

Maintain test quality while keeping pace with releases

Our platform authors, verifies, and diagnoses tests with AI grounded in your applications — across Web UI and APIs.

01 / THE PROBLEM

Testing is frequently a bottleneck to release velocity

Modern web apps and configurable SaaS platforms change constantly — vendor updates, workflow edits, new automations from the business. Teams keep pace by hand-writing and re-fixing scripts, so maintenance eats the hours meant for new coverage.

Four failure patterns behind the bottleneck:

- 1 **Authoring** UI and API tests by hand is slow and skill-heavy.
- 2 **Selectors break** on every UI change, so scripts need re-fixing each release.
- 3 **Maintenance compounds** — negative and edge-case coverage slips first.
- 4 **High release velocity** overloads the testing function.

60%

reduction in regression-testing time across a client's CRM system.

Achieved without additional QA headcount.

Source: TechWish platform deployment at a leading Smart-home Solutions company

02 / THE PLATFORM

TechWish built an AI-Augmented Test Automation Platform

TechWish's proprietary AI-based platform delivers intelligent, self-verifying test automation across the full lifecycle — from authoring and execution through final assurance of code quality. It operates inside your real system, not against a general idea of how an application should behave, which is what makes AI-generated tests trustworthy rather than brittle.



Grounded authoring

Tests written from AI analysis of what's actually on the page.

Plain English · live Web UI · OpenAPI spec



Verified at creation

Generated cases are proven to pass, via self-correction.

Executes before save · multi-pass self-correction



Enriched business context

Tests are created from real-world stories within ADO / Jira.

Jira & Azure DevOps · results pushed back



Traceable & enterprise-ready

Coverage that ties back to commitments, in your own cloud.

Root cause analysis · data stays in your own cloud

Two lenses, one report

Web UI testing

Grounded generation from the live DOM · Test Studio authoring · Auto-Suite crawl and planning

API testing

OpenAPI-spec-driven endpoint tests — status codes, schema validation, positive and negative cases

03 / HOW IT WORKS

From plain English to verified test coverage

1 Ground

The AI reads a live screenshot and a catalog of the page's real interactive elements — plus sprint stories and acceptance criteria from Jira or Azure DevOps — and builds each step from what is actually there.



2 Verify

Every generated test runs against the live system before it is saved. Failing steps are repaired automatically across multiple passes, so no broken script is ever left behind.



3 Operate

Suites run on a schedule with consolidated UI + API reporting. On failure, AI triage returns a verdict — genuine defect or flaky selector — with a likely root cause and suggested fix.

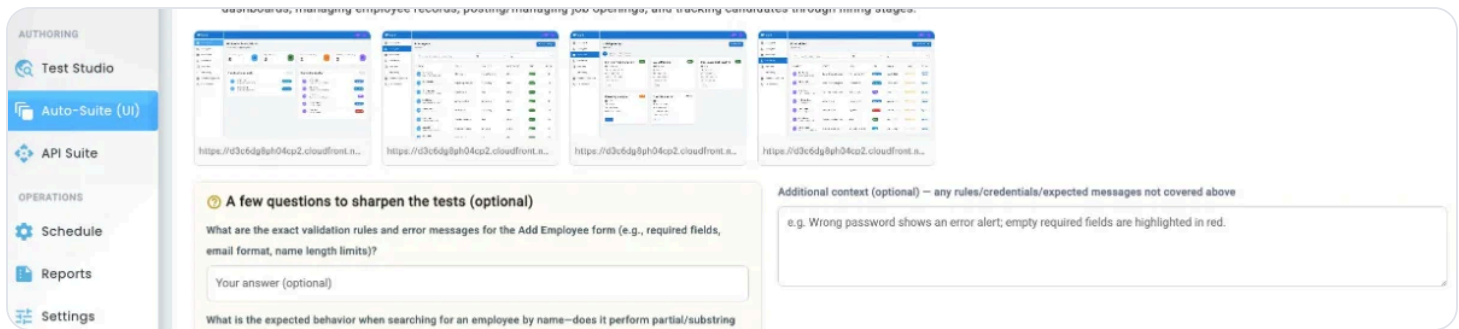
AI across the lifecycle

Human-in-the-loop

AI is applied at distinct points — UI test generation, interactive authoring, site analysis and planning, requirements grounding, API generation, self-correction, and failure analysis. **Test engineers supply business rules, thresholds, and credentials; the AI does the heavy lifting.**

What teams see

Test Studio; Auto-Suite planning; consolidated run reports



04 / PROOF FROM THE FIELD

What a live deployment delivered

An IoT/Smart-home Solutions provider's CRM implementation

60%

reduction in
regression-testing time
per release cycle, across three CRM
modules

50%

more defects caught before rollout
instead of after

200+

test cases generated within days
instead of months of manual scripting

- ✓ Dozens of critical-path scenarios moved from manually maintained checks to continuous, scheduled runs.
- ✓ Results published back into Azure DevOps Test Plans as Test Runs — the existing tracker stays the system of record.
- ✓ Test plans grounded in the team's own Jira and Azure DevOps requirements, not a generic guess at coverage.
- ✓ A materially lighter testing load, redirecting engineer time toward reviewing new scenarios instead of re-running old ones.

Results from a single deployment across three CRM modules, not a general benchmark. Outcomes/metrics will vary depending on application complexity and release cadence.

Talk to TechWish

Let us pilot our platform on your target application and a current sprint.

techwish.com

info@techwish.com

Washington DC · Chicago · India