InterSystems
Creative data technology

Deployment Excellence

Wie moderne Software sicher, schnell und reproduzierbar in Produktion gelangt



Julian Seegebarth, Tobias Krause | HBT

Stephan Lienemann, Manuel Heß | globalerp

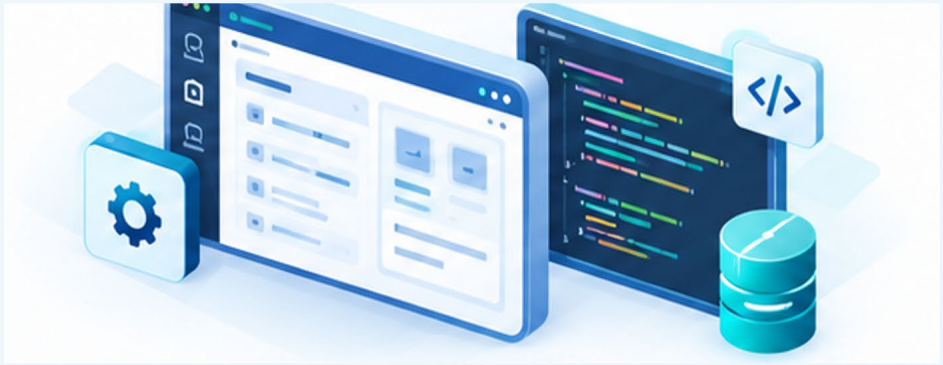
Markus Mechnich | InterSystems

Deployment-Strategien

Wie werden Änderungen sicher produktiv?

InterSystems
Creative data technology

Applications



UI & Business Functions · **Software-Bugfixes** · **Applikations-versionen**

Einzelnes Update oder komplettes Release?



Versionieren & dokumentieren

Git · Embedded Git · Eigene Integration

Interoperability Productions



Business Hosts als PTD-Dateien · **Komplette Exportpakete** · **Einzelne Schnittstellen ändern**

Gezielt deployen – kontrolliert zurückrollen?

1 **Versionierung** → 2 **Test** → 3 **Freigabe** → 4 **Deployment** → 5 **Betrieb**

↶ Rollback

HBT Kleine & große Gesundheitseinrichtungen · Industrie

globalerp ERP für Lagerhaltung, Logistik & KFZ-Teile-Handel

Vom Commit — zum Deployment.

Automatisierte Produktionsexporte aus Git – unser Weg in IRIS-Projekten.

AUSGANGSPUNKT

Git bewusst zur Quelle der Wahrheit machen.



OHNE GIT

Die Instanz verwaltet die Stände.

- Quellcode liegt in IRIS
- Export von Stage zu Stage, nie isoliert
- Stages vergleichen nur von Hand



