

# Spickzettel: GitHub Issues effizient nutzen (im Projektworkflow)

## Ziel

GitHub Issues sinnvoll in Projektabläufe einbetten – für Planung, Umsetzung, Review und Transparenz im Team.

---

## Workflow-Einbindung

### Projektstart

- Aufgaben sammeln → als Issues anlegen (strukturierte Templates nutzen)
- Aufteilen in kleinere Teil-Issues bei komplexen Themen
- Zuweisung an Meilensteine, Labels und Teammitglieder

### Umsetzung

- Pro Issue ein Feature Branch erstellen:

```
git checkout -b feature/42-kamera-anbindung
```

- Branch-Namen mit Issue-Nummer verknüpfen (z. B. feature/42-...)
- PR erstellen mit Verlinkung:

```
Fixes #42 # schließt das Issue automatisch bei Merge
```

### Review & Merge

- PR wird im Issue sichtbar verlinkt
  - Labels wie in review, blocked, ready setzen
  - Nach Merge: Issue automatisch geschlossen
- 

## Automatisierung & GitHub-Integration

### Issue Templates verwenden

- `.github/ISSUE_TEMPLATE/*.md` → vordefinierte Strukturen
- Erleichtert Qualität & Vergleichbarkeit

### Automatisches Zuweisen von Labels oder Reviewer

- Über [GitHub Actions](#)

- Beispiel: Label basierend auf Dateitypen oder Branch-Namen

## Verknüpfung mit Projects & Milestones

- Issue taucht automatisch in verknüpftem Project auf
  - Fortschritt sichtbar im Sprint oder Release
- 

## Typische Branch-Namenskonventionen

Zweck	Namensmuster
Feature	feature/123-login-form
Bugfix	fix/456-api-timeout
Refactoring	refactor/789-db-layer

## Best Practices

- Issue-Beschreibung ist auch PR-Vorlage – nutze sie gut!
  - Immer Issue → Branch → PR → Merge → Issue geschlossen
  - Im Team: klare Konventionen für Labels & Branch-Namen
  - Vermeide: viele offene, verwaiste Issues
  - Nutze Projekte & Meilensteine für Sichtbarkeit
- 

Mit klarer Struktur und Automatisierung werden Issues zur **zentrale Steuerungsstelle** eines Projekts – von der Idee bis zum Review.