Deployment Prozesse Praxisbeispiele

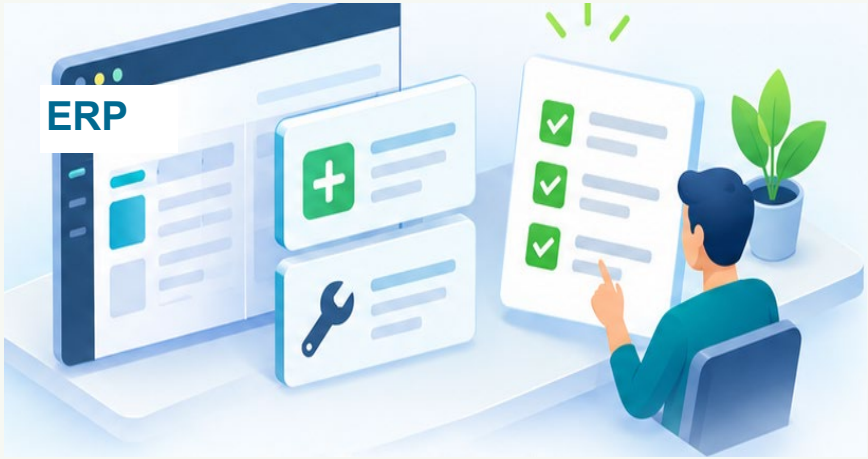
Gleiche Ziele - unterschiedliche Prozesse - unterschiedliche Anforderungen

Rückblick: HBT



Deployment-Prozesse für Interoperability Productions

Jetzt: globalerp



Neue Funktionen & Bugfixes Beim Kunden testen und validieren

Einfach bereitstellen. Gemeinsam prüfen. Nachvollziehbar verbessern.

Mehr Dokumentation | **Höhere Softwarequalität** | **Mehr Anwenderakzeptanz**

INTERSYSTEMS DACH SYMPOSIUM 2026 · FRANKFURT

Von Release-Stress zu planbarer Stabilität

Pre-Deployment mit Git, isolierte Entwicklung und kontrollierte Testfreigaben mit InterSystems IRIS

Stephan Lienemann
Entwicklungsleitung, globalerp.de
Softwarearchitektur & Qualität

Manuel Heß
Entwicklungsleitung, globalerp.de
Prozesse & Produktstabilität

berta plus | globalerp.de

AUSGANGSLAGE

Zentral zu entwickeln wurde zum Risiko

UNSER UMFELD

Mehrere Mandanten auf einer Instanz

24/7 in Logistik, Rechnung und Fertigung

Keine kleinen Änderungen

Alles in einem Stand

Änderungen mischten sich, lange bevor sie fertig waren.

Fehlende Nachvollziehbarkeit

Was einen Fehler auslöste, war kaum zu finden.

Spätes Feedback

Oft erst vom Kunden, im schlimmsten Fall aus dem Produktivsystem.

Also sind wir dezentral geworden und haben uns damit neue Probleme eingekauft.

UNSER WEG

Ein Schritt ermöglichte den nächsten



Der Import der Änderungen aus Git wurde zur Grundlage für alles Weitere.

DAS FUNDAMENT · SCHRITTE 1–3

Zwei neue Probleme, zwei Antworten

NEUES PROBLEM

Jeder Arbeitsplatz anders eingerichtet

→ Skripte gleichen VS Code für alle an: Encoding, Konventionen, Erweiterungen, Kundenserver.

NEUES PROBLEM

Git-Stand ist noch kein IRIS-Stand

→ Der Import macht den Stand konsistent, per REST nur die Änderungen.

Embedded Git

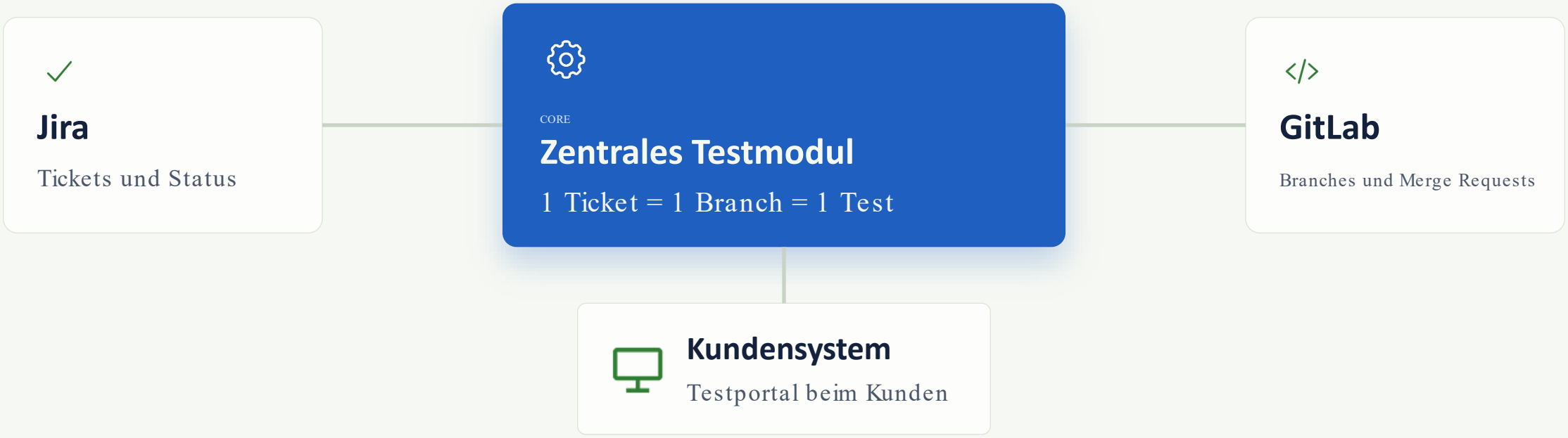
Namespace führt, Git schreibt mit

Unser Weg

Git gibt den Stand vor, IRIS folgt

DIE KONSEQUENZ · SCHRITTE 4–6

Dieselbe Technik lädt den Branch im Kundensystem



DER WORKFLOW · SCHRITT 7

Sechs Schritte vom Ticket bis zum Merge

1

Anforderung im Jira-Ticket

Jede Anforderung beginnt als Ticket, dem Single Point of Truth.

