**Workflow – CycleGraph utviklingsflyt**

**Roller**

* **Du**: Starter sprint (S1–S13), kjører Copilot med oppstartspakke, limer sluttrapport til ChatGPT.
* **Copilot**: Kode, tester, feilsøking – stegvis — leverer sluttrapport.
* **ChatGPT**: Plan/roadmap, oppdaterer Masterplan, Dynamisk DoD & Backlog, skriver kort sluttrapport (logg), kvalitetssjekker.

**Standard arbeidsflyt (ende-til-ende)**

**1) Du → Copilot (én ny chat per sprint)**

* Start *ny Copilot-chat* for sprinten (Sx).
* Lim inn **Oppstartspakke** (mal under).
* Copilot lager kjøreplan → kjør stegvis sammen.

**Oppstartspakke (mal):**

KJØRING: Du skal nå implementere Sprint S1 – Datagrunnlag & modus i CycleGraph.

Prosjekt: CycleGraph – Rust (PyO3) + Python bindings

Struktur (kort):

- core/src/lib.rs → Rust-funksjoner (PyO3)

- metrics.rs → beregninger (NP, IF, VI, Pa:Hr, W/beat, Watt-engine)

- cli/analyze.py → CLI

- tests/ → pytest + golden (Rust)

- .github/workflows/ci.yml → CI

DoD-utdrag for denne sprinten (S1):

- Strava OAuth & activity fetch (fields/streams mappes).

- Auto-deteksjon indoor/outdoor (trainer, sport\_type, device\_watts).

- CLI-flag: --mode roller|outdoor (rute til riktig pipeline).

- Fetch av én aktivitet med streams OK.

- Indoor/outdoor korrekt i logg + JSON.

- --mode påvirker kjørevei.

Status: Ferdig til og med M7.5 (analyse, CGS v1, backend-forfining, tester, CI).

Neste sprint: S1 – Datagrunnlag & modus

Arbeidsmåte:

1) Lag en kjøreplan (filer, tester, forventet output).

2) Følg planen steg for steg sammen med meg.

**Avslutningskommando (mal):**

LAG EN SLUTTRAPPORT med følgende format:

✅ Sprint: [ID – Navn]

Branch: [navn]

Commits: [hash – message]

Endrede filer: [paths]

Tester: [pytest + cargo test resultater]

Observasjoner: [korte punkter]

Status: [Ferdig / Delvis ferdig / Feil gjenstår]

**2) Du → ChatGPT**

* Lim inn **sluttrapporten fra Copilot** i denne chatten.

**3) ChatGPT (automatikk)**

Oppgave til deg:

1) Skriv en KORT sluttrapport for logs/ (bruk standardformat).

2) Foreslå konkrete endringer i DoD & Backlog (flytt sprint til ✅ Ferdig, noter ev. funn til Frontlog).

3) Synk Masterplan (oppdater statuslinje / ny rad hvis relevant).

1. 4. Før Push etter en Sprint avslutt alltid med pytest -q og cargo test --tests -q grønne lokalt **og** i CI.
2. Idempotens: kjør samme analyse to ganger; diff skal være tom (filtrer tidsstempler).
3. Golden/determinisme: toleranse oppfylt (±1–2 W der definert); ingen flaky tester.
4. Docs/DoD/Masterplan synket; åpne en SxB-mini­sprint hvis noe gjenstår.

(For sprinter som berører Strava-publisering: verifiser at tests/test\_strava\_client.py er grønn.)

5) Comit push

6) Oppdater Log

**Navnestandarder**

* **Branch:** feature/sprint-sx-<kort-navn> (eks: feature/sprint-s1-mode-detection)
* **Commit-prefix:** Sx: (eks: S1: add --mode roller|outdoor)
* **Sluttrapport-filnavn:** YYYY-MM-DD\_sprint-Sx\_<kort-navn>.md

**Eksempel (din S1 er gjort)**

Neste sprint blir S2. Kopiér oppstartsmalen, sett:

* Sprint S2 – Vær & profiler
* Lim inn S2-DoD-utdraget (vær-API, profilsettings, caching, validering).