

# Spickzettel: GitHub Projects effektiv nutzen

## Ziel

Projekte, Aufgaben und Zusammenarbeit in GitHub effektiv abbilden – mit klaren Strukturen, Automatisierung und Verantwortlichkeit.

---

## Projektstruktur sinnvoll planen

- **Ein zentrales Project für Releases/Sprints** oder
  - **Pro Team/Modul/Komponente ein eigenes Project**
  - GitHub Projects (Beta) bevorzugen – mehr Möglichkeiten & Integration
- 

## Einträge im Project Board

- **Verwende Issues als Basis** für Aufgaben – direkt verlinkbar, kommentierbar, nachverfolgbar
  - Pull Requests werden automatisch aktualisiert (Status, Checks)
  - **Keine losen Notizen**, sondern klare ToDos mit Kontext und Ziel
- 

## Felder & Filter clever einsetzen

- Eigene Felder z. B. Typ, Priorität, Verantwortlich, Sprint, Status
  - Gruppierung nach Status, Sortierung nach Priorität
  - Filter: z. B. `status:todo priority:high`
- 

## Automatisierung nutzen

- Auto-Feld-Updates bei Issue-Zuweisung
  - PR gemerged → Status = „Erledigt“
  - Regeln definieren, um wiederkehrende Aufgaben korrekt einzuordnen
- 

## Regelmäßige Pflege

- Wöchentliche Projektdurchsicht im Team
- Neue Issues direkt im Project einordnen
- Archivierung abgeschlossener Elemente (nicht löschen)

---

## Zugriffssteuerung

- Project-Board Sichtbarkeit: Öffentlich oder privat
  - Rechte über Teams definieren: Lesen, Bearbeiten, Admin
- 

## Best Practices

- Nutze **Labels + Projects + Milestones kombiniert**
  - Halte die Felder einfach & einheitlich
  - Verwende klare, sprechende Titel für Karten (Issue-Titel)
  - Arbeite mit Templates für wiederkehrende Aufgaben
- 

Ein gut gepflegtes GitHub Project erhöht die Transparenz, reduziert Koordinationsaufwand und fördert fokussierte Teamarbeit.