

SpecShift SPS Evaluator Public Review Packet - v0.1-sprint

Repository: <https://github.com/mrclarkonline-cyber/specshift-sps-evaluator>

Release tag: v0.1-sprint

Prepared for: public technical review, sprint judging, and initial evaluation routing

This packet is intentionally public-safe. It summarizes the released scaffold, why it matters, what it does not claim, and what a serious next-step evaluation can look like. It does not disclose protected mechanisms, private scoring logic, feature construction, weighting, bidder materials, or commercial negotiation strategy.

1. Executive Summary

SpecShift SPS Evaluator is a small, runnable prototype for generated-code evaluation. It demonstrates a practical gap: passing visible example tests is not the same as satisfying the underlying specification. The current release is a judge-readable scaffold using synthetic generated-code tasks. It normalizes tasks into audit checks across visible baselines, schema expectations, specification perturbations, and adversarial cases. Its purpose is to make specification robustness failures observable, reproducible, and easy to inspect locally.

2. What the Public Release Demonstrates

- Visible-test gap: a task can appear to pass simple examples while still missing important specification requirements.
- Structured review layer: generated-code tasks can be reviewed through multiple check types rather than one endpoint pass/fail signal.
- Runnable artifact: reviewers can clone the repo, run the demo, and inspect the dataset and output.
- Scope discipline: the repository is intentionally small and synthetic so it can be reviewed without exposing private methods.

3. Demo Command and Expected Output

Run locally:

```
python3 -m pip install -r requirements.txt
python3 runner.py --file dataset.json
```

Expected demo summary:

- Loaded 12 synthetic tasks
- Generated 67 check cases
- baseline: 20
- schema: 12
- spec_perturbation: 14
- adversarial: 21

4. What This Is Not

SpecShift SPS Evaluator is not a finished verifier, theorem prover, sandbox, exploit system, production security product, or claim of final correctness. It does not execute generated code. It does not prove safety. It does not provide a commercial scoring engine. It is a public-safe scaffold for demonstrating why endpoint test success is an incomplete evaluation signal.

5. Public-Scope Boundary

The repository intentionally excludes protected mechanisms, private scoring logic, specialized feature construction, weighting, spoof-resistance details, bidder materials, valuation language, or negotiation strategy. Public artifacts remain limited to synthetic examples, runnable scaffolding, limitations, and reviewable demonstration behavior.

6. Suggested Review Questions

- Does the README clearly distinguish visible-test success from specification satisfaction?
- Is the demo easy to run and inspect?
- Are the synthetic tasks understandable without private context?
- Are the limitations stated clearly enough to avoid overclaiming?
- Does the scaffold suggest a useful review layer for generated-code evaluation?

7. Possible Next-Step Evaluation Path

A future evaluation path could involve reviewer-selected blind audits in which reviewers retain hidden labels and independently evaluate returned discrepancy observations. This keeps the focus on observable outputs and reviewable behavior rather than internal model access, source-code transfer, or disclosure of protected implementation details.

8. Contact Surface

Repository: <https://github.com/mrclarkonline-cyber/specshift-sps-evaluator>

Release tag: v0.1-sprint

ORCID: 0009-0005-8622-0251

9. One-Sentence Summary

SpecShift SPS Evaluator is a public, runnable scaffold showing that generated-code evaluation needs to look beyond visible test passage toward specification robustness and observable discrepancy review.