

PLAYBOOK

E-Commerce-Migration mit KI-Tools

Ein WordPress-zu-MedusaJS-Migrationsmuster aus echten Projekten: KI-gestützte Inventur, URL- und Datenmapping sowie gestufte Umstellung mit verifiziertem Projekttagbuch.

Warum Migrationen schiefgehen

Commerce-Migrationen scheitern leise. Die neue Storefront geht live, Redirects funktionieren grösstenteils, und drei Monate später merkt jemand, dass eine Produktattribut-Familie nie umgezogen ist. Der Fehlermodus ist selten technisch - es ist undokumentierter Umfang. Die Disziplin dagegen: ein schriftliches Inventar vor jeder Codezeile und ein öffentliches Tagebuch, das 'im Quellcode gemergt' von 'live in Produktion' unterscheidet.

Das Vier-Phasen-Muster

01 Inventur

Die Legacy-Site crawlen und das CMS exportieren: Seiten, Produkte, Kategorien, Medien, Redirects, aktiv genutzte Plugins. KI-Tools sind hier wirklich gut - Crawl-Output zusammenfassen und Seitentypen klassifizieren macht aus einer Woche Tabellenarbeit einen Tag.

02 Mapping

Jede Legacy-Entität bekommt ein Ziel: Medusa-Produkt, Kategorie, CMS-Seite, Redirect oder bewusster Verzicht. Die Liste bewusst ausgelassener Inhalte wird vom Kunden abgenommen - nichts verschwindet still.

03 Neuaufbau

Das neue Theme ist eigener Code, kein Builder-Export. Commerce-Daten wandern über Importskripte in MedusaJS-Entitäten, die idempotent erneut laufen können, solange sich Inhalte auf der alten Site noch ändern.

04 Umstellung

Gestuft: neue Storefront auf einer privaten Entwicklungsadresse, gegen eine Checkliste verifiziert, dann DNS-Wechsel. Das Tagebuch dokumentiert jeden Stand getrennt - Quellcode gemergt, Staging verifiziert, Produktion abgenommen.

Wo KI-Tools ihren Nutzen verdienen

- Inventur: Hunderte Legacy-URLs in Produkte, Kategorien, Redaktionelles und tote Seiten klassifizieren
- URL-Mapping: Redirect-Tabellen aus Crawl-Daten vorschlagen - immer geprüft, nie blind angewendet
- Datentransformations-Entwürfe: WordPress-Exportformen in Medusa-Importformen zur Prüfung durch Engineers übersetzen

- Regressions-Sweeps: gerenderte Ausgabe vor und nach Theme-Umbauten vergleichen

Entwicklungsadressen bleiben privat. Staging-URLs sind für Team und Kunde - öffentliche Projektseiten zeigen die Arbeit, nie den Vorschau-Hostnamen.

Die Disziplin des ehrlichen Tagebuchs

Seiten zu laufenden Projekten funktionieren, weil sie exakt sagen, wo die Dinge stehen. Jeder Eintrag nennt ein verifiziertes Ereignis - eine gemergte Änderung, eine Staging-Lieferung, einen Abnahmevermerk - und der Seitenkopf sagt klar, dass die Produktionsmigration läuft. Kunden, die ihre eigene Projektseite lesen, sollen ihr Projekt erkennen; Fremde sollen lernen, wie Zusammenarbeit mit uns aussieht.

- ☐ Schriftliches Inventar existiert, bevor Migrationscode beginnt
- ☐ Jede Legacy-Entität hat ein gemapptes Ziel oder einen genehmigten Verzicht
- ☐ Importskripte sind idempotent und während der Transition erneut lauffähig
- ☐ Staging-Verifikation abgeschlossen und dokumentiert vor dem DNS-Wechsel
- ☐ Redirect-Abdeckung nach Launch gemessen, nicht angenommen

Dieses Dokument ist Teil der UHL-Ressourcen. Direkter Download und verwandte Arbeiten: <https://uhl-services.ch/de/resources/>