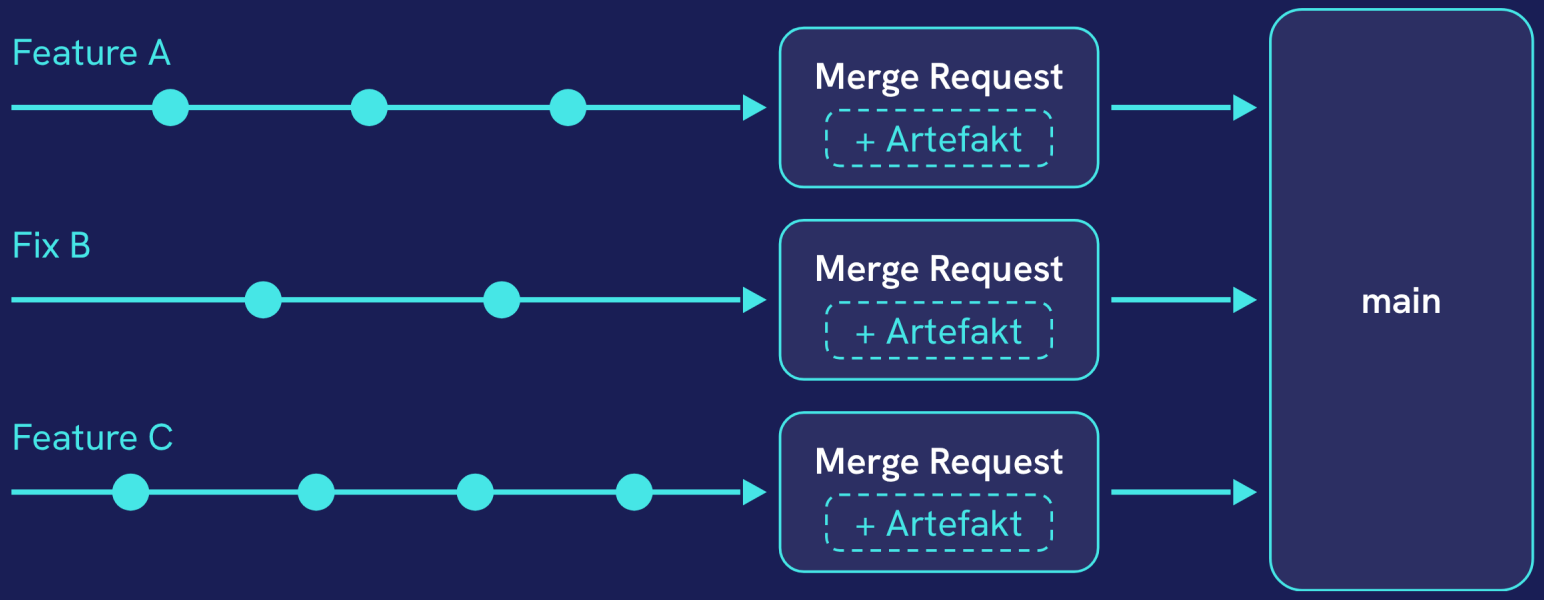
MIT GIT

Das Repository verwaltet die Stände.

- Feature-Branches, parallel und isoliert
- Historie und Diff zwischen beliebigen Ständen
- Instanz ist nur noch Laufzeitumgebung

ZIELBILD

Jedes Feature in seinem eigenen Branch.



Entkoppelt

eigener Stand, Merge Request und Artefakt

Isoliert

Unfertiges blockiert nichts und gerät nicht ins Deployment

Parallel

niemand überschreibt die Arbeit der anderen

Das Team kann wachsen, und jedes Deployment bleibt kontrollierbar.

BEIM KUNDEN

Drei Stages, ein Repository.

Bei unseren Kunden heißen die Stages häufig ENTW, KONS und PROD. Wir unterstützen sie dabei, Git einzuführen.