2

Änderung im Git-Branch

Entsteht direkt aus dem Ticket. Benennen muss ihn niemand.

3

Isolierte Entwicklung

Die Änderung bleibt für sich, bis sie geprüft und getestet ist.

4

Code-Review im Team

Ein zweiter Entwickler prüft den Code: die erste Verteidigungslinie.

5

Isolierter Kundentest

Kunde oder Support testen genau diesen Stand, mit klarer Anleitung.

6

Freigabe und Auto-Merge

Mit dem OK des Kunden geht die Änderung automatisch ins Release.

Test fehlgeschlagen? Das Ticket fließt mit konkreten Fehlerhinweisen automatisch an den Entwickler zurück.

PRAXISERFAHRUNGEN

Ruhigere Releases, fokussierte Kunden

DAS PROBLEM VOM ANFANG	HEUTE
Alles in einem Stand	✓ Ein Branch pro Änderung
Fehlende Nachvollziehbarkeit	✓ „1 Ticket = 1 Branch“
Spätes Feedback	✓ Abnahme vor dem Merge

Der Preis ist Disziplin: „Mal eben auf Zuruf im Release patchen“ gibt es nicht mehr.

HEUTE

KI-Agenten unterstützen uns

zum Beispiel bei **Konzeption** **Architekturreview** **Fehleranalyse**

REGELN, SKILLS UND KONTROLLEN DES AGENTEN · ÜBER DIESELBEN SKRIPTE

Änderung → **Review** → **Merge** → **Automatisch an alle Arbeitsplätze**

> **Der Prozess ist für unsere Mitarbeiter entstanden.**
>
> Heute ist er auch die Grundlage dafür, dass uns KI-Agenten sinnvoll unterstützen.

KEY TAKEAWAYS

Vier Kernbotschaften

1 **Isolation braucht Synchronisation**

Erst der Abgleich macht dezentrale Entwicklung beherrschbar.

2 **Abnahme vor dem Merge**

Isolierte Teststrecken schaffen Vertrauen.

3 **Das System führt den Prozess**

Automatisierte Schnittstellen schließen die Lücken zwischen den Werkzeugen.

4 **IRIS als starkes Bindeglied**

DevOps-Werkzeuge und ERP-Fachlogik nahtlos verknüpft

ABSCHLUSS

Prozessdisziplin schlägt Heldentum.

> Klare Prozesse und Regeln machen aus Release-Stress planbare Stabilität.

DANKE

Vielen Dank!

für Ihre Aufmerksamkeit

Stephan Lienemann
Entwicklungsleitung
Softwarearchitektur & Qualität

Manuel Heß
Entwicklungsleitung
Prozesse & Produktstabilität

berta plus | globalerp.de



Gemeinsame Prinzipien. Nachvollziehbare Releases.

Fazit aus den Praxisbeispielen von HBT und globalerp

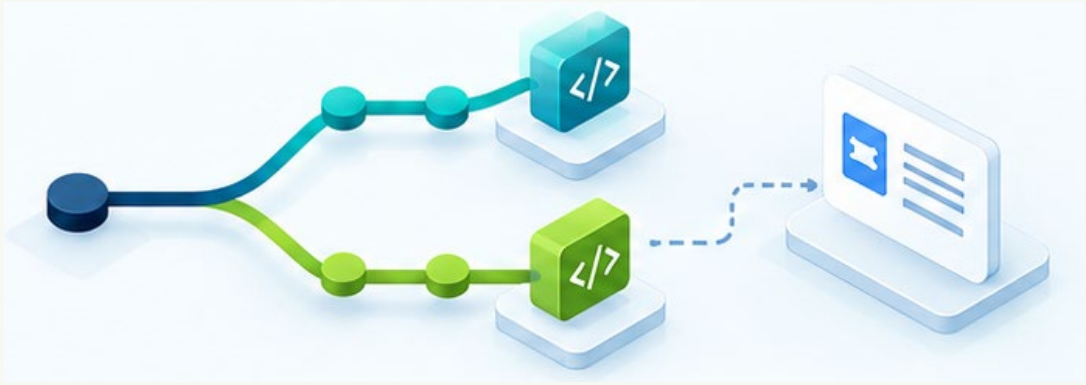
Git als gemeinsame Basis



- ✓ **De-facto-Standard in beiden Beispielen**
- ✓ **Dokumentation & Nachvollziehbarkeit auf hohem Niveau**

Unabhängig umgesetzt: dasselbe Prinzip

HBT · globalerp



- ✓ **One Feature – One Branch**
- ✓ **One Ticket – One Branch**

  bei Einbindung eines Ticketsystems

Jetzt ist Zeit für Ihre Fragen

Kurze Fragerunde mit HBT, globalerp und InterSystems