

Abnahme Workflow (Git-Workflow, CI/CD, Ticketprozess)

1. Zweck und Geltungsbereich

Diese Anlage beschreibt den verbindlichen Workflow für Entwicklung, Abnahme, Deployment und Livegang im TYPO3-Projekt **cms01** der Universität Potsdam (UP/ZIM) in Zusammenarbeit mit dem Auftragnehmer (AN). Grundlage ist der Git-Workflow inkl. CI/CD-Deployment über GitLab.

Projekt/Repository: <https://gitup.uni-potsdam.de/zim/ba/cms01>

2. Git-Workflow und Branch-Konzept

Änderungen werden grundsätzlich über **Merge Requests (MR)** umgesetzt. MRs bilden die Basis für Review, Nachvollziehbarkeit und Abnahme.

Branch-Zuordnung zu Zielumgebungen:

- **develop** → Deployment auf **Develop-Server**
- **testing** → Deployment auf **Testing-Server**
- **main** → Deployment auf **Staging** und **Production**

3. Deployment-Workflow (CI/CD über GitLab Runner)

Das Deployment erfolgt automatisiert mittels **GitLab Runner** und **CI/CD-Pipeline** (u. a. gitlab-ci.yml im Projekt).

3.1 Deployment: develop Branch

- **2 Stages** → Deployment auf **Develop-Server**

3.2 Deployment: testing Branch

- **2 Stages** → Deployment auf **Testing-Server**

3.3 Deployment: main Branch

- **3 Stages**
 - **Stage 1 & 2** → Deployment auf **Staging-Server**
 - **Stage 3** → Deployment auf **Production**
- Für Production bestehen **zwei Varianten**:
 - Deployment **mit Cache löschen**
 - Deployment **ohne Cache löschen**

3.4 UP-Deployment (Site Configuration)

Änderungen an der **Site Configuration** erfolgen durch UP mittels **Git Extension** (UP-Deployment).

4. Test- und Abnahmeumgebung

Die Abnahme und das Testen erfolgen in zwei Schritten über getrennte Umgebungen:

1. **Develop-System (Abnahmeumgebung)**

Nach Abschluss der Umsetzung und Deployment auf das Develop-System wird das Ticket zur **fachlich/technischen Abnahme durch UP** übergeben. Ziel ist die Prüfung, ob die Änderung korrekt umgesetzt ist und dem abgestimmten Umfang entspricht.

2. **Testing-System (Finaltest / Release-Vorbereitung)**

Nach erfolgter Abnahme auf dem Develop-System wird die Änderung auf das **Testing-System** übernommen und dort final getestet. Das Testing-System dient der **Release-Vorbereitung**: Alle dort verfolgten Änderungen/Features sind grundsätzlich dafür vorgesehen, zum **definierten Release-/Produktivtermin** produktiv gesetzt zu werden.

Auf dem Testing-System werden grundsätzlich nur Änderungen/Features geführt, die für den nächsten abgestimmten Release-/Produktivtermin vorgesehen sind; Ausnahmen sind vorab mit UP abzustimmen und im Ticket zu dokumentieren.

5. Ticket-Workflow (GitLab Boards)

Board: <https://gitup.uni-potsdam.de/zim/ba/cms01/-/boards/394>

Grundregel: Tickets sind der Person zuzuweisen, die das Ticket erstellt hat (Assignee).

5.1 Status-Spalten und Verantwortlichkeiten

Open (Backlog)

- UP erstellt Tickets in der Regel und legt sie zunächst in **Open** ab.
- AN kann Tickets ebenfalls in **Open** erstellen und ablegen; hierzu erfolgt ein Hinweis an UP (z. B. im Meeting oder per E-Mail).
- Tickets werden vor Bearbeitungsbeginn kurz abgestimmt; danach Wechsel nach **To Do**.

To Do (nächste Bearbeitung)

- UP verschiebt Tickets nach **To Do**, die als nächstes bearbeitet werden sollen.
- Zuständigkeit wird über Labels gekennzeichnet (siehe 5.2).
- Sobald die Arbeit beginnt, verschiebt die zuständige Partei das Ticket nach **Doing**.

Doing (in Bearbeitung)

- Ticket ist aktiv in Arbeit.
- Nach Fertigstellung oder bei Rückfragen wird das Ticket nach **Rückfragen an UP/Abnahme von UP** verschoben.

Blocked

- Ticket ist absehbar blockiert (z. B. Rückmeldung aus Datenschutz/Barrierefreiheit oder anderen Bereichen erforderlich).
- Blockaden werden durch UP koordiniert; Status/Grund sind im Ticket zu dokumentieren.

Rückfragen an UP/Abnahme von UP

- Rückfragen an UP oder Abnahme.
- Wenn Nacharbeiten erforderlich sind, wird das Ticket zurück nach **Doing** verschoben (an zuständige Partei).

