreten in: LicManager 1.0.0 → 1.0.1

F-16 Install-Skriptklasse wird nicht gefunden

SYMPTOM	Das Skript läuft schlicht nicht – kein Fehler, keine Wirkung.
URSACHE	Joomla bildet den Klassennamen je Erweiterungstyp anders; bei mehrteiligen Elementen müssen Unterstriche erhalten bleiben (`Mod_Paypal_DonationInstallerScript`).
LÖSUNG	Verifizierte Namensschemata nutzen oder das moderne Muster mit anonymer Klasse und `InstallerScriptInterface`.

Aufgetreten in: PayPal-Spendenmodul

F-17 Update wird nicht angezeigt – fehlendes <client>

SYMPTOM	Feed erreichbar, Version und Hashes stimmen, der Katalog zeigt die Erweiterung – aber Joomla meldet kein Update.
URSACHE	Joomla ordnet Updates auch über die `client_id` zu. Ohne `<client>` im Update-XML bleibt sie leer; ein Paket fiel im Generator in den Standardzweig.
LÖSUNG	`<client>` je Typ setzen (Komponente: `administrator`, sonst `site`). Nach einer Reparatur „Cache leeren“, sonst prüft Joomla stundenlang nicht neu.

Aufgetreten in: Security Scanner 1.00.16

F-18 Update-Site fehlt nach einem Update

SYMPTOM	Keine Update-Meldung, obwohl der Server die neue Version ausliefert.
URSACHE	Joomla legt Update-Sites aus `<updateservers>` zuverlässig nur beim Erst-Install an.
LÖSUNG	Backend: Aktualisierungs-Sites → „Wiederherstellen“. Im Paket: `syncUpdateSite()` im `postflight`.

Aufgetreten in: mehrere JoomION-Erweiterungen

F-19 TUF-Meldung „timestamp expired“

SYMPTOM	Nur eine von mehreren gleichen Joomla-Instanzen meldet einen Fehler beim Kern-Update.
URSACHE	Lokal gecachte, abgelaufene TUF-Metadaten. Die Meldung betrifft ausschließlich das Kern-Update, nie eigene Update-XMLs.
LÖSUNG	Die Tabelle `#__tuf_metadata` keinesfalls leeren – das legte die Update-Seite zweimal komplett lahm. Stattdessen Backup oder den Wert einer gesunden Schwester-Instanz übernehmen.

Aufgetreten in: Kundeninstanz, Juni 2026

F-20 Falsche Versionsnummer in der Oberfläche

SYMPTOM	Das Backend zeigt nach einem Update weiterhin die alte Version.
URSACHE	Die Version war hartcodiert – oder wurde aus `manifest_cache` gelesen, den Joomla nach Updates nicht zuverlässig erneuert.
LÖSUNG	Version immer aus der installierten Manifest-Datei lesen (`simplexml_load_file`), `manifest_cache` nur als Rückfall. Bei der Suche XML-Dateien nicht ausschließen – dort sitzen die `note`-Felder.

Aufgetreten in: Dateienupload, Bilderupload, FlipMe, Admin-URL

F-21 Acht Versionen ohne Update-Feed

SYMPTOM	Nutzer mit der freien Version bekommen keine Updates angeboten.
URSACHE	Der Feed wurde gedanklich als „Nachgang“ behandelt und bei schnellen Feature-Iterationen vergessen.
LÖSUNG	Feed, Download-URL und alle drei Hashes (SHA-256/384/512) gehören in denselben Arbeitsschritt wie der Build – und werden per Skript gegen die Datei geprüft.

Aufgetreten in: Security Scanner 1.00.67–1.00.74

Nach jeder Reparatur: Cache leeren

Joomla merkt sich den Zeitpunkt der letzten Prüfung auch dann, wenn sie fehlgeschlagen ist. Wer nur „Aktualisierungen suchen“ klickt, wartet unter Umständen Stunden auf das Ergebnis.

KAPITEL 8

Formulare, Felder & Backend

> *Joomlas Formular-System ist mächtig – und verzeiht wenig. Viele Fehler zeigen sich nicht als Meldung, sondern als Wert, der still nicht gespeichert wird.*

Besonders tückisch sind Fehler ohne Fehlermeldung: ein Häkchen, das nach dem Speichern wieder fehlt; ein eigenes Feld, das als leeres Textfeld erscheint; ein Button, der schlicht nichts tut. Hier hilft nur, den Weg eines Werts vom Formular bis in die Datenbank Schritt für Schritt zu verfolgen.

PROMPT AN CLAUDE CODE

Das Häkchen „Joomla 6“ wird nach dem Speichern nicht übernommen. Verfolge den Wert vom gerenderten <input> über den POST-Request, Form::filter() und validate() bis zum Model::save(). Zeige mir, an welcher Stelle er verloren geht, bevor du etwas änderst.

F-22 Falsche Basisklasse für eigene Felder

SYMPTOM	`Class "Joomla\CMS\Form\Field\FormField" not found` beim Öffnen der Optionen.
URSACHE	Die abstrakte Basisklasse liegt in `Joomla\CMS\Form\FormField`; im Unter-Namespace `Field` stehen nur konkrete Typen.
LÖSUNG	Richtig importieren – oder von einem konkreten Typ wie `TextField` ableiten.

Aufgetreten in: LicManager 1.0.52

F-23 Eigenes Feld erscheint als leeres Textfeld

SYMPTOM	Balken und Hilfe-Tab fehlen, stattdessen ein unbeschriftetes Eingabefeld. Cache leeren hilft nicht.
URSACHE	`addfieldprefix` mit namespaced Feldklassen ist in Modulen und Plugins unzuverlässig; Joomla fällt still auf `text` zurück.
LÖSUNG	`addfieldpath` plus klassische `JFormField<Typ>`-Klasse ohne eigenen Namespace.

Aufgetreten in: JoomlON Besucherzähler

F-24 Listenauswahl wird nicht gespeichert

SYMPTOM	Nach dem Speichern springt ein Auswahlfeld auf den alten Wert zurück.
URSACHE	Die Option „Standard“ trug `value=""`; ein leerer Wert wird in manchen Fällen wie „nicht gesetzt“ behandelt.
LÖSUNG	Einen echten Platzhalterwert wie `default` verwenden – in Option und `default`-Attribut.

Aufgetreten in: JoomlON Besucherzähler 1.3.1 (vorbeugend behoben)

F-25 Häkchen werden nicht gespeichert

SYMPTOM	Der POST-Wert kommt korrekt an, landet aber nie in der Datenbank.
URSACHE	Ein eigenes Feld gab mehrere Eingaben unter fremden Namen aus. `Form::filter()` verwirft alles, was nicht im XML deklariert ist.
LÖSUNG	Für jeden fremden Namen ein `<field type="hidden">` vor dem eigenen Feld deklarieren.

Aufgetreten in: Erweiterungs-Downloader 1.0.29

F-26 Editor speichert fremden Inhalt

SYMPTOM	Im Frontend erscheinen Feldbeschriftungen als Sektionsinhalt.
URSACHE	Ein TinyMCE-Editor wurde in einem beim Laden versteckten Container (`showon`, eingeklapptes `<details>`) initialisiert und band sich an den falschen DOM.
LÖSUNG	Editor-Felder immer sichtbar rendern; kein `showon` auf Editor-Feldern.

Aufgetreten in: JoomION Pager 1.0.10

F-27 Auswahlfelder in wiederholbaren Subforms

SYMPTOM	`Uncaught Error: Misconfiguration` beim Verschieben einer Zeile; Beitragsauswahl in neu hinzugefügten Zeilen reagiert nicht.
URSACHE	Die Web-Components von `media` - und Modal-Feldern vertragen das Klonen und Neu-Einhängen der Zeilen nicht.
LÖSUNG	In sortierbaren Subforms `type="text"` bzw. ein `type="sql"` -Dropdown verwenden.

Aufgetreten in: JoomION Pager, JoomION Badge

F-28 Speichern scheitert am veralteten Filter

SYMPTOM	„calls a deprecated filter class JComponentHelper“ beim Speichern eines Editor-Felds.
URSACHE	Der Filter stammt aus Joomla 3 und ist nicht namespaced.
LÖSUNG	`filter="\Joomla\CMS\Component\ComponentHelper::filterText"` .

Aufgetreten in: JoomION Widerruf, JoomION Projekte

F-29 Titel da, Toolbar-Buttons fehlen

SYMPTOM	Die Seite zeigt den Titel, aber kein „Neu“ und kein „Speichern“.
URSACHE	`createToolbar()` erzeugt jedes Mal eine neue, leere Instanz; das Admin-Template rendert eine andere.
LÖSUNG	`Toolbar::getInstance('toolbar')` verwenden.

Aufgetreten in: JoomION Pager 1.0.3

F-30 „Direkter Aufruf dieser Seite nicht gestattet“

SYMPTOM	Ein Link auf ein Bearbeitungsformular endet mit Fehler #<id>.
URSACHE	Joomla verlangt, dass die ID über den Edit-Task in die Bearbeitungssitzung aufgenommen wird.
LÖSUNG	Über `task=<controller>.edit&id=…` verlinken, nicht über `view=…&layout=edit`.

Aufgetreten in: Template-Paket 1.9.78 (fünf Stellen)

F-31 Plugin bleibt nach „Zurück“ gesperrt

SYMPTOM	Nach eigenen „Speichern & zurück“-Buttons zeigt das Plugin ein Schloss.
URSACHE	Nur Joomlas eigene Buttons checken den Datensatz wieder ein.
LÖSUNG	Vor der Weiterleitung selbst einchecken (`plugins.checkin` mit Token).

Aufgetreten in: Template 1.9.83, Scanner 1.00.49

F-32 Button „Speichern & zurück“ tut nichts

SYMPTOM	Klick ohne jede Reaktion, keine Konsolenfehler.
URSACHE	Zwei Fehler zugleich: `getElementById('adminForm')` liefert `null` (die ID variiert je View), und eine HTML-escapte URL mit `&` wurde als JavaScript-String benutzt.
LÖSUNG	`document.adminForm` verwenden; URLs für JavaScript per `json_encode()` ausgeben.

