Spickzettel: GitHub Labels, Milestones & Projects effizient nutzen

Ziel

GitHub-Issues und Projektarbeit strukturieren und priorisieren – für Teams, Releases und offene ToDos.

Labels – Inhalte kategorisieren

Nutzen

- Klassifizieren von Issues und Pull Requests
- Farbkodiert, filterbar, automatisierbar

Beispiele

Label Bedeutung
bug Fehler oder Regression

enhancement Neue Funktion oder Verbesserung question Klärungsbedarf oder unklarer Punkt

hw Betrifft Hardware doc Dokumentation

wontfix Wird bewusst nicht gelöst

Best Practices

- Einheitliches Label-Set für alle Repositories einer Organisation
- Farben zur Orientierung (z. B. Rot für Bugs, Grün für Features)
- Labels kombinieren für Mehrfachkategorisierung

Milestones – Zeitliche Planung & Releases

Nutzen

- Zieltermine, Etappenziele, Versionen
- Bündelt Issues & PRs zu einem "Zielzustand"

Anwendung

- "v1.0 Release" = Meilenstein
- Übersicht: Fortschritt als Prozentbalken

• Filterbar über Boards, PRs, Issues

Best Practices

- Pro Release oder Sprint ein Milestone
- Regelmäßig aktualisieren / pflegen
- Deadline optional setzen für Übersicht

GitHub Projects – Kanban & Aufgabenübersicht

Nutzen

- Visualisierung von ToDos & Workflows (Kanban, Tabellen, Zeitachsen)
- Integration von Issues, Pull Requests, Notizen

Typische Spalten

SpalteInhaltToDoGeplante AufgabenInProgressIn BearbeitungReviewZur Prüfung bereitDoneAbgeschlossen

Arten

- Classic Projects (repo-gebunden)
- Projects (Beta / GitHub Projects v2): organisationsweit, flexibler

Best Practices

- Automatisierung aktivieren (PR merged → "Done")
- Ein Project pro Release, Sprint oder Team
- Filter & Gruppen nutzen für Struktur
- Projekte regelmäßig pflegen (z. B. beim Team-Meeting)

Kombinierte Nutzung

- Labels = Was? (Inhalt)
- Milestones = Wann? (Zeitplan)
- Projects = Wo im Workflow? (Prozessstatus)

Ein gut strukturiertes GitHub-Board mit Milestones und Labels hilft bei Transparenz, Koordination und Fortschrittsmessung – egal ob Solo oder im Team.