**Sprint 7 – QA & hardening (🛡️)**

**Kontekstblokk**

* Fra S5: Indoor/outdoor + GPS/Wind stabil, CLI-felter (wind\_rel, v\_rel, calibrated, status) på plass.
* Fra S6: Rapport/observabilitet levert; strukturert JSON-logging og metrikker fungerer.
* Dynamisk DoD oppdatert med nye A-krav: **JSON-schema versjonering**, **datasettkrav (≥30 samples)**, **CLI-flagg-dok**.
* Avhengigheter: Ingen nye tredjepartstjenester; bygger på eksisterende CLI/API og testoppsett.
* Neste steg (Frontlog): M8/S8 scaffold + dataadapter.

**Sprintbeskrivelse**

**Oppgaver:**

1. Edge-cases: manglende vær, GPS-drift, null HR, kort økt.
2. Flere golden-tester (ekte rides + variasjon i vær/terreng).
3. Golden på økter **uten powermeter** + plausibilitet mot HR/trend.
4. Oppdatér **CGS v1.1-kobling** (forbruker nye felter uten regress).
5. 🆕 Refaktorér testsetup m/ helpers (tests/conftest.py, tests/test\_utils.py|rs).
6. 🆕 Innfør **JSON-schema versjonering** for CLI/API-output (schema\_version, semver).
7. 🆕 Sikre at **alle testdatasett har ≥30 samples** (både indoor og outdoor varianter).

**DoD (må alle bestås):**

* pytest + cargo test grønne.
* CGS konsumerer nye felter uten regress (eksisterende CGS-tester passerer).
* CLI/API-output inneholder schema\_version (semver) **og valideres i test**.
* Alle golden-tester oppfyller datasettkravet (≥30 samples); HR-only golden har plausibilitetssjekker (ingen NaN/negativ, Pa:Hr rimelig intervall).

**Estimert tid:** 10–14h

**Arbeidsmetode (for Copilot) – “halvfast default” (justert)**

1. **PLAN (5–7 trinn)**

* Hvilke filer, hvorfor, forventet output, risikopunkter. Vent på “OK PLAN”.

1. **Kjør trinnvis**

* Hvert trinn: *diff → kjør tester → vis CLI-output*. Vent på “OK <trinn>”.

1. **Stop-regler (justert for QA-sprint):**

* **Stopp** hvis: public API-semantikk endres **uten** schema\_version bump, eller fysikkmotor/modell endres.
* **Tillatt >5 filer** når ≥60 % er tester/docs/datasett. Hvis >3 kodefiler i core/ → stopp og avklar.
* Ingen nye crates; kun dev-avhengigheter ved behov (test/verktøy).

1. **Schema-bump sjekkliste (ny):**

* Øk schema\_version (semver).
* Opprett/oppdater docs/schema/session\_vX.Y.json (+ kort docs/schema.md).
* Legg til tests/test\_schema.py som validerer felt og semver-mønster.

1. **Debug-partner**

* Bruk ChatGPT ved uklare feilmeldinger eller grensevalg i tester.

1. **Ferdigkriterium**

* Alle DoD verifisert i tester + manuell CLI-kjøring.

**Svarformat (Copilot):**

* PLAN → vent på “OK PLAN”.
* TRINN 1 … → vent på “OK 1”, “OK 2”, osv.

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

* S7 er en **QA/hardening-sprint**: hovedsakelig tester, helper-refaktor, schema-kontrakt, og CGS-stabilitet.
* Ingen endring i fysikkmotor; fokus på robusthet og kontrakter (schema/datasett).

**Tekniske rammer (kontrakt)**

* **Input:** sessions (med/uten watts), HR, ev. GPS/wind, profile, weather.
* **Output:** CLI/JSON m/ rapportfelter + schema\_version (semver: X.Y.Z).
* **CLI:** eksisterende flagg; logging styrt via --log-level/LOG\_LEVEL.
* **CGS:** må lese felter fra S6 uten å feile; ingen endring i beregningslogikk i denne sprinten.

**Endringsgrenser**

* **Tillatt:** tests/\*\*, tests/conftest.py, tests/test\_utils.\*, tests/test\_schema.py, tests/test\_golden\*.py|rs, cli/analyze.py, cli/session.py, cli/session\_api.py, docs/schema.\*, docs/how\_it\_works.md, datasett under tests/data/\*\*.
* **Ikke rør:** core/src/physics.rs, kalibreringsalgoritmer (unntak: små felt-koblinger), Strava-klient (funksjonelt uendret).

**Tester / kommandoer**

cargo test -q

pytest -q

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

# Valider schema\_version tilstede og semver:

python -m cli.analyze session --input tests/strava\_segment.csv --format json | python -c "import sys,json,re; d=json.load(sys.stdin); assert re.match(r'^\d+\.\d+\.\d+$', d.get('schema\_version',''))"

**Verifikasjonsplan (DoD → test)**

1. **Schema-kontrakt:** tests/test\_schema.py sjekker schema\_version (semver) og påkrevde felter tilstede.
2. **Datasettkrav:** tests/test\_golden\_dataset\_size.py verifiserer ≥30 samples for hver golden (indoor/outdoor/HR-only).
3. **HR-only plausibilitet:** tester for Pa:Hr innen rimelig intervall, ingen NaN/inf/negativ i metrikker.
4. **CGS regressfri:** eksisterende CGS-tester passerer uendret.
5. **Edge-cases:** egne tester for manglende vær, GPS-drift, null HR, kort økt.
6. **Grønn pipeline:** pytest + cargo test grønne i CI.

**Forventet output (filer/endringer)**

* tests/conftest.py, tests/test\_utils.py (nye/helpers)
* tests/test\_schema.py (ny)
* tests/test\_golden\_hr\_only.py, tests/test\_golden\_variants.py (nye)
* tests/data/golden\_\*.csv (utvidet til ≥30 samples; indoor/outdoor/HR-only)
* docs/schema/session\_vX.Y.json, docs/schema.md (nye)
* Små justeringer: cli/analyze.py, cli/session.py, cli/session\_api.py (felt-eksponering/validering; ingen semantikkendring)

**Frontlog (kommende oppgaver)**

* **S8 – Scaffold & dataadapter (🧩):** React/Tailwind scaffold, routing, state; adapter med analyze\_session(); **CLI-flagg-dokumentasjonstabell** i docs/.
* **S9 – Økt-kort & nøkkelmetrikker (📊):** UI for NP/IF/VI/Pa:Hr/W/slag/CGS/PrecisionWatt; no-watt varsel; formatter-tester.

Gi meg så detaljerte detaljer trinn for trinn at jeg og chat gpt kan gjøre det sammen mens du styrer fremgangen. Det vil si for hvert trinn list opp hvilke filer som er involvert , hva som skal gjøres med de og forventet outptut.

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.