Aufgetreten in: JoomION Widerruf

F-33 „500 searchtools::default not found“

SYMPTOM	Listenansichten brechen unter Joomla 6 ab.
URSACHE	Der HTMLHelper `searchtools.default` aus Joomla 3 wurde entfernt.
LÖSUNG	`LayoutHelper::render('joomla.searchtools.default', ['view' => $this])`.

Aufgetreten in: Security Scanner

F-34 „Incorrect datetime value: ''“

SYMPTOM	Speichern bricht ab, sobald ein Datums- oder Zahlenfeld leer bleibt.
URSACHE	Leere Formularwerte kommen als Leerstring an; DATE- und DECIMAL-Spalten akzeptieren das nicht.
LÖSUNG	In `Table::check()` leere Datumsfelder auf `null`, Zahlen auf `0` setzen – bei jeder neuen Tabelle gleich mit.

Aufgetreten in: JoomION Projekte 1.00.35

F-35 „Ungültige Übersetzungs-Zeichenkette“

SYMPTOM	Die Sprachprüfung meldet eine defekte Zeile.
URSACHE	Ein gerades Anführungszeichen `"` im Wert einer INI-Datei beendet den String vorzeitig – etwa bei „Zufällig generieren" mit falschem Schlusszeichen.
LÖSUNG	Typografische Anführungszeichen verwenden und je Zeile die Anzahl gerader `"` prüfen. Automatische Korrekturen danach verifizieren – ein Skript hatte auch das abschließende Zeichen ersetzt.

Aufgetreten in: Admin-URL-Plugin, JoomION Widerruf

KAPITEL 9

JavaScript, CSS & Assets

> *Im Browser entscheidet die Reihenfolge: wann ein Skript läuft, welche Regel gewinnt, welche Datei der Cache liefert. Viele dieser Fehler findet nur ein echter Browsertest.*

Ein wiederkehrendes Motiv dieses Kapitels: Die Fehler waren mit Textauswertung, `curl` oder synthetischen Ereignissen nicht zu finden. Erst der Test im echten Browser – mit einem Zustandswechsel wenige Sekunden in der Zukunft – oder auf einem echten Tablet brachte sie ans Licht.

```
// Ready-Wrapper für jedes per addInlineScript() registrierte Skript
(function (fn) {
  if (document.readyState === 'loading') {
    document.addEventListener('DOMContentLoaded', fn);
  } else {
    fn();
  }
})(function () {
  /* ursprünglicher Code mit getElementById(...) */
});
```

F-36 Buttons reagieren auf nichts

SYMPTOM	Kunde meldet: Speichern funktioniert nicht mehr. Kein Konsolenfehler, Verdacht auf `.htaccess`.
URSACHE	`addInlineScript()` gibt Skripte immer gesammelt im `<head>` aus – vor dem eigenen HTML. `getElementById()` findet beim Start nichts, `if (!el) return;` schluckt es.
LÖSUNG	Jedes so registrierte Skript mit sofortigem DOM-Zugriff in einen `DOMContentLoaded`-Wrapper packen (zentrale Hilfsmethode).

Aufgetreten in: JoomION Tagesgruß 2.2.27 – sieben Stellen in sechs Dateien, seit 2.2.25 defekt

F-37 Modul funktionslos nach einem Audit-Fix

SYMPTOM	Nach der Umstellung auf den WebAssetManager öffnet der Button das Formular nicht mehr.
URSACHE	Das Modul wird von einem System-Plugin nachträglich in `onAfterRender()` gerendert – der `<head>` ist da längst fertig. Der Fehler wurde von einem `catch` verschluckt.
LÖSUNG	Im Ad-hoc-Render-Kontext CSS inline ausgeben. Audit-Empfehlungen immer gegen den tatsächlichen Render-Kontext prüfen.

Aufgetreten in: JoomION Widerruf

F-38 CSS-Korrektur wirkt scheinbar nicht

SYMPTOM	Nach einem Update „funktioniert nichts“ – ein riesiges, ungestyltes Logo, Farben unverändert.
URSACHE	CSS- und JS-Dateien ohne Versionsparameter; der Browser liefert die alte Datei aus dem Cache.
LÖSUNG	Von Anfang an `?v=<filemtime>` an jede Datei hängen. Bei „geht nicht“ nach CSS-Änderungen zuerst Strg+F5 testen.

Aufgetreten in: Template bis 1.7.67, Arbeitsnachweis – zweimal

F-39 hidden-Element bleibt sichtbar

SYMPTOM	Der Countdown friert ein, gleichzeitig erscheint der Ablauftext.
URSACHE	Eine eigene Regel wie `display: flex` schlägt den Browser-Standard `[hidden]{display:none}` – unabhängig von der Spezifität.
LÖSUNG	Immer eine Regel `.element[hidden]{display:none}` danebensetzen. Zustandswechsel mit kurzer Testzeit live im Browser beobachten.

Aufgetreten in: Countdown-Modul 1.3.0 – seit 1.0.0 unentdeckt

F-40 Touch-Drag scrollt die Seite

SYMPTOM	Das Badge lässt sich horizontal ziehen, vertikal scrollt die Seite oder das Smartphone lädt neu.
URSACHE	Scrollen, Rubber-Banding und Pull-to-Refresh laufen über drei getrennte Mechanismen; unter iOS kommen Eigenheiten der Pointer Events und von `position:fixed` dazu.
LÖSUNG	Sechs Bausteine kombinieren (touch-action, `preventDefault` auf touchstart, overscroll-behavior, touchmove auf document, abgesichertes `setPointerCapture`, Body-Lock) – oder für Touch direkt auf Touch Events umsteigen.

Aufgetreten in: JoomION Badge – mehrere Versionen auf echten Geräten

F-41 Rahmen nur an geraden Kanten

SYMPTOM	Bei Hexagon, Stern oder Dreieck fehlt der Rahmen an den schrägen Kanten.
URSACHE	Ein CSS-`border` auf einer `clip-path`-Form wird nur entlang des ursprünglichen Rechtecks gezeichnet.
LÖSUNG	Eckige Formen als SVG-Polygon mit `stroke` und `vector-effect="non-scaling-stroke"` zeichnen.

Aufgetreten in: JoomION Badge 1.2.0

F-42 JavaScript erscheint als sichtbarer Text

SYMPTOM	Unter dem Suchfeld der Icon-Auswahl steht Programmcode.
URSACHE	`json_encode()` in einem `oninput`-Attribut kollidierte mit dessen Anführungszeichen.
LÖSUNG	Keine Inline-Handler; ein einmalig registrierter, delegierter Listener auf `document`.

Aufgetreten in: JoomION Badge 1.2.0

F-43 Optionsseite des Templates zerstört

SYMPTOM	Boxen stehen untereinander statt nebeneinander, Vorschauen außerhalb der Karten.
URSACHE	Statt das bestehende Layout-Feld zu erweitern, wurde ein zweites Feld gebaut, das das Layout per DOM-Umbau nachstellte.
LÖSUNG	Neue Vorschau-Funktionen nur additiv im vorhandenen Layout-Feld einbauen. Zurück auf die letzte gute Version (1.9.41) und neu ansetzen.

Aufgetreten in: JoomION Template 1.9.42–1.9.47

Faustregel für Touch-Funktionen

Wenn ein Ziehen per Finger auf iOS nach zwei gezielten Korrekturen immer noch nicht sauber läuft, nicht weiter an Pointer Events feilen, sondern für Touch auf Touch Events umsteigen. Das hat beim Badge mehr gelöst als vier Einzelkorrekturen zuvor.

KAPITEL 10

Laufzeit, Sitzungen & Sicherheit

> *Manche Fehler zeigen sich nur unter echten Bedingungen: mit zwei getrennten Sitzungen, einer Sicherheitserweiterung im Weg oder einem Menüpunkt, den niemand mehr auf dem Schirm hat.*

In diesem Kapitel liegen die Ursachen oft außerhalb des eigenen Codes: in Joomlas Sitzungsverwaltung, im Arbeitsverzeichnis des Servers, in Erweiterungen Dritter oder in der Konfiguration einer Website. Umso wichtiger ist eine saubere Diagnose, bevor der eigene Code „repariert“ wird.

Diagnose zuerst

Zweimal – beim Upload-Fehler und bei der Login-Schleife – lag die Ursache nicht im eigenen Paket. In beiden Fällen führte ein gezielter Test (Diagnose-Logging bzw. eine Standard-Route per `curl`) schneller zum Ziel als jede Codeänderung.

F-44 Das ganze Backend friert ein

SYMPTOM	Die Fortschrittsleiste steht – und kein anderer Backend-Tab reagiert mehr.
URSACHE	Ein AJAX-Request wartete mit `sleep()` auf einen überlasteten externen Dienst und hielt dabei die Sitzungssperre.
LÖSUNG	Wiederholungen clientseitig per `setTimeout`; vor langsamen Aufrufen die Sitzung schließen, Timeouts knapp halten.

Aufgetreten in: Security Scanner 1.00.23 → 1.00.24

F-45 „Ungültiger Token“ nur im Backend-Tab

SYMPTOM	Dieselbe Aktion funktioniert im Frontend, im Backend-Formular nicht.
URSACHE	Backend und Frontend haben getrennte Sitzungen; der im Backend erzeugte CSRF-Token wird vom Frontend-Endpunkt nie akzeptiert.
LÖSUNG	Einen sitzungsunabhängigen, per HMAC mit dem Site-Secret signierten Admin-Token mit Ablaufzeit verwenden und die Berechtigung explizit prüfen.

Aufgetreten in: JoomION Tagesgruß 2.2.28

F-46 Verzeichnis entsteht unter administrator/

SYMPTOM	Das „sichere Verzeichnis“ wird am falschen Ort angelegt.
URSACHE	Bei Backend-Requests ist das Arbeitsverzeichnis `administrator/`; relative Pfade landen dort.
LÖSUNG	Konfigurierte Pfade immer gegen `JPATH_ROOT` auflösen und den tatsächlichen Zielpfad anzeigen.

