

1) Grundprinzip: Alles ist ein Ticket



Ticket-Workflow • 1080×1080 • 

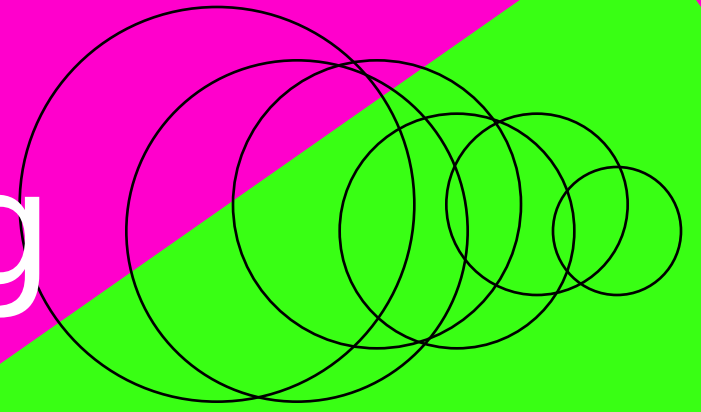
- End-to-End Traceability: Requirement → Ticket → MR → Release.
- Auch spontane Ideen oder Stakeholder-Anfragen: erst Ticket, dann Arbeit.

2) Ticket-Inhalt & Qualität

Ticket-Workflow • 1080×1080 • 

- Pflicht: Beschreibung, Anforderungen, Erwartung, techn. Details.
- Tickets mit unklaren Anforderungen werden nicht gestartet.

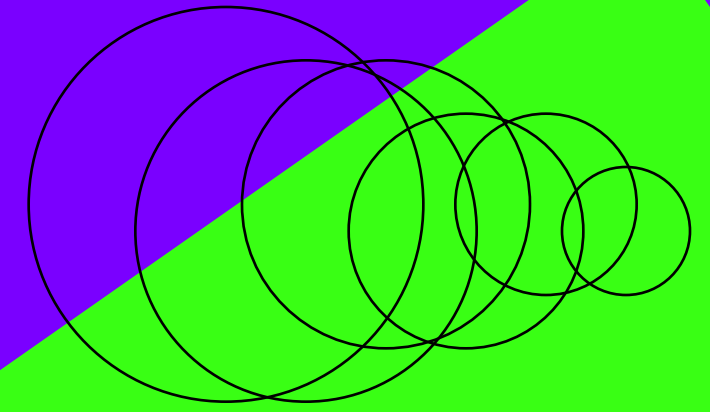
3) Aufwand & Planung



Ticket-Workflow • 1080×1080 • 

- Jedes Ticket: 3-Punkt-Aufwandsschätzung + Due Date.
- Diese Daten ermöglichen realistische Planung in Sprints/Milestones.

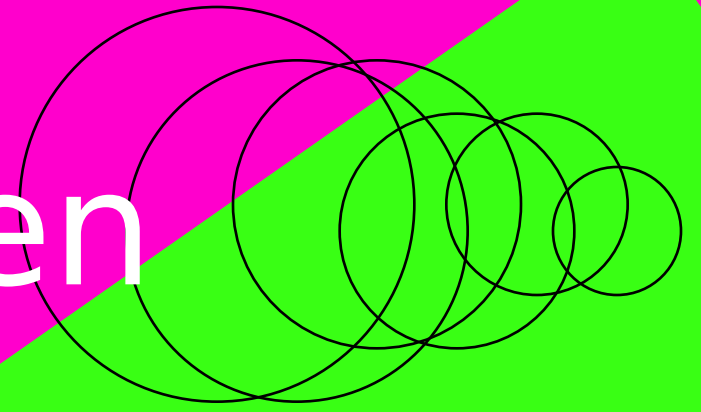
4) Ticket-Pflege



Ticket-Workflow • 1080×1080 • 

- Änderungen dokumentieren: Kommentare, neue Infos → Ticket-Update.
- Scope-Änderungen = neue Schätzung (nicht „still“ weitermachen).

5) Verantwortlichkeiten



Ticket-Workflow • 1080×1080 • 

- Jedes Ticket hat eine*n klare*n Verantwortliche*n.
- Dev-Team trägt die Durchführungsverantwortung, Stakeholder geben Input.

6) Milestones als Container

Ticket-Workflow • 1080×1080 • 

- Tickets werden Milestones zugeordnet (Start- & Enddatum).
- Scope eines Milestones bleibt fix während des Sprints; Änderungen nur im Notfall.

7) Umgang mit Überlauf

Ticket-Workflow • 1080×1080 • 

- Nicht fertig? Prüfen: Restsplit in neues Ticket (Ausnahme) oder verschieben.
- Ziel: Planbarkeit statt Überraschungen.

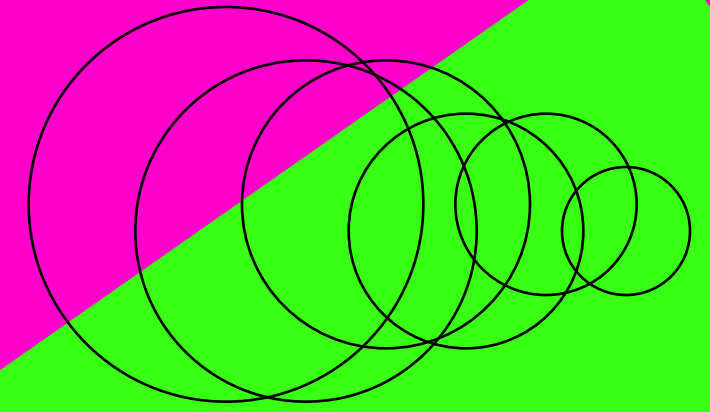
8) Merge Requests & Tickets



Ticket-Workflow • 1080×1080 • 

- MR immer mit Ticket verknüpft („Closes #123“).
- Optimal: 1 MR pro Ticket, MR-Text referenziert Ticketnummer.

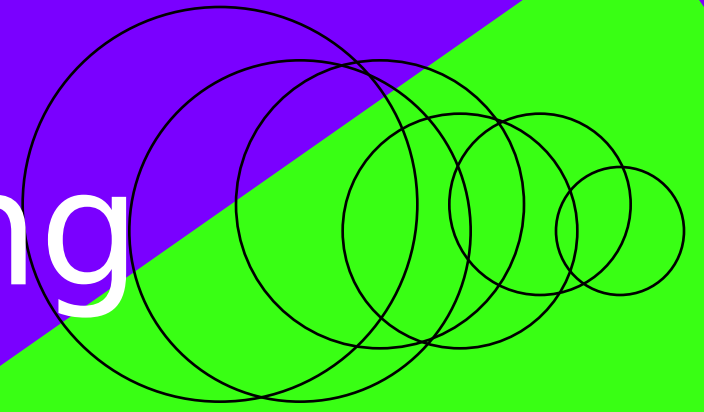
9) Ticket-Abschluss



Ticket-Workflow • 1080×1080 • 

- Schließen erst, wenn alle Anforderungen erfüllt.
- DoD: Requirements erfüllt, Tests/Docs ok, MR gemerged.

10) Effizienz & Taktung



Ticket-Workflow • 1080×1080 • 

- Milestones ≥ 2 Wochen – kürzere bedeuten Overhead.
- Fokus auf Planbarkeit, Sichtbarkeit, Zuverlässigkeit für Stakeholder.