

From traffic to a verdict

The approach the harness takes, end to end

The pipeline

```
journeys —> capture —> normalise —> diff —> claim —> verdict
(same on      (27      (origins,    (one      (every    PASS
both sides)   artefacts) canonical  engine    failing  WARN
                                forms,    per      hunk    FAIL
                                masks)   artefact) claimed?)
```

Each difference an engine finds is a **hunk**: an artefact, a scope a claim can match, a summary and a severity (**fail** gates unless claimed; **info** never gates).

Few journeys, wide capture

Seven journeys in four sets: a student reading and searching the fixture course, the catalogue and live apps loading, a student signing in, the published reference course, and a replay of fixed course URLs.

At **every step** the harness captures:

Per page	Per side	From the images	From the containers
DOM, screenshot, network, console, headers, axe, focus order, timing	metrics, logs, persistence writes, bus	manifest, SBOM, vulnerabilities	runtime posture, startup

Power comes from breadth of capture, not the number of journeys.

Never silent

- A collector that cannot run says so, in one shape:
`NOT COLLECTED: <what> of <app> on side b: <reason>`
- "Not collected" is never reported as "no difference".
- The `runtime` and `startup` artefacts are always required; the image artefacts become required in the nightly and release jobs.

Gate, score, glance

1. **Gate**: the verdict. Decides the exit code.
2. **Release Confidence Score**: eight weighted dimensions (claim coverage, noise health, statistical margin, rehearsals, test signal, requirements traceability, change risk, post-deploy history). One hollow dimension caps the score at 74.
Green 90+ ship · **Amber** 75 to 89 ship once the glance is verified · **Red** below 75 hold
3. **Glance**: at most seven places to look, ranked by novelty × exposure. The Reviewer marks each *verified*, *disputed* or *escalated*.

Standard work

- The monorepo's `release/SOP.md`: twelve steps, three roles.
 - the **Captain** runs the release and owns go or no-go;
 - the **Reviewer** owns the glance and authored no PR in the release;
 - the **Contributors** own their claims and EARS Rules.
- Steps 2 to 7 are one command that asks nothing and always leaves a report:

```
pnpm harness release --candidate 17.1.0-rc.1 --baseline prod
```

The harness tests itself

- **Mutants**: sixteen planted regressions (a dropped header, a route that returns 500, a write by an anonymous student, a vulnerable package, ...). Every one must be caught, weekly and on any engine change.
- **Nightly A/A**: the harness must be quiet against production itself.
- **Kaizen**: a FAIL the team wanted to ship, a Red band, a rollback or a run rule opens a **5 Whys**. It ends in one of seven countermeasures to the system: a mutant, a journey, a mask review, an EARS spec, claim guidance, an SOP change or a glance rule.

Takeaways

- One pipeline, the same for every mode: capture everything, normalise noise, diff, claim, judge.
- The Gate is binary; the score and the glance help a human decide where to look.
- Every escape becomes a permanent self-test, so the harness gets stricter, not looser.