Aufgetreten in: LicManager

F-47 Falsche Uhrzeit in E-Mails

SYMPTOM	Bestätigungsmails zeigen die Serverzeit statt der Ortszeit.
URSACHE	`Factory::getDate()` liefert UTC – richtig für die Datenbank, falsch für Menschen.
LÖSUNG	Für Anzeigen `Factory::getDate('now', $app->get('offset', 'UTC'))`.

Aufgetreten in: JoomION Widerruf

F-48 „could not be passed by reference“

SYMPTOM	Beim Speichern bricht alles ab, das Frontend bleibt leer.
URSACHE	`insertObject()` nimmt das Objekt per Referenz; ein `(object)[…]`-Literal geht ab PHP 8.1 nicht.
LÖSUNG	Objekt erst einer Variablen zuweisen, dann übergeben.

Aufgetreten in: JoomION Gallery 1.0.5

F-49 Plugin aktiv, aber Events feuern nie

SYMPTOM	Das Task-Plugin taucht im Dialog „Neue Aufgabe“ nicht auf.
URSACHE	`provider.php` übergab dem Konstruktor nur die Konfiguration, nicht den `DispatcherInterface`.
LÖSUNG	Dispatcher als ersten Parameter übergeben; `provider.php` gegen ein funktionierendes Plugin prüfen.

Aufgetreten in: SecCheck-Task-Plugin 1.0.4

F-50 Nicht freigegebene Admins sehen den Builder

SYMPTOM	Trotz „nur ein Benutzer“ sieht ein weiterer Administrator die Werkzeugleiste.
URSACHE	Die Zugriffslogik wurde an zwei Stellen entschieden – im Plugin und im Template – und nur eine war angepasst.
LÖSUNG	Beide Stellen mit identischer Logik führen; Änderungen immer an beiden nachziehen.

Aufgetreten in: JoomION Template 1.9.67

F-51 „Ungültige Serverantwort“ nur bei großen Dateien

SYMPTOM	Uploads kleiner Dateien klappen, große scheitern – und im Joomla-Log steht nichts.
URSACHE	Eine WAF-Funktion einer installierten Sicherheitserweiterung blockierte große Multipart-Uploads mit einer 403-Seite, bevor Joomla die Anfrage sah.
LÖSUNG	Clientseitiges Diagnose-Logging: Bei unlesbarer Antwort HTTP-Status und Antwortanfang per kleiner Zusatzanfrage ins Log schreiben – der erste Treffer zeigte die Ursache.

Aufgetreten in: Dateienupload 6.1.21

F-52 ERR_TOO_MANY_REDIRECTS auf der Login-Seite

SYMPTOM	Nur manche Besucher, scheinbar nur ohne „www“, landen in einer Endlosschleife.
URSACHE	Ein (unsichtbarer) Menüpunkt, der die Login-URL erzeugt, hatte die Zugriffsebene „Registriert“. Wer schon eine Sitzung hatte, merkte nichts.
LÖSUNG	Menüpunkt auf „Öffentlich“ setzen. Diagnose: die Standardroute `component/users/login` muss sofort 200 liefern.

Aufgetreten in: Vereinswebsite, September 2026

F-53 Weiterleitungen-Backend stürzt ab

SYMPTOM	Der Aufruf von Komponenten → Weiterleitungen endet im Speicher- oder Zeitlimit.
URSACHE	Listenlimit „Alle“ plus 38.693 automatisch protokollierte 404-Einträge.
LÖSUNG	URL mit `list[limit]=20` aufrufen, leere 404-Einträge per SQL löschen, das Sammeln abschalten.

Aufgetreten in: eigene Website

F-54 Superglobals im Audit

SYMPTOM	Das Prüfwerkzeug bemängelt `$_POST` / `$_FILES` / `$_SERVER`.
URSACHE	Direkte Superglobals statt Joomlas Input-Schicht. Achtung: `Files->get()` ordnet die Upload-Struktur um.
LÖSUNG	Auf `$this->input` umstellen, die Normalisierung für Einzel- und Mehrfach-Upload anpassen und echte Uploads testen.

Aufgetreten in: Bilderupload 5.0.31

F-55 Redirect-Schleife nur lokal unter Windows

SYMPTOM	`Location: \/install.php` – der Browser dreht sich im Kreis.
URSACHE	`dirname()` liefert unter Windows-PHP auch für URL-Pfade einen Backslash.
LÖSUNG	Ergebnis von `dirname()` auf URL-Pfaden immer mit `str_replace('\\', '/', …)` normalisieren.

Aufgetreten in: Fotowettbewerb-Admin

KAPITEL 11

JED-Checker, Audits & Auslieferung

> *Automatische Prüfwerkzeuge sind streng und manchmal eigenwillig. Und der letzte Schritt – die Übergabe – ist der, an dem am leichtesten etwas durchrutscht.*

Das Prüfwerkzeug des Joomla Extensions Directory (JED-Checker) sucht nach Mustern, nicht nach Absichten. Es unterscheidet nicht zwischen Code und Kommentar und nicht zwischen einer harmlosen und einer verdächtigen Kodierung. Wer das weiß, spart sich Runden.

Der Audit-Ablauf

1. Ein Prüfskript je Erweiterung (`audit_<produkt>.py`) erzeugt den Bericht als PDF.
2. Befunde werden nach Schwere sortiert: Critical, High, Medium, Low – jeweils mit Datei und Zeile.
3. Korrekturen fließen in eine neue Version; eine Nachprüfung bestätigt den Stand.
4. Der Bericht landet im Audit-Ordner – nicht nur als Anzeige im Chat.

F-56 „multiple encoded, most probably obfuscated code“

SYMPTOM	Der JED-Checker schlägt Alarm bei völlig harmlosem Code.
URSACHE	Das Muster greift bei jeder Encode/Decode-Funktion (`base64_*` , auch `rawurlencode`) und bei `strtr()` mit Ersetzungs-Array.
LÖSUNG	Kodierung weglassen, wo sie unnötig ist (POST kodiert selbst); Platzhalter per Schleife mit `str_replace()` ersetzen.

Aufgetreten in: JoomION Widerruf

F-57 Erklärkommentar verstärkt den Fund

SYMPTOM	Nach dem Hinzufügen erklärender Kommentare bleibt die Warnung – oder wird häufiger.
URSACHE	Der JED-Checker sucht reinen Text, auch in Kommentaren. Die Kommentare zitierten genau die gesuchten Begriffe.
LÖSUNG	Kommentare so formulieren, dass sie die Information liefern, ohne die Trigger-Wörter zu wiederholen.

Aufgetreten in: Security Scanner 1.00.88–1.00.93

F-58 Methode nach Komplett-Neuschreiben verschwunden

SYMPTOM	Die gesamte Website zeigt einen Fatal Error – zwei Versionen lang unbemerkt.
URSACHE	Eine Klassendatei wurde vollständig neu geschrieben; eine weiterhin aufgerufene Methode ging verloren. `php -l` prüft nur Syntax.
LÖSUNG	Nach jedem Komplett-Rewrite aufgerufene gegen definierte Methoden per `grep` abgleichen.

Aufgetreten in: Countdowner 1.6.0

F-59 HTTP 500 direkt nach dem Live-Deploy

SYMPTOM	Die frisch hochgeladene Anwendung startet nicht.
URSACHE	Eine lokale `config.php` mit Testzugangsdaten war nach einer späteren Testrunde wieder entstanden und mit ausgeliefert worden.
LÖSUNG	Vor jeder Übergabe – nach jeder Testrunde erneut – auf Config-, Lock- und Testdateien prüfen.

Aufgetreten in: Fotowettbewerb-Admin

Vor jeder Übergabe

Nach der letzten Testrunde – nicht nach der ersten – nach Konfigurationsdateien, Lock-Dateien, Testbildern und Debug-Skripten suchen. Bei installierbaren PHP-Anwendungen entstehen sie mit jedem Testlauf neu.

TEIL IV

Die Lizenz-Geschichte

12 · Vom Lizenzsystem zur freien Software

KAPITEL 12

Vom Lizenzsystem zur freien Software

> *Die spannendste Wendung des Zeitraums war keine technische, sondern eine strategische: Ein fertiges Lizenz- und Update-Sperrsystem wurde gebaut – und wenige Wochen später bewusst stillgelegt.*

Akt 1 – Der LicManager

Im Juni entstand **JoomION LicManager**: eine Joomla-6-Komponente für den Softwareverkauf mit Produktverwaltung, Warenkorb, PayPal-Kauf, automatischer Kontoanlage, Lizenztypen mit frei einstellbarer Laufzeit und einem geschützten Download-Verzeichnis außerhalb des Webroots – dazu HMAC-signierte Lizenzschlüssel, Rechnungen und Widerruf.

Ziel	Bezahlte Erweiterungen; bei abgelaufener Lizenz keine Updates
Prinzip	Die Funktion bleibt erhalten – nur Updates werden serverseitig gesperrt
Technik	Update-Site mit `extra_query` (Lizenz, E-Mail, Domain), öffentliche API `api.validate`, `api.update`, `api.download`
Besonderheit	Wiederverkaufbar: Der Lizenzserver ist nicht fest verdrahtet, sondern konfigurierbar

Akt 2 – Die Mühen der Update-Sperre

Der native Joomla-Updater erwies sich als schwieriger Partner. Updates sollten nur bei gültiger Lizenz erscheinen, ohne Lizenz sollte eine klare Meldung stehen. Unterwegs fanden sich echte Fehler – etwa ein Paketpfad, der Dateien im geschützten Verzeichnis nie fand –, dazu die Eigenheiten von Update-Sites, Cache und `<client>` aus Kapitel 7.

Eine Zwischenlösung ließ Joomla das Update immer anzeigen und markierte fehlende Lizenzen nur im Namen – technisch sauber, für Anwender aber verwirrend. Der Punkt blieb als offene Frage stehen.

