**Sprint 6 – CLI/Reports & observabilitet (📈)**

**Kontekstblokk**

* Fra S3 (Fysikkmotor): deterministisk output (±1–2W), smoothing og NP/avg etablert.
* Fra S4 (Kalibrering): CdA/Crr-fit med MAE ≤10 %, profil lagret i JSON.
* Fra S5 (Indoor/outdoor + GPS/Wind): pipeline ferdig, CLI-output har wind\_rel og calibrated.
* Avhengighet: Sprint 6 bygger på at core/CLI allerede gir utvind-relaterte felter.
* Neste steg (Frontlog): QA & hardening (S7).

**Sprintbeskrivelse**

**Oppgaver:**

1. Rapportfelt: NP, Avg, VI, Pa:Hr, W/beat, PrecisionWatt ±usikkerhet.
2. Strukturerte logger (nivå, tidsbruk, cache hits).
3. Metrics for no-watt (sessions\_no\_power\_total).
4. Docs: “How it works” + “Known limits”.

**DoD (må alle bestås):**

* python -m cli.analyze … genererer deterministisk rapport.
* Rapport inkluderer NP, Avg, VI, Pa:Hr, W/beat, PrecisionWatt ±usikkerhet.
* Strukturerte logger på plass, loggnivå styrbart via flagg/env.
* Metrics sessions\_no\_power\_total logges korrekt.
* Docs oppdatert med “How it works” og “Known limits”.

**Estimert tid:** 6–8h

**Arbeidsmetode (for Copilot)**

1. **Lag plan (5–7 trinn):**
   * Hvilke filer, hvorfor, forventet output, risiko. Vent på “OK PLAN”.
2. **Kjør trinnvis:**
   * Diff + tester + CLI-output. Vent på “OK <trinn>”.
3. **Stop-regler:**
   * Stopp ved >3 filer endret, public API-endring eller nye crates.
4. **Debug-partner:**
   * Bruk ChatGPT ved feil eller uklarhet.
5. **Ferdigkriterium:**
   * Alle DoD verifisert med tester og CLI.

**Svarformat (Copilot):**

* PLAN → vent på “OK PLAN”.
* TRINN 1 → vent på “OK 1”.
* TRINN 2 … til ferdig.

**Sprint-delta (mini-kontekst)**

* S5 leverte GPS- og vindkorrigert watt (wind\_rel, v\_rel).
* S6 fokuserer på observabilitet og rapportering, ikke fysikkmotor.
* S6 er dermed en “rapport/QA-sprint” med lav risiko, men viktig for frontend i M8.

**Tekniske rammer (kontrakt)**

* **Input:** samples (time, hr, watts, gps/wind når tilgjengelig), profile, weather.
* **Output:** rapportfelter NP, Avg, VI, Pa:Hr, W/beat, PrecisionWatt ±usikkerhet.
* **CLI:** loggnivå styrbart; metrics logges (sessions\_no\_power\_total).
* **Docs:** oppdatert med bruksbeskrivelse og kjente begrensninger.

**Endringsgrenser**

* Tillatt: core/src/metrics.rs, core/src/lib.rs, cli/session.py, cli/analyze.py, tests/\*\*, docs/.
* Ikke rør: fysikkmotor (physics.rs), Strava-API, kalibreringsmodul.

**Tester / kommandoer**

cargo test -q

pytest -q

python -m cli.analyze session --input tests/test\_golden\_segment.csv --format json --dry-run --debug

**Verifikasjonsplan (DoD → test)**

1. Rapportfelt vises i CLI-output (NP, Avg, VI, Pa:Hr, W/beat, PrecisionWatt).
2. Samme input gir samme rapport (deterministisk).
3. Strukturerte logger genereres (cache hits, tidsbruk).
4. Metrics sessions\_no\_power\_total øker korrekt for no-watt-økter.
5. Docs viser oppdatert “How it works” og “Known limits”.
6. pytest og cargo test grønne.

**Forventet output**

* core/src/metrics.rs – PrecisionWatt ±usikkerhet.
* cli/session.py – rapportfelter, metrics og loggnivå.
* cli/analyze.py – CLI flagg/env for loggnivå.
* tests/test\_reports.py – tester på rapportfelter.
* docs/how\_it\_works.md, docs/known\_limits.md – dokumentasjon.

**Frontlog (kommende oppgaver)**

* **S7 – QA & hardening (🛡️)**: edge-cases, golden på ekte ride uten powermeter, testsetup-refaktor.
* **M8 – Demo & frontend (🧩)** starter med scaffold & dataadapter.

Lag en Sluttrapport etter 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]

**Sluttføring av Workflow (ChatGPT v2)**

1. **Sluttrapport:** lim inn Copilot-rapport i template.
2. **Delta-sammendrag:** skriv 2–3 linjer etter komprimeringsmal:
   * Hva ble gjort.
   * Teststatus.
   * Observasjon.
3. **Evaluering & Oppdatering:** oppdater DoD og Frontlog basert på nye funn/innsikter.
   * Klassifiser funn som A (DoD), B (Frontlog), C (Observasjon).
   * Marker ferdige sprinter, flytt avhengigheter.
   * Juster Sprintplan om nødvendig.
4. **Synk Masterplan:** oppdater statuslinje eller legg til ny rad.
5. **Oppdater Sprint Log:** kort notat med status og carry-over.
6. **Template neste sprint:** sjekk om “halvfast default” trenger justering og fyll ut template basert på neste sprint.