

# Glossar

---

## Übersicht

**Projekt:** Projekt Episko

*Inkrement:* 0

**Autor:** Simon Blum

**Datum:** 11.11.2024

**Zuletzt geändert:**

*von:* Max Rodler

*am:* 02.04.2025

**Version:** 2

**Prüfer:** Ben Oeckl

**Letzte Freigabe:**

*durch:* Ben Oeckl

*am:* 02.04.2025

## Changelog

Datum	Verfasser	Kurzbeschreibung
11.11.2024	Simon Blum	Initiales Erstellen und Verfassen
02.04.2025	Max Rodler	Hinzufügen weiterer Begriffe

## Distribution List

- Simon Blum [simon21.blum@gmail.com](mailto:simon21.blum@gmail.com)
  - Ben Oeckl [ben@oeckl.com](mailto:ben@oeckl.com)
  - Maximilian Rodler [maximilianreinerrodler@gmail.com](mailto:maximilianreinerrodler@gmail.com)
  - Paul Stöckle [paul.stoeckle@t-online.de](mailto:paul.stoeckle@t-online.de)
- 

Begriff	Erklärung
Metadaten	Repräsentierung von Stammdaten eines Projektes im Zuge der Software. Eine genaue Definierung welche Daten hier beinhaltet sind erfolgt im Laufe des ersten Inkrements.
Projekt	Softwareprojekt des Nutzers welches durch Metadaten beschrieben werden kann.
Manifest (Repr.)	Repräsentierung von Metadaten in einer Datei. Die genaue Struktur wird im Laufe des ersten Inkrements festgelegt.
Inkrement	Zeitraum in dem festgelegte Arbeitspakete bearbeitet werden. Abschluss dieser kann auch als Meilenstein bezeichnet werden.

---

Begriff	Erklärung
Meilenstein	Abschluss eines Inkrementes.
Kritikalität	Im Rahmen des Projektes kann die Kritikalität verschiedener ???? folgendermaßen angegeben werden: 0 - Absolut unabdingbar, 1 - Sehr wichtig, 2 - Normal 3 - Unwichtig
Serialisieren	Umwandlung der menschenlesbaren Manifestdatei (toml) in speicherbare / nutzbare Daten.
Deserialisieren	Umwandlung gespeicherter Daten in menschenlesbare Manifestdatei (toml).
Gecached	Metadaten welche in einer Datenbank zwischengespeichert sind um das wiederholte deserialisieren von Manifest-Dateien zu vermeiden.
Lokale Daten	Metadaten welche sich in Manifest-Dateien befinden und nicht gecached sind.
Cache	Der Begriff Cache deutet im Rahmen des Projektes auf einen persistenten Cache in Form einer Datenbank hin. Diese ist nur ein “Cache” da die Manifeste die “single source of thruth” sind.
CI/CD Pipeline	Prozess zum automatischen Testen beim Deployment.
Use- Cases	Anwendungsfälle die durch die Software abgedeckt werden sollen.
Requirements	Funktionale und Nichtfunktionale Anforderungen, die die Software erfüllen muss.
Designpaper	Dokument zur Beschreibung bestimmter Elemente der Anwendung.
Config	Dateien welche, für den Programmablauf notwendige, Configurationen beinhalten.
Linting	Statische Codeanalyse, die Programmierfehler aufzeigt.
Rust- Crates	Eine, für den Rust-Compiler zusammengefasste Kompilierungseinheit, verwaltet durch den Cargo-Paketmanager.

---