Akt 3 – Die Entscheidung

Ende Juni fiel die Entscheidung: **Alle JoomION-Erweiterungen werden freie Software (GPL v2+).** Keine Lizenzprüfung, keine Update-Garantie, Support auf Anfrage, freiwillige Spende. Updates laufen über einen einfachen, statischen Feed auf einem eigenen Update-Server. LicManager und Warenkorb-Plugin bleiben unangetastet – sie werden nur nicht mehr eingebunden.

Das Rollout-Rezept

Weil zwei Dutzend Erweiterungen umzustellen waren, entstand ein festes Rezept:

1. Manifest: `<updateservers>` auf den freien Feed; bei Paketen nur am Paket.
2. Statische Feed-Datei mit zwei Blöcken (Joomla 5 und 6) und drei Prüfsummen erzeugen.
3. Lizenz-Helfer, Lizenzfelder und Synchronisation entfernen.
4. Hinweisbalken: „Freie Software (GPL v2+) – keine Update-Garantie … freiwillige Spende“.

5. Das Install-Skript migriert bestehende Update-Sites auf den freien Feed und leert alte Parameter.
6. Dokumentation neu erzeugen: Lizenzabschnitt → frei, GPL, Spende.

Aus dem LicManager wurde außerdem per Fork der **Erweiterungs-Downloader**: gleiche Oberfläche, aber ohne Preise, Warenkorb, Lizenzen und Rechnungen – dafür mit Datenschutz-Einwilligung, Sammel-Download als ZIP und einem Spenden-Link auf der Dankeseite.

Was bleibt

Die Arbeit am Lizenzsystem war nicht umsonst. Viele Fallstricke rund um Update-Sites, `<client>`, Install-Skripte und Versionsanzeige wurden genau dabei gefunden – und kommen heute jedem freien Update-Feed zugute.

Die Lehre

Eine Architektur, die sich gegen das Werkzeug stemmt – hier gegen den nativen Updater –, kostet dauerhaft Kraft. Die einfachere Lösung aus statischem Feed und freier Lizenz hat die meisten Probleme nicht gelöst, sondern überflüssig gemacht.

TEIL V

Fallstudien jenseits von Joomla

13 · Werkzeuge für einen Fotoverein

14 · Büro-Software

15 · Abseits vom Code

KAPITEL 13 · FALLSTUDIE

Werkzeuge für einen Fotoverein

> *Ein Verein braucht keine Unternehmenssoftware, sondern passgenaue kleine Werkzeuge. Genau hier spielt die Entwicklung mit KI ihre Stärken aus.*

Projekte	Bilderabstimmung · Fotowettbewerb · Jury · Fotoquiz · Mitgliedsantrag · Fragebogen
Technik	PHP ohne Framework; zunächst JSON-Dateien, später MySQL; Bootstrap-Popups
Kernidee	Kleine, eigenständige Werkzeuge, die sich per Popup in die Vereinswebsite einfügen

Wettbewerb, Abstimmung, Jury

Die **Bilderabstimmung** führt eine Jury durch Annahme-Runden, eine Punkterunde und die Endplatzierung – einschließlich automatischer Klärungsrunden bei Gleichstand. Aus einer Analysefrage („Wie staffeln wir fair nach Bildanzahl?“) wurde noch am selben Tag eine prozentuale Behaltequote mit Rundenschätzung.

Der **Fotowettbewerb** wurde von einer statischen Seite zu einem konfigurierbaren System mit Admin-Bereich, dynamischen Altersgruppen, geschützten Einreichungen und Sponsoren. Am 29. September wurden Wettbewerb und Jury zu *einer* Anwendung zusammengeführt – mit echten Jury-Konten, getrennter Wertung je Kategorie und Altersgruppe und Namen, die erst nach feststehender Platzierung sichtbar werden.

Nebenbei gefunden

Beim Portieren der Jury-Logik fiel ein Fehler der alten Version auf: Wurden zwei Klärungsrunden parallel angelegt, blieb eine davon für immer unsichtbar. Die neue Version verknüpft jede Runde ausdrücklich mit ihrer Vorgängerin.

Quiz, Antrag, Fragebogen

Drei kleine Werkzeuge ersetzen oder ergänzen Formulare der Vereinswebsite und laufen im Popup:

- **Mitgliedsantrag:** aus einem Formular-Export mit 84 Feldern nachgebaut: dreistufiger Assistent, Sichtbarkeitsregeln, Altersberechnung auf dem Server, Honeypot und Mindest-Ausfüllzeit gegen Spam.
- **Fragebogen:** zehn Fragen, Punkte und Farbskala – komplett ohne Server-Verarbeitung, weil nichts gespeichert werden muss.
- **Fotoquiz:** 200 Fragen in drei Stufen, Zeitbonus, Ranglisten. Die richtige Antwort verlässt den Server erst bei der Auswertung; statt Cookies sichern HMAC-signierte Tokens den Ablauf – damit läuft das Quiz auch eingebettet im iFrame.

PROMPT AN CLAUDE CODE

Härte das Quiz gegen Manipulation: Namen und E-Mail per Whitelist prüfen, Content Security Policy mit Nonce, nur JSON annehmen, Anfragen je IP begrenzen, Admin-Login mit Sperre nach Fehlversuchen. Zeige zuerst, welche Eingaben heute durchgehen würden.

Eine Überraschung aus der Härtung

`filter_var(…, FILTER_VALIDATE_EMAIL)` akzeptiert Adressen wie `"<script>"@x.de` – in Anführungszeichen ist der lokale Teil fast beliebig. Die Ausgabe war maskiert, die Eingabe aber offen. Seitdem gilt eine strikte Whitelist.

KAPITEL 14 · FALLSTUDIE

Büro-Software

> *Rechnungen schreiben, Belege erfassen, Arbeitszeit dokumentieren: vier Werkzeuge mit sehr unterschiedlicher Technik – und eine gemeinsame Lehre über Entscheidungen.*

Werkzeug	Technik	Kern
Arbeitsnachweis	PHP, MySQL, FPDF	Kunden, Baustellen, Fotos per Tablet, Material, Zeiten, Unterschrift am Touchscreen, PDF-Bericht
Rechnungstool	Excel + VBA	fortlaufende Nummern, Journal nur zum Anhängen, PDF-Export
Rechnungsauslese	Browser, pdf.js, Tesseract.js	Eingangsrechnungen einlesen, Beträge erkennen, auch Scans per OCR, Export nach Excel
REÜR	PHP, MySQL, tFPDF	Kunden, Artikel, Rechnungen, PDF-Import von Ausgaben, EÜR und Umsatzsteuer

Entscheidungen gehören dem Menschen

Beim Rechnungstool standen zwei Wege zur Wahl: eine robuste Datenbanklösung oder eine schnelle Excel-Mappe mit Makros. Claude Code stellte die Abwägung dar – einschließlich des ehrlichen Hinweises, dass Blattschutz und VBA keine echte Unveränderbarkeit garantieren. Die Entscheidung fiel bewusst für Excel. Knapp einen Monat später entstand mit REÜR die Datenbankvariante.

Ein Werkzeug an einem Tag

Die Rechnungsauslese entstand an einem Tag: Text-PDFs lesen, Rechnungsnummer, Datum, Verkäufer und Beträge per Heuristik erkennen, unsichere Felder markieren, Summen je Steuersatz bilden, nach Excel exportieren. Als gescannte Belege dazukamen, folgte die Texterkennung im Browser – mit drei technischen Hürden, die in Anhang A dokumentiert sind.

Live im Einsatz

Der Arbeitsnachweis ist seit August bei einem Handwerksbetrieb im täglichen Einsatz. Die Weiterentwicklung geschieht seitdem am laufenden System – mit der Regel, nach jeder Änderung die betroffenen Dateien aufzulisten, damit einzelne Dateien gezielt per FTP ersetzt werden können.

ANWEISUNG AUS DER PRAXIS

Schreibe mir nach jeder Änderung, welche Dateien ersetzt werden müssen.

KAPITEL 15 · FALLSTUDIE

Abseits vom Code

> *Nicht jede Aufgabe war Programmierung im engeren Sinn. Gerade die Ausflüge in fremde Werkzeuge zeigen, wie breit ein agentischer Assistent einsetzbar ist – und wo er improvisieren muss.*

Dalli-Klick: ein Partyspiel in PowerPoint

Sechseckige Kacheln verdecken ein Bild und verschwinden im Sekundentakt, dazu läuft Musik. Das Spiel lebt in einer PowerPoint-Datei mit Makros. Änderungen am VBA-Code nahm Claude Code per COM-Automatisierung vor – und lernte dabei, dass zwei Kopien der Datei (Arbeitskopie und synchronisierte Spielkopie) immer gemeinsam geändert werden müssen.

Wenn der Mensch „nein“ sagt

Ein technisch eleganterer Musik-Player mit einem Dauerprozess wurde gebaut und getestet – und verworfen, weil „Anhalten“ und „Aufdecken“ sich nicht mehr richtig verhielten. Die kleine Verzögerung beim Musikstart ist seitdem bewusst akzeptiert.

Präsentationen ohne Node.js

Die übliche Werkzeugkette für Foliensätze setzt Node.js voraus, das auf dem Arbeitsrechner nicht installiert ist. Also entstanden Präsentationen – etwa für einen Fotografie-Anfängerkurs – mit python-pptx, selbst gezeichneten Icons, LibreOffice für die PDF-Umwandlung und PyMuPDF für die Sichtprüfung.

Ein Kirchenbuch von 1864

Fotografierte, gewölbte Buchseiten wurden entzerrt, auf Lesbarkeit optimiert und transkribiert – eine Aufgabe zwischen Bildverarbeitung und Handschriftenlesen.

Die Offline-Kopie einer Vereinswebsite

Eine per Baukasten erstellte Website sollte offline lesbar bleiben. Das Spiegel-Werkzeug hatte jedoch JavaScript beschädigt, und die Fotogalerien funktionierten ohne Server gar nicht. Claude Code reparierte die Verweise, schrieb eine eigene Offline-Galerie mit Diashow und Lightbox und lud 523 fehlende Fotos gezielt nach.

Übertragbares Muster

In allen vier Fällen war das Vorgehen gleich: das vorhandene Werkzeug respektieren, die Einschränkungen der Umgebung früh prüfen (etwa mit `node -v`) und eine Umgehung wählen, die ohne zusätzliche Installation auskommt.

