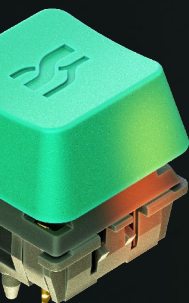
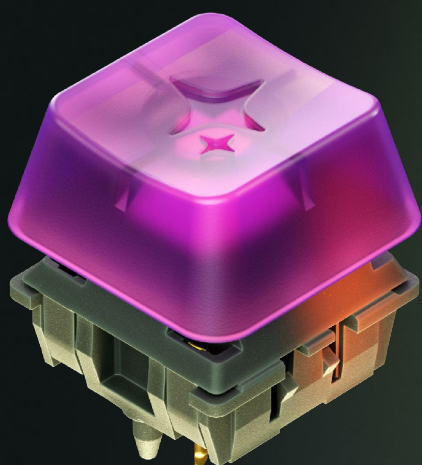




Studio



THE AGENTIC QUALITY PLATFORM

Independent environment where specialized agents build, run, and repair the tests that prove your software works



AGENT-NATIVE, NOT WRAPPED

Functionize built the models, the data layer, and the execution engine, and trained them on real production UI behavior. It is not a general model behind a testing prompt.



ADVERSARIAL BY DESIGN

Studio checks the running application from the outside, separate from the work that built it. So a pass shows what the software does, not what someone meant to build.



COVERAGE THAT SCALES

Studio runs tests in parallel across the whole product, past what a team can cover by hand. It records and versions every run, so any result traces back to what changed.



Everything is a conversation

You describe the outcome in plain language. Studio does the rest, from a live application against a running standard.

Agent Tasks		Prompt
1	BUILD THE TESTS	> Log in, file a new claim, confirm it appears in the adjuster's queue. A sentence becomes ordered steps against the live app. No selectors, no scripting. You update the instruction, not a hundred clicks.
2	FIND THE CAUSE	> Why did checkout stop at payment? Studio inspects the run, compares it to the last pass, and returns a real regression or drift, with the failing step and screenshots attached.
3	REPAIR THE DRIFT	> The submit button moved. Keep the test honest. It re-observes the page, updates the match, and reruns. Every change leaves a restore point you can roll back in one click.
4	SCHEDULE ORCHESTRATIONS	> Nightly regression on Chrome and Firefox. Start, schedule, stop, or check the last orchestration from the same conversation. Parallel cloud browsers, nothing to provision.
5	DATA-DRIVEN RUNS	> Run this once for every row in the CSV. Attach a dataset. Studio loops the same workflow across accounts, states, or SKUs without duplicating the test.
6	ADMIN THE WORKSPACE	> Add staging to the web project. Invite Jane as admin. Projects, environments, users, variables, Jira and TestRail. Anything behind a menu is available in chat.

Who it is for



Testers

Enforce the standard across the product at a scale no person covers by hand. Judgment stays with you. Execution leaves the plate.



Developers

Proof the change did what you meant, not only that it compiled. Studio verifies your running application while you build, so a break surfaces before review does.



Leaders

A go / no-go signal on every release. Coverage that compounds instead of eroding. Engineering hours moving back into building.



Architecture

1. INTERFACE

Chat First

Log in and start testing. No infrastructure to stand up.

2. PLANNING

Intent to plan

An agent reads your intent and produces ordered steps.

3. EXECUTION

Run & observe

An isolated machine runs the steps against your live app.

4. CONTEXT

App model

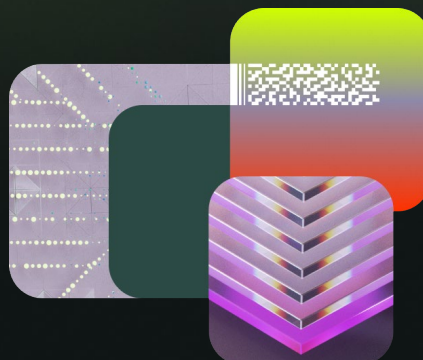
Elements resolve by what they are, not a brittle selector.

5. DATA

Compounding

Every run adds to what Studio knows about your product.

Agents interpret what you mean. A machine-learning core verifies what the application does. **Execution is deterministic:** pin the runtime and you pin the model version, so the same input gives the same result.



Studio is not a prompt wrapped around someone else's API. The understanding is ours to sharpen, and it gets more accurate on your app instead of drifting under a third-party endpoint.

Platform

Meet Studio. Ask a question or describe the test you want to run...

Press Enter to send



A WORKING TEST FROM A SENTENCE

Describe the outcome in plain language. Studio turns it into ordered steps, reusable page objects, variables, and generated data. Update the instruction when the product changes, not a hundred clicks.



SUITES THAT SURVIVE THE REDESIGN

Elements resolve by structure, text, role, state, and appearance, 200+ datapoints each. Healing tuned per step or project. Dynamic IDs, shadow DOM, and deep iframes are where selectors fail, and where Studio holds.



PROOF AT THE SPEED YOU SHIP

Chrome, Firefox, and Edge, single or in parallel. Isolated cloud VMs with nothing to provision. Suites schedule, re-run on failure, and alert. Dev, QA, and staging per project.



A GREEN RUN THAT MEANS THE PRODUCT WORKS

The same flow can verify APIs, databases, files, email, and SMS, not only what the user sees. Full-page and element visual validation, with screenshots on every step and video of the session.



FAILURES THAT EXPLAIN THEMSELVES

Studio compares a miss to the last pass and returns the cause. A fix can be applied and validated, with a restore point on every change. Nothing updates silently.



QUALITY YOU CAN TAKE THROUGH SECURITY

SOC 2 Type II, SS0, and role-based access. Isolated execution per run. Cloud or private deployment with configured data residency.

Powering quality at scale

McAfee

MARKEL


GE HealthCare

CONDUENT


SAGENT

Deployment & access

SOC 2 Type II

HIPAA Compliant

SSO & RBAC

Multi tenant or private cloud

Encrypted tunnels

Data residency

Isolated execution

Web, cross-browser

Plans

EXPLORE

Free.
See what Studio does
on your own app, one
check at a time.

PROVE

For individuals and
teams raising the bar
on quality.

SCALE

For the org where the
standard has to hold
at scale.

ENTERPRISE

Pooled credits,
configured residency,
and a plan built around
your contract.

+

Stop chasing quality. **Direct it.**

+

Go from zero to a working end-to-end test in
minutes. Nothing to stand up, nothing to maintain.

Start Free at

[FUNCTIONIZE.COM](https://functionize.com)



Studio