For: business leaders and pilot sponsors

Evaluation Pilot: Executable Specs for One Critical Distributed Subsystem

A focused 4–6 week pilot to evaluate executable specs on one important subsystem.

Igor Konnov, Thomas Pani demo@wunderspec.com

Executive summary. In 4–6 weeks, we work with your team to evaluate executable specs on one important subsystem: model the behavior, generate scenarios or counterexamples, and connect the result to testing, review, or design workflows.

1. Evaluate whether executable specs fit your system and team

We focus on one subsystem with behavior that examples and E2E tests do not explore well: retries, timeouts, partial failures, stale reads, reconciliation loops, concurrent operations, leader changes, or unexpected message orderings.

2. When this pilot is a good fit

Strong pilot fit

- ☐ Bugs depend on ordering, timing, or concurrency
- ☐ Incidents are hard to reproduce
- ☐ E2E tests are slow or miss rare cases
- ☐ Important invariants exist only in docs, code, and tribal knowledge

Poor pilot fit

- ☐ Simple CRUD systems
- ☐ Low-risk workflows
- ☐ Mostly UI or single-node performance problems
- ☐ Teams that only need more unit test coverage

> **Good candidates:** Control planes, schedulers, databases and storage systems, consensus protocols, reconciliation loops, distributed workflows, payment or state-transition systems, stateful cloud infrastructure, and systems with important invariants.

3. What we do

1. Scope the subsystem boundary, failure modes, and pilot question
2. Identify operations, states, messages, and invariants
3. Build an executable spec
4. Explore behavior with simulation, search, or model checking
5. Generate scenarios and traces; include counterexamples when applicable
6. Connect the model to implementation tests or review workflows
7. Deliver findings and a next-step recommendation

4. What you get

- ✓ An executable spec that supports review, design, and regression testing
- ✓ Generated scenarios, traces, or counterexamples that stress meaningful behavior
- ✓ A clearer set of system invariants and modeled assumptions
- ✓ A test oracle or model-based test harness when implementation testing is included
- ✓ Reproduction attempts for known failure patterns
- ✓ Evidence about whether the model reveals new bug classes or behavioral gaps
- ✓ A findings report and next-step recommendation

5. Timeline and customer effort

A pilot is scoped to produce a clear decision in 4–6 weeks without taking over your team's roadmap.

Week 1

Scope subsystem; agree pilot question; identify operations, states, and invariants.

Weeks 2–4

Build model; explore behavior; generate traces; refine with the team.

Weeks 4–6

Connect to tests or review; document findings; decide next steps.

Customer effort

- ☐ One technical owner
- ☐ 1 kickoff session
- ☐ 1–2 technical modeling sessions
- ☐ Short review sessions
- ☐ Optional support for implementation harnessing

Inputs

- ☐ System overview or architecture sketch
- ☐ Main operations, transitions, or workflows
- ☐ Known incidents, edge cases, or failure scenarios
- ☐ Important invariants or safety properties
- ☐ Existing tests and API, CLI, or harness access if implementation testing is included

We do not need production credentials or production customer data.

---

6. Next step

Send us a short description of one system you care about.

Include five bullets:

- ☐ What kind of system it is
- ☐ What makes it hard to test
- ☐ What failures or incidents you worry about
- ☐ What tests or tools you already use
- ☐ What you want to learn from a small pilot

Email: demo@wunderspec.com

GitHub: wunderspec/wunderspec