TEIL VI

Lehren

16 · Sechzehn Arbeitsregeln
17 · Gedächtnis, Regeln, Generatoren
18 · Bilanz & Checklisten

KAPITEL 16

Sechzehn Arbeitsregeln

> *Aus den Fehlermustern sind Regeln geworden. Jede hat eine konkrete Vorgeschichte – und jede steht heute im Gedächtnis von Claude Code, damit sie in der nächsten Sitzung wieder gilt.*

Die Regeln sind bewusst einfach formuliert. Ihr Wert liegt nicht in der Einsicht – die meisten klingen selbstverständlich –, sondern darin, dass sie *jedes Mal* angewendet werden.

#	Regel	Was sie bedeutet	Anlass
1	**Exakte Fehlermeldung vor dem zweiten Versuch**	Bei jedem Fehler, dessen Ursache nicht aus der eigenen Änderung hervorgeht, zuerst Rohdaten holen: Stacktrace, Log, Konsole.	Zwei falsche Diagnosen beim Totalausfall einer Website
2	**Die gelieferte Datei ist die Wahrheit**	Nennt der Mensch ein ZIP, wird daraus gearbeitet – lokale Arbeitsordner nur zum Vergleich.	Bugfix auf veraltetem Code in fünf Erweiterungen
3	**Den Auftrag exakt eingrenzen**	Bei „nur dort, wo X“ zuerst alle Fundstellen von X ermitteln, dann nur diese ändern.	Neue Option in fünfzehn statt in wenigen Auswahlfeldern
4	**Nur das Verlangte**	Nichts „zur Sicherheit“ mit erledigen – auch keine Dokumente anderer Erweiterungen.	Drei fremde Dokumentationen neu erzeugt
5	**Bei großen Neuprojekten fragen**	Speicherung, Hosting, Startzustand, Etappen: Architekturentscheidungen trifft der Mensch.	„Bei Fragen entscheide nicht selbst, sondern frage mich.“
6	**Nach jeder Änderung eine Dateiliste**	Geänderte und neue Dateien mit relativem Pfad nennen, nicht nur „ZIP neu hochladen“.	Gezielter FTP-Austausch im Live-Betrieb
7	**Jede Änderung, neue Version – alte Pakete bleiben**	Versionsnummer in allen Manifesten und im Build-Skript erhöhen; ältere ZIPs nie überschreiben.	Nachvollziehbarkeit über hunderte Zwischenstände
8	**Der Update-Feed gehört zum Build**	Version, Download-URL und drei Prüfsummen im selben Schritt – per Skript gegen die Datei geprüft.	Acht Scanner-Versionen ohne Feed
9	**Vor jeder Übergabe aufräumen – jedes Mal**	Nach der letzten Testrunde erneut nach Config-, Lock- und Testdateien suchen.	HTTP 500 beim Live-Deploy
10	**Nach einem Komplett-Rewrite abgleichen**	Aufgerufene gegen definierte Methoden per `grep`; `php -l` reicht nicht.	Verlorene Methode, Website-Ausfall
11	**Präzise Einstellungen statt Voreinstellungen**	Bei gestuften Optionen je Stufe ein eigenes Zahlenfeld; Vorlagen höchstens zusätzlich.	Badge-Umbau von vier Presets auf vier Ebenen
12	**Feste Ablageorte, fremde Ordner tabu**	Alles in den vereinbarten Unterordnern des Ausgabeverzeichnisses; vorgefundene Ordner nicht anfassen.	Pakete im Ordner für Veröffentlichtes

#	Regel	Was sie bedeutet	Anlass
13	**Cache-Busting von Anfang an**	Jede CSS- und JS-Datei mit `?v=<filemtime>` einbinden.	Zweimal scheinbar wirkungslose CSS-Korrekturen
14	**Kopierte Vorlagen gegen bekannte Fallen prüfen**	Übernommener Code ist nicht geprüfter Code – gegen die Fallstrick-Liste gegenlesen.	Rückfall in den `echo`-Fehler
15	**Audit-Empfehlungen im Kontext prüfen**	Eine allgemein richtige Empfehlung kann im konkreten Render-Kontext falsch sein.	WebAssetManager im Ad-hoc-Rendering
16	**Im echten Browser und auf echten Geräten testen**	Zustandswechsel live beobachten; Touch auf echten Tablets prüfen.	Versteckte Elemente, die sichtbar blieben

Wie aus einer Erfahrung eine Regel wird

Eine Regel wird mit ihrem Grund (*Warum*) und ihrer Anwendung (*Wie*) notiert. Ohne Grund wird sie bei der ersten Ausnahme über Bord geworfen; ohne Anwendung bleibt sie ein guter Vorsatz. So sieht ein solcher Gedächtniseintrag aus:

```
---
name: feedback-exakte-fehlermeldung-vor-fix
description: Bei gemeldetem Fehler zuerst die exakte Meldung einholen, nicht raten
metadata:
  type: feedback
---
Bei einem gemeldeten Fehler zuerst nach der exakten Fehlermeldung fragen …

**Why:** Zwei plausible, aber falsche Diagnosen beim Totalausfall …
**How to apply:** Spätestens nach einem erfolglosen Versuch Rohdaten anfordern …
```

KAPITEL 17

Gedächtnis, Regeln, Generatoren

> *Was die Zusammenarbeit über hundert Sitzungen trägt, ist keine einzelne Fähigkeit, sondern Infrastruktur: ein Gedächtnis, verbindliche Regeln, feste Ordner und Generatoren.*

Das Gedächtnis

Jede Sitzung beginnt ohne Erinnerung an die vorige. Deshalb führt Claude Code ein Verzeichnis kurzer Notizen mit einem Index, der zu Beginn jeder Sitzung geladen wird. Vier Arten haben sich bewährt:

Art	Inhalt	Beispiel
user	Wer hier arbeitet, welche Vorlieben gelten	Immer auf Deutsch, echte Umlaute
feedback	Korrekturen und bestätigte Vorgehensweisen	Nach jeder Änderung eine Dateiliste
project	Stand, Entscheidungen, offene Punkte	Fotowettbewerb: Variante C, Test auf dem Server
reference	Technische Erkenntnisse, Fundstellen	Joomla: packagename = Dateiname

Gedächtnis altert

Notizen spiegeln den Stand beim Schreiben. Mehrfach widersprachen sich Einträge oder verwiesen auf Ordner, die inzwischen verschoben waren. Deshalb gilt: Bevor eine Notiz zur Grundlage einer Änderung wird, wird geprüft, ob Datei oder Funktion noch existieren.

Die REGELN

Für alle JoomION-Erweiterungen gibt es ein verbindliches Regelwerk – als Datei im Ausgabeordner und als Skill, der in jeder Sitzung gelesen wird. Es legt unter anderem fest:

- [ ] Jede Komponente hat ein Dashboard als Startseite.
- [ ] Überall ein einheitlicher Backend-Balken mit Logo, Name und Version.
- [ ] Eine Hilfeseite in der Erweiterung, Deutsch und Englisch per Tab.
- [ ] Externe Anleitung und FAQ in Word, HTML und PDF, jeweils Deutsch und Englisch.
- [ ] Update-Funktion über den freien Feed.
- [ ] Sicherheits- und API-Audit für jede Erweiterung.
- [ ] Versionsbump bei jeder Änderung, Ausgabe nur in die vorgesehenen Ordner.

Feste Ordner

Ordner	Inhalt
`dist\ZIPs`	alle Installationspakete, alle Versionen
`dist\ZIPs\FERTIG`	veröffentlichte Pakete – gehört ausschließlich dem Menschen
`dist\Audits`	alle Prüfberichte als PDF

Ordner	Inhalt
`dist\Anleitungen und FAQ`	Dokumentation in allen Formaten und Sprachen
`dist\Updateserver`	Update-Feeds je Erweiterung
`dist\py`	Generatoren und Hilfsskripte
`dist\_work`	Quell- und Arbeitsordner
`dist\Webseiten`	Web-Projekte und Beschreibungstexte

Generatoren statt Handarbeit

- **Dokumentation:** eine Inhaltsdatei je Erweiterung → Anleitung und FAQ in vier Formaten, zwei Sprachen
- **Audits:** ein Prüfskript je Erweiterung → PDF-Bericht im Audit-Ordner
- **Build:** ein Build-Skript je Paket → ZIP mit korrekten Pfaden und LF-Zeilenenden
- **Feeds:** Prüfsummen und Versionen werden aus der gebauten Datei berechnet, nicht abgetippt
- **Dieser Bericht:** Buch und Website entstehen aus derselben Datenbasis per Generator

KAPITEL 18

Bilanz & Checklisten

> *Was bleibt nach vier Monaten? Eine nüchterne Bilanz – und vier Checklisten, die man neben die Tastatur legen kann.*

Die Bilanz

Was sich bewährt hat	Was Lehrgeld gekostet hat
Komplette Pakete aus einer Beschreibung	Raten statt nach der Fehlermeldung fragen
Dokumentation und Audits per Generator	Arbeiten mit veralteten Quellen
Diagnose-Werkzeuge statt Vermutungen	Aufträge eigenmächtig ausweiten
Belegen im Joomla-Quellcode	Vorlagen und Empfehlungen ungeprüft übernehmen
Wiederverwendung bewährter Bausteine	Pflichtschritte am Ende vergessen
Ein Gedächtnis mit Regeln und Gründen	Nur textuell statt im echten Browser testen

Die rechte Spalte hat sich im Lauf der Monate nicht von selbst geleert. Die KI beginnt jede Sitzung neu – aber jede Lektion wurde als Regel mit Grund notiert und galt beim nächsten Mal wieder. Das ist die eigentliche Methode hinter diesem Bericht.

Die eine Kerngewohnheit

Wenn Sie sich nur eine Sache merken: **Erst belegen, dann ändern.** Belegen heißt: die exakte Meldung, die tatsächlich ausgelieferte Datei, die konkrete Stelle im Quellcode, der Test im echten Browser.