Vom Commit zum Deployment?

PRODUKTIONSEXPORT

IRIS bringt Export und Deployment schon mit.

SCHRITT 1 · AUF DEM QUELLSYSTEM

Export

- Deployment-Paket. Eine XML-Datei.
- Komponenten auswählen. Für ein Update nur die geänderten, bei Bedarf mit ihren Settings als PTD.



SCHRITT 2 · AUF DEM ZIELSYSTEM

Deployment im laufenden Betrieb

- Rollback-Paket. Der alte Stand als Paket.
- Gezielter Neustart. Nur betroffene Komponenten.
- Atomar. Kompilierfehler rollt alles zurück.
- Historie. Mit Log, im Portal zurückrollbar.

PRODUKTIONSEXPORT KONKRET

So sieht ein Deployment-Paket aus.

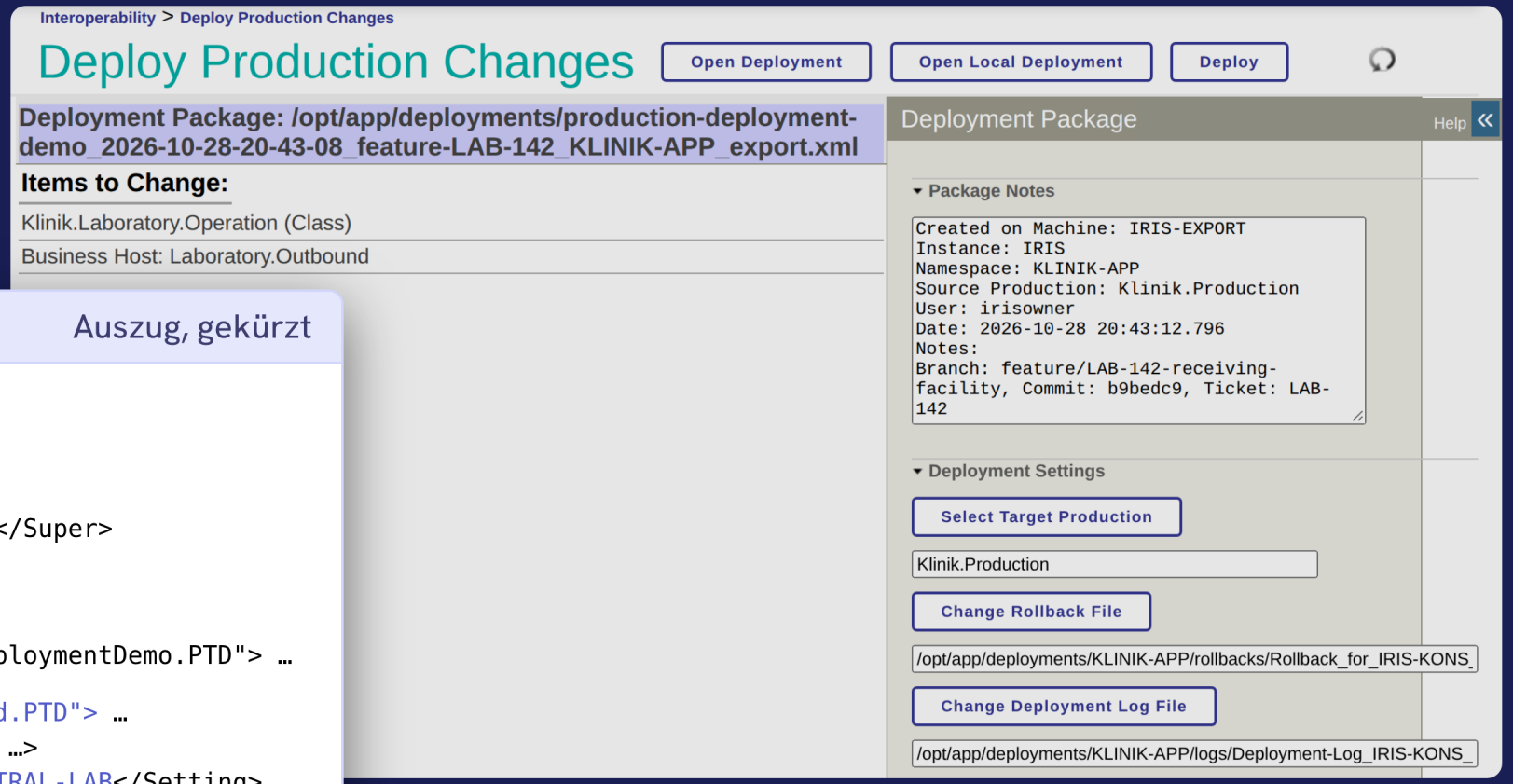
…_feature-LAB-142_KLINIK-APP_export.xml — Auszug, gekürzt

