One CI Result Is Useful But Not Always Enough

Reproducibility of install, build, and test outcomes in Dependabot npm update PRs

4.0% of analyzed PRs

show detected non-reproducibility. Most npm update evidence is stable, but single-run outcomes deserve caution in risk-prone settings.

3,083	2,777	2,477	3,142	9,426
Dependabot candidates	analyzed npm update PRs	repositories	experiments	Dockerized runs

1. PROBLEM

CI as dependency-update evidence

Dependabot PRs help projects keep dependencies secure and compatible. This study asks whether evidence behind merge decisions stays stable when the same dependency update is executed again.

2. RELATED WORK

Why the gap exists

- Automated update PRs depend on CI evidence to support merge decisions.
- CI and flaky-test studies show build/test evidence can be noisy.
- BUMP preserves Java/Maven breaking updates; this study measures npm outcomes.

3. METHODOLOGY

Six-stage pipeline

Candidate dependency updates become repeated executions and exported analyses.

S1 Normalize	S2 Qualify	S3 Plan
S4 Execute	S5 Analyze	S6 Export

4. RESPONSIBLE RESEARCH

Evidence is derived, checked, and shareable

- Pipeline logic documented and covered by 450+ tests.
- Numerical claims checked against exported analysis data.
- Public dataset focuses on reproducible research evidence, not raw logs.

5. RQ1: How stable are npm update outcomes?

Most outcomes are reproducible

Reproducibility dominates, but non-reproducibility is visible enough to matter for CI-based update decisions.

111 / 2,777

analyzed PRs show detected non-reproducibility

80.7% reproducible-like | 4.8% non-reproducible | 14.4% insufficient

6. RQ2: Where does instability appear?

Instability clusters in risky settings

Extra caution is needed when update evidence comes from projects with fragile runtime or dependency interactions.

Dominant phase among non-reproducible experiments

46.7%	19.7%	17.1%
test phase	install phase	build phase

Extra caution signals

Browser | Native | Peer deps | Yarn

7. RQ3: How well does BUMP transfer to npm?

BUMP transfers only coarsely

Broad build/test categories support comparison, but npm needs additional labels for its own mechanisms.

Failed-run category comparison

43.1%	26.0%
npm build/compile vs 42.6% in BUMP	npm test vs 32.9% in BUMP

Needed npm labels

Native compilation; browser/front-end tests; peer-dependency interaction; registry/artifact availability; npm-specific unknowns.

8. CONCLUSION

CI is useful evidence, but one run is not a verdict

Most npm update outcomes are stable, but risky settings need extra caution.

- Trust stable repeated outcomes more than isolated pass/fail results.
- Re-check updates involving tests, browsers, native addons, or peer dependencies.
- Treat CI as decision support, not as complete proof of update safety.

TUDelft

Author: Vlad Maksymiuk | v.maksymiuk@tudelft.nl

Supervisors: Sebastian Proksch | S.Proksch@tudelft.nl; Cathrine Paulsen | C.R.Paulsen@tudelft.nl

Artifact: doi.org/10.5281/zenodo.20774694