Checkliste: Vor dem Build

- [ ] Versionsnummer in allen Manifesten und im Build-Skript erhöht?
- [ ] Manifest-Dateinamen, `<name>`, `<packagename>` und Namespace geprüft?
- [ ] Install-Skripte ohne `echo`, mit `try/catch`?
- [ ] Alle Textdateien mit LF, ZIP mit Schrägstrich-Pfaden – Eintragsnamen ausgelesen?
- [ ] Update-Feed mit Version, URL und drei Prüfsummen aktualisiert und verifiziert?
- [ ] Anleitung, FAQ und Hilfeseite der betroffenen Erweiterung angepasst – nur dieser?

Checkliste: Vor der Übergabe

- [ ] Nach der letzten Testrunde erneut nach Config-, Lock- und Testdateien gesucht?
- [ ] Nach Komplett-Rewrites Methodenaufrufe gegen Definitionen abgeglichen?
- [ ] Zustandswechsel und Interaktionen im echten Browser geprüft?
- [ ] Liste der geänderten Dateien mit relativen Pfaden erstellt?
- [ ] Ablage in den vereinbarten Ordnern, nichts in fremden Ordnern verändert?

Checkliste: Bei einem Fehlerbericht

- [ ] Exakte Fehlermeldung, Log oder Konsole vorhanden – oder angefordert?
- [ ] Mit der tatsächlich ausgelieferten Version gearbeitet?
- [ ] Browser-Cache (Strg+F5) als Ursache ausgeschlossen?
- [ ] Liegt die Ursache vielleicht außerhalb des eigenen Codes (Erweiterung, Konfiguration, Server)?
- [ ] Fallstrick-Katalog (Anhang A) nach dem Symptom durchsucht?

Checkliste: Bei einem neuen Projekt

- [ ] Architekturfragen (Speicher, Hosting, Etappen) gestellt und beantwortet?
- [ ] Vorhandene, bewährte Bausteine identifiziert – und gegen bekannte Fallen geprüft?
- [ ] Arbeitsordner an der vereinbarten Stelle angelegt?
- [ ] Cache-Busting, Ready-Wrapper und LF von Anfang an eingeplant?

ANHANG

Nachschlagen

A · Katalog der Fallstricke

B · Prompt-Bibliothek aus der Praxis

ANHANG A

Katalog der Fallstricke

> *Das Nachschlagewerk zum Buch: jeder Fallstrick mit Nummer, Symptom und Lösung – zum schnellen Suchen, wenn etwas nicht funktioniert.*

Alle 68 Fallstricke in Kurzform, nach Themen geordnet. Die ausführliche Fassung mit Symptom, Ursache und Lösung steht im jeweiligen Kapitel.

Paketbau & Manifest

Nr.	Fallstrick & Symptom	Lösung
F-01	**ZIP mit Backslash-Pfaden** Joomla meldet „Manifest-Datei konnte nicht gefunden werden“, obwohl die XML-Datei korrekt im ZIP-Wurzelverzeichnis liegt.	ZIP über `System.IO.Compression.ZipArchive` bauen und jeden Eintragsnamen auf `/` normalisieren. Nach dem Build die Eintragsnamen stichprobenartig auslesen.
F-02	**CRLF statt LF in allen Paketdateien** Das Prüfwerkzeug meldet „Incorrect end-of-line character found“ für Sprach- und PHP-Dateien.	Beim Schreiben `newline='\n'` erzwingen und vor dem Zippen alle Textdateien normalisieren; danach per `grep -c $'\r'` gegenprüfen.
F-03	**Plugin-Manifest mit falschem Dateinamen** Installation bricht ab: „Die Datei <element>.xml konnte nicht gefunden werden“.	Plugin-Manifest exakt wie das Element benennen. Das Präfix `plg_<gruppe>_` gehört nur an Sprachdateien und den ZIP-Namen.
F-04	**Leeres „element“ beim Plugin-Install** Datenbankfehler „Field 'element' doesn't have a default value“.	Im Manifest `<folder plugin="name">services</folder>` setzen.
F-05	**Komponente landet im falschen Ordner** „Class … not found“, im Menü erscheint der rohe Sprachschlüssel.	`<name>` immer als Sprachschlüssel in Element-Form setzen (`COM_JOOMIONDOWNLOADER`); den Anzeigenamen liefert die `.sys.ini`.
F-06	**Namespace mit Administrator-Suffix** Autoload schlägt fehl oder funktioniert nur, weil `provider.php` Namespaces zur Laufzeit nachregistriert.	Im Manifest nur die Basisform (`Joomla\Component\Bilderupload`), dann den Workaround entfernen.
F-07	**packagename passt nicht zum Dateinamen** Die Versionsspalte wird nach „Cache aktualisieren“ leer, der native Updater zeigt nie ein Update – obwohl Feed und ZIP korrekt sind.	`<packagename>` exakt passend zum Dateinamen setzen; verwaiste Paket-Zeile danach aufräumen.
F-08	**Leerer media-Knoten** JED-Checker: „The node extension/media is empty“ – obwohl die Installation funktioniert.	Explizite `<folder>css</folder>` / `<folder>js</folder>`-Kinder eintragen.
F-09	**Komponenten-Optionen öffnen nicht** „Form::loadForm could not load file“ beim Klick auf „Optionen“.	Eigene `admin/config.xml` anlegen und im `<files>`-Abschnitt als `<filename>` registrieren.
F-10	**SQL-Ordner wird nicht gefunden** Installationsfehler „administrator/…/sql existiert nicht“.	`sql/` in den Admin-Ordner legen und dort in `<files>` auflisten.

Nr.	Fallstrick & Symptom	Lösung
F-11	**Jeder Menüklick landet auf dem Dashboard** Komponente installiert und aktiv, aber jeder Klick im Admin-Menü springt still auf das Joomla-Dashboard.	Menü-Links nur als Query schreiben: `option=com_x&view=y`. Diagnose-Trick: die Adresszeile ansehen.
F-12	**„Version ()“ auf der Datenbank-Wartungsseite** System → Wartung → Datenbank meldet eine nicht passende Update-Version mit leerer Klammer.	Manifest umbenennen, die alte Datei im `postflight` löschen und den `#__schemas`-Eintrag dort selbst setzen.

Installer, Updates & Versionen

Nr.	Fallstrick & Symptom	Lösung
F-13	**„Unexpected token '<'“ bei der Installation unter Joomla 6** Die Installation bricht im Browser ab; unter Joomla 5 lief dasselbe Paket.	Ausgabe puffern (`ob_start()`) und per `enqueueMessage()` anzeigen. Jedes kopierte Install-Skript vor der Auslieferung auf `echo` prüfen.
F-14	**Hinweise der Unter-Erweiterungen fehlen** Aktivierungs-Buttons aus dem Template-Skript erscheinen nach der Paketinstallation nicht.	Hinweise in das `script.php` des Pakets verlagern.
F-15	**Rote Leiste ohne Text** Die Installation endet mit einer leeren roten Fehlermeldung.	Natives PHP (`mkdir`, `file_put_contents`) verwenden und das ganze Skript in `try/catch` kapseln.
F-16	**Install-Skriptklasse wird nicht gefunden** Das Skript läuft schlicht nicht – kein Fehler, keine Wirkung.	Verifizierte Namensschemata nutzen oder das moderne Muster mit anonymer Klasse und `InstallerScriptInterface`.
F-17	**Update wird nicht angezeigt – fehlendes <client>** Feed erreichbar, Version und Hashes stimmen, der Katalog zeigt die Erweiterung – aber Joomla meldet kein Update.	`<client>` je Typ setzen (Komponente: `administrator`, sonst `site`). Nach einer Reparatur „Cache leeren“, sonst prüft Joomla stundenlang nicht neu.
F-18	**Update-Site fehlt nach einem Update** Keine Update-Meldung, obwohl der Server die neue Version ausliefert.	Backend: Aktualisierungs-Sites → „Wiederherstellen“. Im Paket: `syncUpdateSite()` im `postflight`.
F-19	**TUF-Meldung „timestamp expired“** Nur eine von mehreren gleichen Joomla-Instanzen meldet einen Fehler beim Kern-Update.	Die Tabelle `#__tuf_metadata` keinesfalls leeren – das legte die Update-Seite zweimal komplett lahm. Stattdessen Backup oder den Wert einer gesunden Schwester-Instanz übernehmen.
F-20	**Falsche Versionsnummer in der Oberfläche** Das Backend zeigt nach einem Update weiterhin die alte Version.	Version immer aus der installierten Manifest-Datei lesen (`simplexml_load_file`), `manifest_cache` nur als Rückfall. Bei der Suche XML-Dateien nicht ausschließen – dort sitzen die `note`-Felder.

Nr.	Fallstrick & Symptom	Lösung
F-21	**Acht Versionen ohne Update-Feed** Nutzer mit der freien Version bekommen keine Updates angeboten.	Feed, Download-URL und alle drei Hashes (SHA-256/384/512) gehören in denselben Arbeitsschritt wie der Build – und werden per Skript gegen die Datei geprüft.