Fertig zur Übernahme

- UP verschiebt abgenommene Tickets nach **Fertig zur Übernahme**.
- Wenn die Änderungen zu einem Ticket in den Main-Branch übernommen wurden: zusätzlich Label „im Masterbranch“

Verschobenes

- Bearbeitung wird auf unbestimmte Zeit ausgesetzt.
- Ticket kann nach **Open** zurück und erhält Label „**Verschoben**“.
- UP verschiebt Tickets entsprechend.

Abgewiesenes

- Ticket wird nach **Closed** verschoben und erhält Label „**Abgewiesen**“.
- UP verschiebt abgewiesene Tickets.

Closed

- Final abgeschlossene Tickets werden durch UP nach **Closed** verschoben.

5.2 Labels (Mindeststandard)

- **Feature**
- **Fehler**
- **Site-Configuration**
- **TYPO3-Core**
- **Agentur**
- **UP**
- **Verschoben**
- **im Masterbranch**
- optional weitere nach Bedarf

Labels „**Agentur**“ und „**UP**“ kennzeichnen primär die Zuständigkeit.

6. Abnahme-Workflow (operativ)

Ein Ticket gilt als „bereit zur Abnahme“, wenn:

- Umsetzung abgeschlossen ist,
- Deployment auf das **Develop-System** erfolgt ist,
- und das Ticket nach „**Rückfragen an UP/Abnahme von UP**“ verschoben wurde.

Nach Abnahme durch UP wird die Änderung auf das **Testing-System** übernommen und dort final geprüft.

UP bestätigt die finale Freigabe über den Board-Status (z. B. „**Fertig zur Übernahme**“; ggf. zusätzlich Label „im Masterbranch“, sofern bereits gemerkt).

7. Definition of Done (DoD)

Ein Ticket gilt erst dann als **fertig** und **zur Abnahme bereit**, wenn alle folgenden Kriterien erfüllt sind:

7.1 Dokumentation im Ticket

- Der **zugehörige Merge Request (MR)** ist im Ticket verlinkt (alternativ Commit-Referenz, falls erforderlich).
- Kurze Beschreibung: **Was wurde geändert und warum?**
- Hinweis auf **betroffene Instanzen/URLs/Module** (sofern relevant).

7.2 Technische Mindestprüfungen

- Deployment auf die vereinbarte Zielumgebung ist erfolgt (Develop/Testing/Staging gemäß Branch und Prozess).
- Es liegen **keine neuen kritischen PHP-/TYPO3-Errors/Exceptions** im Zusammenhang mit der Änderung vor (stichprobenartig geprüft).
- Bei Änderungen an Konfiguration/Integrationen: kurzer Hinweis, **wie geprüft wurde** (z. B. „SSO-Flow geprüft“, „Scheduler-Lauf geprüft“).

7.3 Übergabe zur Abnahme

- Ticket ist nach „**Rückfragen an UP/Abnahme von UP**“ verschoben und enthält eine eindeutige Abnahmeaufforderung.
- Ticket ist dem/der Ticketersteller:in zugewiesen (Assignee).
- Bei potenziell riskanten Änderungen: kurzer **Rollback-/Fallback-Hinweis** (1–2 Sätze).

8. Updateprozess / Livegang von Features (Release-Fenster)

Dieser Prozess gilt für Updates/Livegänge, die ein abgestimmtes Release-Fenster erfordern. Der Vorlauf der Updateankündigung richtet sich nach Art und Schwere der Änderung.

8.1 Updateankündigung (AG) – Vorlauf je Änderungstyp

Der AG kündigt Updates abhängig von Umfang und Risiko an:

- **Große Änderungen / Migrationen / Major-Releases / strukturverändernde Maßnahmen:** Ankündigung **bis zu 14 Tage** vor dem Termin
- **Reguläre Updates:** Ankündigung **2 Tage** vor dem Termin
- **Sicherheitsupdates (kritisch):** Ankündigung **1–2 Tage** vor dem Termin bzw. **unverzüglich nach Abstimmung**, sofern eine kurzfristige Umsetzung erforderlich ist

Ab Ankündigung gilt nach Möglichkeit ein **Change-Freeze** für nicht zwingend erforderliche Änderungen; Ausnahmen sind vorab mit dem AG abzustimmen und im Ticket zu dokumentieren.

8.2 T-1 Tag

- Der AG erstellt ein **Release-Ticket**, das alle Features/Änderungen enthält, die zum Termin produktiv gehen sollen.

- Der AN merged die für den Termin vorgesehenen Features/Änderungen in den **main Branch**.
- Der AG testet die geplanten Änderungen auf **Staging**.

8.3 Updatetag

- Der AG erstellt einen **DB-Dump**.
- Der Livegang (Deployment Production) erfolgt **in der Regel durch den AG**.
- Der AN wird bei **größeren Updates** (z. B. Migrationen) in die Durchführung eingebunden bzw. unterstützt den Livegang nach Abstimmung.