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <Export generator="IRIS" version="26" …>

13  <Class name="Klinik.Laboratory.Operation">
17    <Super>EnsLib.HL7.Operation.FileOperation</Super>
25    <Property name="ReceivingFacility"> …
45  </Class>

48  <Document name="EnsExportNotes.ProductionDeploymentDemo.PTD"> …

72  <Document name="Settings:Laboratory.Outbound.PTD"> …
73    <![CDATA[<Item Name="Laboratory.Outbound" …>
        <Setting … Name="ReceivingFacility">CENTRAL-LAB</Setting>
```

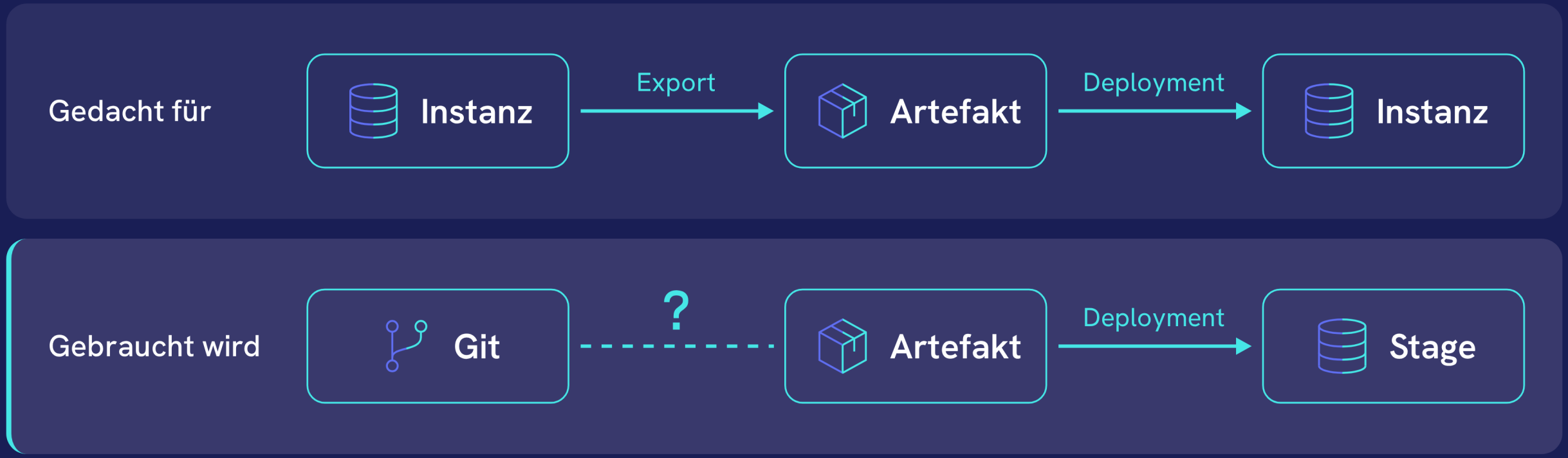


Dasselbe Paket im Management Portal, vor dem Einspielen

Oder per ObjectScript: `##class(Ens.Deployment.Deploy).DeployCode()`

DIE LÜCKE

Von Git zum Artefakt fehlt ein Schritt.



DER FEHLENDE SCHRITT

Eine CI/CD-Pipeline baut das Artefakt. Ereignisse in Git lösen sie aus.



EIN DURCHLAUF

Vom Commit zum Artefakt in sechs Stationen.

1 **Anlass**
Merge Request, Release oder von Hand

2 **Image bauen**
aus dem Dockerfile des Projekts

3 **Änderungen**
geänderte Dateien, betroffene Namespaces

4 **IRIS starten**
frische Instanz, nur aus dem Commit

5 **Produktionsexport**
je Namespace ein Artefakt

6 **Ergebnis**
Artefakte an der Pipeline, Link am Merge Request



STAND UND AUSBLICK

Was heute läuft und woran wir arbeiten.



Ausblick

Betroffene Komponenten

Aus den geänderten Dateien ableiten, was neu starten muss, und als PTD ins Paket legen.

Ausblick

Schnellerer Export

Leichtgewichtig auf Python-Basis, für große Projekte.

ZUSAMMENFASSUNG

Drei Punkte zum Mitnehmen.

01

Es gilt Git

Mit Git gilt der Quellcode im Repository, auch für das Deployment.

02

IRIS bringt den Prozess mit

Der Produktionsexport: Paket als XML, Einspielen mit Rollback.

03

Die Pipeline baut das Artefakt aus Git

Automatisch am Merge Request, beim Release oder nach Bedarf.

FAZIT

Vom Commit zum Deployment: automatisiert, reproduzierbar, auditierbar.



Kein Export mehr von Hand



Reproduzierbare Artefakte



Auditierbare Änderungen



Entkoppelt und skalierbar

HBT

4. November 2026

IT Driven Business.

Vielen Dank
für Ihre Aufmerksamkeit.

DEVOPS

Julian Seegebarth

julian.seegebarth@hbt.de

DEVELOPER

Tobias Krause

tobias.krause@hbt.de

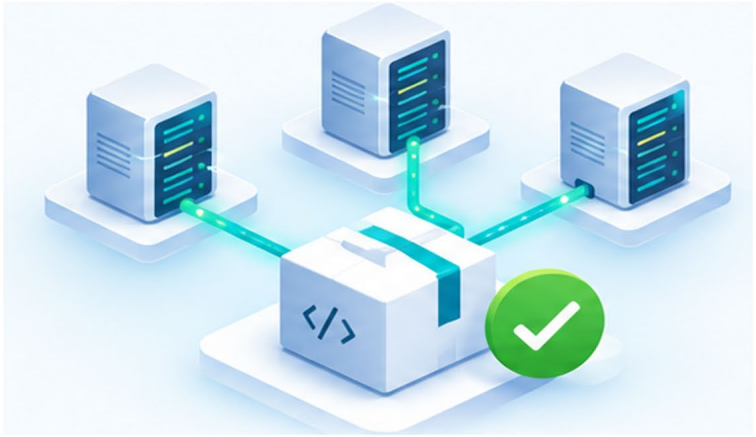


KI-generierte Illustration, erstellt mit ChatGPT

Deployment Prozesse Praxisbeispiele

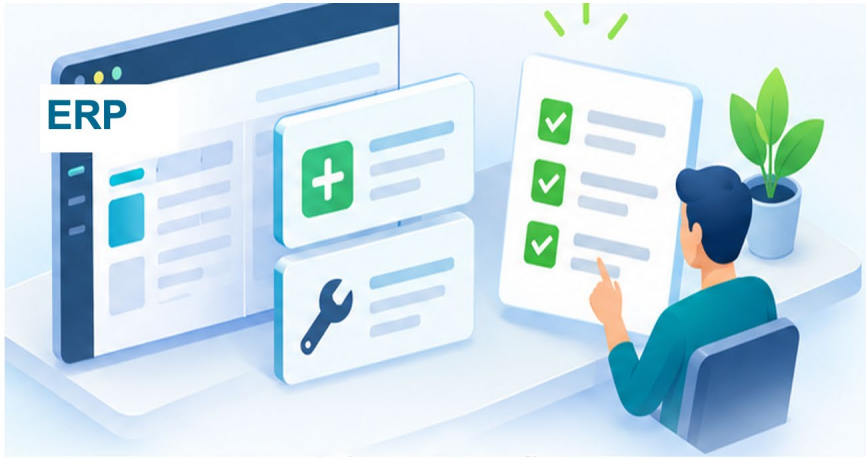
Gleiche Ziele - unterschiedliche Prozesse - unterschiedliche Anforderungen

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**