Formulare, Felder & Backend

Nr.	Fallstrick & Symptom	Lösung
F-22	**Falsche Basisklasse für eigene Felder** `Class "Joomla\CMS\Form\Field\FormField" not found` beim Öffnen der Optionen.	Richtig importieren – oder von einem konkreten Typ wie `TextField` ableiten.
F-23	**Eigenes Feld erscheint als leeres Textfeld** Balken und Hilfe-Tab fehlen, stattdessen ein unbeschriftetes Eingabefeld. Cache leeren hilft nicht.	`addfieldpath` plus klassische `JFormField<Typ>`-Klasse ohne eigenen Namespace.
F-24	**Listenauswahl wird nicht gespeichert** Nach dem Speichern springt ein Auswahlfeld auf den alten Wert zurück.	Einen echten Platzhalterwert wie `default` verwenden – in Option und `default`-Attribut.
F-25	**Häkchen werden nicht gespeichert** Der POST-Wert kommt korrekt an, landet aber nie in der Datenbank.	Für jeden fremden Namen ein `<field type="hidden">` vor dem eigenen Feld deklarieren.
F-26	**Editor speichert fremden Inhalt** Im Frontend erscheinen Feldbeschriftungen als Sektionsinhalt.	Editor-Felder immer sichtbar rendern; kein `showon` auf Editor-Feldern.
F-27	**Auswahlfelder in wiederholbaren Subforms** `Uncaught Error: Misconfiguration` beim Verschieben einer Zeile; Beitragsauswahl in neu hinzugefügten Zeilen reagiert nicht.	In sortierbaren Subforms `type="text"` bzw. ein `type="sql"`-Dropdown verwenden.
F-28	**Speichern scheitert am veralteten Filter** „calls a deprecated filter class JComponentHelper“ beim Speichern eines Editor-Felds.	`filter="\Joomla\CMS\Component\ComponentHelper::filterText"`.
F-29	**Titel da, Toolbar-Buttons fehlen** Die Seite zeigt den Titel, aber kein „Neu“ und kein „Speichern“.	`Toolbar::getInstance('toolbar')` verwenden.
F-30	**„Direkter Aufruf dieser Seite nicht gestattet“** Ein Link auf ein Bearbeitungsformular endet mit Fehler #<id>.	Über `task=<controller>.edit&id=…` verlinken, nicht über `view=…&layout=edit`.
F-31	**Plugin bleibt nach „Zurück“ gesperrt** Nach eigenen „Speichern & zurück“-Buttons zeigt das Plugin ein Schloss.	Vor der Weiterleitung selbst einchecken (`plugins.checkin` mit Token).

Nr.	Fallstrick & Symptom	Lösung
F-32	**Button „Speichern & zurück“ tut nichts** Klick ohne jede Reaktion, keine Konsolenfehler.	`document.adminForm` verwenden; URLs für JavaScript per `json_encode()` ausgeben.
F-33	**„500 searchtools::default not found“** Listenansichten brechen unter Joomla 6 ab.	`LayoutHelper::render('joomla.searchtools.default', ['view' => $this])`.
F-34	**„Incorrect datetime value: ''“** Speichern bricht ab, sobald ein Datums- oder Zahlenfeld leer bleibt.	In `Table::check()` leere Datumsfelder auf `null`, Zahlen auf `0` setzen – bei jeder neuen Tabelle gleich mit.
F-35	**„Ungültige Übersetzungs-Zeichenkette“** Die Sprachprüfung meldet eine defekte Zeile.	Typografische Anführungszeichen verwenden und je Zeile die Anzahl gerader `"` prüfen. Automatische Korrekturen danach verifizieren – ein Skript hatte auch das abschließende Zeichen ersetzt.

JavaScript, CSS & Assets

Nr.	Fallstrick & Symptom	Lösung
F-36	**Buttons reagieren auf nichts** Kunde meldet: Speichern funktioniert nicht mehr. Kein Konsolenfehler, Verdacht auf `.htaccess`.	Jedes so registrierte Skript mit sofortigem DOM-Zugriff in einen `DOMContentLoaded`-Wrapper packen (zentrale Hilfsmethode).
F-37	**Modul funktionslos nach einem Audit-Fix** Nach der Umstellung auf den WebAssetManager öffnet der Button das Formular nicht mehr.	Im Ad-hoc-Render-Kontext CSS inline ausgeben. Audit-Empfehlungen immer gegen den tatsächlichen Render-Kontext prüfen.
F-38	**CSS-Korrektur wirkt scheinbar nicht** Nach einem Update „funktioniert nichts“ – ein riesiges, ungestyltes Logo, Farben unverändert.	Von Anfang an `?v=<filemtime>` an jede Datei hängen. Bei „geht nicht“ nach CSS-Änderungen zuerst Strg+F5 testen.
F-39	**hidden-Element bleibt sichtbar** Der Countdown friert ein, gleichzeitig erscheint der Ablauftext.	Immer eine Regel `.element[hidden]{display:none}` danebensetzen. Zustandswechsel mit kurzer Testzeit live im Browser beobachten.
F-40	**Touch-Drag scrollt die Seite** Das Badge lässt sich horizontal ziehen, vertikal scrollt die Seite oder das Smartphone lädt neu.	Sechs Bausteine kombinieren (touch-action, `preventDefault` auf touchstart, overscroll-behavior, touchmove auf document, abgesichertes `setPointerCapture`, Body-Lock) – oder für Touch direkt auf Touch Events umsteigen.
F-41	**Rahmen nur an geraden Kanten** Bei Hexagon, Stern oder Dreieck fehlt der Rahmen an den schrägen Kanten.	Eckige Formen als SVG-Polygon mit `stroke` und `vector-effect="non-scaling-stroke"` zeichnen.
F-42	**JavaScript erscheint als sichtbarer Text** Unter dem Suchfeld der Icon-Auswahl steht Programmcode.	Keine Inline-Handler; ein einmalig registrierter, delegierter Listener auf `document`.

Nr.	Fallstrick & Symptom	Lösung
F-43	**Optionsseite des Templates zerstört** Boxen stehen untereinander statt nebeneinander, Vorschauen außerhalb der Karten.	Neue Vorschau-Funktionen nur additiv im vorhandenen Layout-Feld einbauen. Zurück auf die letzte gute Version (1.9.41) und neu ansetzen.

Laufzeit, Sitzungen & Sicherheit

Nr.	Fallstrick & Symptom	Lösung
F-44	**Das ganze Backend friert ein** Die Fortschrittsleiste steht – und kein anderer Backend-Tab reagiert mehr.	Wiederholungen clientseitig per `setTimeout`; vor langsamen Aufrufen die Sitzung schließen, Timeouts knapp halten.
F-45	**„Ungültiger Token“ nur im Backend-Tab** Dieselbe Aktion funktioniert im Frontend, im Backend-Formular nicht.	Einen sitzungsunabhängigen, per HMAC mit dem Site-Secret signierten Admin-Token mit Ablaufzeit verwenden und die Berechtigung explizit prüfen.
F-46	**Verzeichnis entsteht unter administrator/** Das „sichere Verzeichnis“ wird am falschen Ort angelegt.	Konfigurierte Pfade immer gegen `JPATH_ROOT` auflösen und den tatsächlichen Zielpfad anzeigen.
F-47	**Falsche Uhrzeit in E-Mails** Bestätigungsmails zeigen die Serverzeit statt der Ortszeit.	Für Anzeigen `Factory::getDate('now', $app->get('offset', 'UTC'))`.
F-48	**„could not be passed by reference“** Beim Speichern bricht alles ab, das Frontend bleibt leer.	Objekt erst einer Variablen zuweisen, dann übergeben.
F-49	**Plugin aktiv, aber Events feuern nie** Das Task-Plugin taucht im Dialog „Neue Aufgabe“ nicht auf.	Dispatcher als ersten Parameter übergeben; `provider.php` gegen ein funktionierendes Plugin prüfen.
F-50	**Nicht freigegebene Admins sehen den Builder** Trotz „nur ein Benutzer“ sieht ein weiterer Administrator die Werkzeugleiste.	Beide Stellen mit identischer Logik führen; Änderungen immer an beiden nachziehen.
F-51	**„Ungültige Serverantwort“ nur bei großen Dateien** Uploads kleiner Dateien klappen, große scheitern – und im Joomla-Log steht nichts.	Clientseitiges Diagnose-Logging: Bei unlesbarer Antwort HTTP-Status und Antwortanfang per kleiner Zusatzanfrage ins Log schreiben – der erste Treffer zeigte die Ursache.
F-52	**ERR_TOO_MANY_REDIRECTS auf der Login-Seite** Nur manche Besucher, scheinbar nur ohne „www“, landen in einer Endlosschleife.	Menüpunkt auf „Öffentlich“ setzen. Diagnose: die Standardroute `component/users/login` muss sofort 200 liefern.
F-53	**Weiterleitungen-Backend stürzt ab** Der Aufruf von Komponenten → Weiterleitungen endet im Speicher- oder Zeitlimit.	URL mit `list[limit]=20` aufrufen, leere 404-Einträge per SQL löschen, das Sammeln abschalten.

Nr.	Fallstrick & Symptom	Lösung
F-54	**Superglobals im Audit** Das Prüfwerkzeug bemängelt `$_POST` / `$_FILES` / `$_SERVER`.	Auf `$this->input` umstellen, die Normalisierung für Einzel- und Mehrfach-Upload anpassen und echte Uploads testen.
F-55	**Redirect-Schleife nur lokal unter Windows** `Location: \/install.php` – der Browser dreht sich im Kreis.	Ergebnis von `dirname()` auf URL-Pfaden immer mit `str_replace('\\', '/', …)` normalisieren.

JED-Checker, Audits & Auslieferung

Nr.	Fallstrick & Symptom	Lösung
F-56	**„multiple encoded, most probably obfuscated code“** Der JED-Checker schlägt Alarm bei völlig harmlosem Code.	Kodierung weglassen, wo sie unnötig ist (POST kodiert selbst); Platzhalter per Schleife mit `str_replace()` ersetzen.
F-57	**Erklärkommentar verstärkt den Fund** Nach dem Hinzufügen erklärender Kommentare bleibt die Warnung – oder wird häufiger.	Kommentare so formulieren, dass sie die Information liefern, ohne die Trigger-Wörter zu wiederholen.
F-58	**Methode nach Komplett-Neuschreiben verschwunden** Die gesamte Website zeigt einen Fatal Error – zwei Versionen lang unbemerkt.	Nach jedem Komplett-Rewrite aufgerufene gegen definierte Methoden per `grep` abgleichen.
F-59	**HTTP 500 direkt nach dem Live-Deploy** Die frisch hochgeladene Anwendung startet nicht.	Vor jeder Übergabe – nach jeder Testrunde erneut – auf Config-, Lock- und Testdateien prüfen.

