

# Code Reviews

⚠ Dieser Prozess ist nicht final und kann sich jederzeit ändern oder weiterentwickeln.

## Grundprinzipien

- Code Reviews erfolgen **pro Ticket**, nicht gesammelt über mehrere Tickets hinweg
- Reviews sollen **schnell und effizient** sein (typischerweise **3-10 Minuten** inklusive eine Feedbackrunde)
  - Wenn Code deutlich länger braucht um verstanden zu werden, ist das ein Hinweis auf ein Problem
  - **Ausnahmen** gelten für komplexe oder kritische Logik (z. B. Algorithmen)

## Ziele

Der Fokus liegt auf dem Gesamtbild, nicht auf Detailoptimierungen.

Im Review wird geprüft, ob:

- die Lösung zum bestehenden Code und Aufbau des Systems passt
- die Struktur des Codes nachvollziehbar und sinnvoll ist
- die Umsetzung leicht verständlich ist
- die Änderung keine negativen Auswirkungen auf andere Teile des Systems hat
- die grundlegende Qualität der Lösung stimmt

Ziel ist es, schnell zu beurteilen, ob die Lösung insgesamt sinnvoll und wartbar ist.

## Ablauf

### Vorbereitung

Vor dem Erstellen eines Commits:

- Sicherstellen, dass der Code funktioniert
- Relevante Tests ausführen
- Änderungen klar und nachvollziehbar halten
- Kurz beschreiben:

- Was wurde geändert?
- Warum wurde es geändert?

## Review

Im Review wird vor allem geprüft:

- Ist die Struktur des Codes verständlich?
- Ist die Lösung im Gesamtsystem sinnvoll?
- CI/CD Tests können hier übersprungen werden

Diskussionen konzentrieren sich auf:

- **Sicherheit**
- **Robustheit**
- **Performance**
- **Einfachheit**

## Ergebnis

Ein Review gilt als nicht bestanden, wenn:

- der Code schwer verständlich ist
- die Struktur unklar ist
- Designentscheidungen nicht nachvollziehbar sind

In diesem Fall sind Vereinfachung oder bessere Struktur erforderlich.

---

# Präventive Code Reviews

Bei Unsicherheiten bezüglich Ansatz oder Design sollte frühzeitig Feedback eingeholt werden, bevor mit der Implementierung begonnen wird.

---

# Best Practices

## Autoren

- Code einfach und verständlich halten
- Unnötige Komplexität vermeiden
- Änderungen klar pro Ticket trennen
- Vor ein Merge Request sollte der main-Branch im Ticket-Branch gemergt werden

## Reviewer

- Fokus auf Architektur und Gesamtbild
  - Keine Diskussionsbeiträge ohne Mehrwert
  - Feedback klar und begründet formulieren
  - Feedback verfolgt zwei zentrale Ziele:
    1. Verbesserung der Code-Qualität
    2. Wissenstransfer und Weiterentwicklung im Team
- 

## Fazit

Code Reviews sind ein schneller, fokussierter Prozess zur Sicherstellung von verständlichem und gut integriertem Code.

---

Revision #14

Created 2026-03-30 10:07:54 CEST by Nikolay Hadzhiev

Updated 2026-06-26 11:14:04 CEST by Nikolay Hadzhiev