Werkzeuge jenseits von Joomla

Nr.	Fallstrick & Symptom	Lösung
F-60	**OCR-Worker lädt nicht** Tesseract.js meldet „importScripts … URL invalid“.	`workerBlobURL: false`, alle Pfade als absolute URLs, `gzip: false`.
F-61	**PDF-Rendering hängt endlos** Bei manchen Bild-PDFs blockiert `page.render()` ohne Fehler und ohne Timeout.	pdf.js-Version wechseln; bei Hängern Schritt für Schritt messen statt zu raten.
F-62	**Browser-Tool per Doppelklick ohne Funktion** Das lokal geöffnete HTML-Werkzeug liest keine PDFs.	Mitgelieferter Ein-Klick-Starter: kleiner PowerShell-`HttpListener`-Server plus `.bat`, der den Browser öffnet.
F-63	**Excel-Automatisierung wirft InvalidCastException** Sporadische Umwandlungsfehler beim Befüllen von Zellen per PowerShell.	Alle `.Value2`-Werte als Text übergeben, Formeln über `.Formula`.
F-64	**VBA-Änderung wird still nicht gespeichert** Das Makro-Patch-Skript meldet Erfolg, die Datei bleibt unverändert.	PowerPoint starten und per `GetActiveObject` übernehmen; Prozeduren über `ProcStartLine` / `ProcCountLines` ersetzen.

Nr.	Fallstrick & Symptom	Lösung
F-65	**Schlagschatten auf jeder Form** In mit python-pptx erzeugten Folien haben Rechtecke einen ungewollten Schatten.	`<p:style>` aus dem XML entfernen und ein leeres `<a:effectLst/>` einfügen.
F-66	**Zwei Modals schließen sich gemeinsam** Beim Schließen des inneren Popups verschwindet auch der Fragebogen.	Innerhalb eines Modals zwischen zwei Ansichten umschalten (Quiz ↔ eingebettetes Formular).
F-67	**Lokale Datenbank startet nicht mehr** XAMPP-MariaDB bricht beim Start mit „Incorrect file format“ ab.	Verwaiste `.frm`-Datei entfernen, `mysql_upgrade` ausführen.
F-68	**Offline-Kopie mit zerstörten Skripten** Logo-Link kaputt, Galerien bleiben leer, SyntaxError in der Konsole.	Zerstörte Stellen gezielt zurückbauen, eigene Offline-Galerie schreiben, fehlende Fotos gezielt nachladen.

Werkzeuge jenseits von Joomla – ausführlich

Die Fallstricke aus den Fallstudien (Teil V) stehen nur hier in voller Länge.

F-60 OCR-Worker lädt nicht

SYMPTOM	Tesseract.js meldet „importScripts … URL invalid“.
URSACHE	Der Worker läuft standardmäßig über eine Blob-URL, dadurch brechen relative Pfade; die schnellen Sprachdateien sind zudem nicht gzip-komprimiert.
LÖSUNG	`workerBlobURL: false`, alle Pfade als absolute URLs, `gzip: false`.

Aufgetreten in: Rechnungsauslese

F-61 PDF-Rendering hängt endlos

SYMPTOM	Bei manchen Bild-PDFs blockiert `page.render()` ohne Fehler und ohne Timeout.
URSACHE	Ein Fehler der eingesetzten pdf.js-Version, gefunden erst durch Zeitmessung jedes einzelnen Schritts.
LÖSUNG	pdf.js-Version wechseln; bei Hängern Schritt für Schritt messen statt zu raten.

Aufgetreten in: Rechnungsauslese

F-62 Browser-Tool per Doppelklick ohne Funktion

SYMPTOM	Das lokal geöffnete HTML-Werkzeug liest keine PDFs.
URSACHE	Web Worker funktionieren über `file://` in Chrome nicht zuverlässig.
LÖSUNG	Mitgelieferter Ein-Klick-Starter: kleiner PowerShell-`HttpListener`-Server plus `.bat`, der den Browser öffnet.

Aufgetreten in: Rechnungsauslese

F-63 Excel-Automatisierung wirft InvalidCastException

SYMPTOM	Sporadische Umwandlungsfehler beim Befüllen von Zellen per PowerShell.
URSACHE	Gemischte Zahl- und Text-Zuweisungen an `.Value2` über COM.
LÖSUNG	Alle `.Value2`-Werte als Text übergeben, Formeln über `.Formula`.

Aufgetreten in: Rechnungstool (Excel/VBA)

F-64 VBA-Änderung wird still nicht gespeichert

SYMPTOM	Das Makro-Patch-Skript meldet Erfolg, die Datei bleibt unverändert.
URSACHE	Mit `New-Object -ComObject` fehlen Pfad und Name der Präsentation; Text-Suche in Modulen scheitert an Zeilenumbrüchen.
LÖSUNG	PowerPoint starten und per `GetActiveObject` übernehmen; Prozeduren über `ProcStartLine` / `ProcCountLines` ersetzen.

Aufgetreten in: Dalli-Klick (PowerPoint)

F-65 Schlagschatten auf jeder Form

SYMPTOM	In mit python-pptx erzeugten Folien haben Rechtecke einen ungewollten Schatten.
URSACHE	`add_shape()` setzt ein `<p:style>` mit Theme-Effekten – `shadow.inherit = False` reicht nicht.
LÖSUNG	`<p:style>` aus dem XML entfernen und ein leeres `<a:effectLst/>` einfügen.

Aufgetreten in: Präsentationen ohne Node.js

F-66 Zwei Modals schließen sich gemeinsam

SYMPTOM	Beim Schließen des inneren Popups verschwindet auch der Fragebogen.
URSACHE	Bootstrap 5 unterstützt gestapelte Modals nicht sauber; das Aufräumen betrifft alle offenen.
LÖSUNG	Innerhalb eines Modals zwischen zwei Ansichten umschalten (Quiz ↔ eingebettetes Formular).

Aufgetreten in: Fragebogen-Popup

F-67 Lokale Datenbank startet nicht mehr

SYMPTOM	XAMPP-MariaDB bricht beim Start mit „Incorrect file format“ ab.
URSACHE	Eine korrupte Systemtabelle `mysql.proxies_priv`.
LÖSUNG	Verwaiste `.frm`-Datei entfernen, `mysql_upgrade` ausführen.

Aufgetreten in: alle lokalen PHP/MySQL-Projekte

F-68 Offline-Kopie mit zerstörten Skripten

SYMPTOM	Logo-Link kaputt, Galerien bleiben leer, SyntaxError in der Konsole.
URSACHE	Das Spiegel-Werkzeug hatte JavaScript-String-Verkettungen als URLs „repariert“; die Galerien des Baukastens laden Inhalte ohnehin nur vom Server.
LÖSUNG	Zerstörte Stellen gezielt zurückbauen, eigene Offline-Galerie schreiben, fehlende Fotos gezielt nachladen.

Aufgetreten in: Offline-Kopie einer Vereinswebsite (523 Fotos)

ANHANG B

Prompt-Bibliothek aus der Praxis

> *Formulierungen, die sich bewährt haben – jeweils für eine typische Situation.*

Fehlersuche mit Belegen

PROMPT AN CLAUDE CODE

Hier ist die exakte Fehlermeldung aus dem Joomla-Log: [...]. Erkläre zuerst, welche Zeile sie auslöst und warum. Ändere noch nichts.

Mit der richtigen Quelle arbeiten

PROMPT AN CLAUDE CODE

Entpacke das ZIP, das ich dir gebe, und arbeite ausschließlich daraus. Vergleiche danach mit dem lokalen Arbeitsordner und nenne mir jede Abweichung.

Auftrag eingrenzen

PROMPT AN CLAUDE CODE

Ändere nur die Boxen, die den Schalter „Unterstrichen“ enthalten. Liste mir zuerst alle Fundstellen auf und warte auf meine Freigabe.

Neues Projekt

PROMPT AN CLAUDE CODE

Bevor du beginnst: Stelle mir alle Fragen zu Speicherung, Hosting, Startzustand und Etappen. Entscheide nichts davon selbst.

Auslieferung

PROMPT AN CLAUDE CODE

Baue das Paket, aktualisiere den Update-Feed mit allen drei Prüfsummen, prüfe die Hashes gegen die Datei und liste mir alle geänderten Dateien mit Pfad auf.

Audit

PROMPT AN CLAUDE CODE

Erstelle ein PSR-12- und Sicherheitsaudit als PDF im Audit-Ordner. Prüfe jede Empfehlung gegen den tatsächlichen Render-Kontext, bevor du sie als Befund einstufst.

Test im Browser

PROMPT AN CLAUDE CODE

Setze die Zielzeit des Countdowns auf zehn Sekunden in der Zukunft, öffne die Seite im Browser und prüfe vor und nach dem Ablauf, welche Elemente sichtbar sind.

Impressum

Hundert Sitzungen mit Claude Code – Ein Erfahrungsbericht

© 2026 Roland Löffler

Alle Rechte vorbehalten. Dieses Werk einschließlich aller Texte und Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechts ist ohne Zustimmung des Autors unzulässig. Das gilt insbesondere für Vervielfältigungen, Übersetzungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Text, Satz & Layout: Roland Löffler · JoomION

Kontakt: Roland Löffler, Eisenbahnstraße 9, 72461 Albstadt

Die Inhalte dieses Buches wurden mit großer Sorgfalt erarbeitet. Trotz sorgfältiger Prüfung können Fehler nicht ausgeschlossen werden. Eine Garantie für Richtigkeit und Vollständigkeit wird nicht übernommen; die Anwendung der beschriebenen Vorgehensweisen erfolgt auf eigene Verantwortung.

Hundert Sitzungen mit Claude Code – Ein Erfahrungsbericht

Roland Löffler · JoomION · Ausgabe 2026 (Version 1.0.0) · Stand: September 2026

Grundlage: rund 100 Arbeitssitzungen mit Claude Code (27.05.–29.09.2026) und die dabei entstandenen Arbeitsnotizen. Versionsnummern entsprechen dem Stand dieser Notizen. Zugangsdaten, Kundendaten und Namen Dritter wurden bewusst weggelassen.

Joomla! ist eine eingetragene Marke von Open Source Matters, Inc. Claude und Claude Code sind Marken von Anthropic. Alle übrigen Marken gehören ihren jeweiligen Inhabern.