

---

# How Automated Testing Protects Quality

MessageFoundry treats quality as something automation enforces, not something people remember to check. Every proposed change to the engine runs through a large battery of automated tests before it can be accepted, and the same suite runs again — with extra checks — once a change lands in the main codebase. This page explains, in plain language, what that means and why it keeps the product reliable.

## The headline

---

- **Every proposed change is automatically run through roughly 2,500 tests** before it can be approved.
- **Every change accepted into the main codebase is run through roughly 2,600 tests** — the same set, plus around 100 additional checks that require real databases to run.

Because each round of testing runs on four different system setups (Windows and Linux, across two supported language versions), that adds up to on the order of **10,000 individual test runs** each time.

## What is a "test"?

---

A test is a small automated check that confirms one part of the software still works as intended. Tests run by themselves, in minutes, with no one watching. If a required test fails, the change is automatically blocked until the problem is fixed — so a mistake can't quietly slip into the product.

## Two checkpoints

---

- **When someone proposes a change.** All the everyday checks run. This is the gate a change must pass before a reviewer approves it.
- **When a change is accepted into the official codebase.** Everything above runs again, plus heavier checks that use real databases and a full Windows installation — the things too slow or costly to run on every proposal.

## What the project's automation runs

---

- **Behavior tests** — roughly 2,500 automated checks of what the software actually does.
- **Code style and safety** — automatic formatting plus type-checking that catches whole categories of bugs early.
- **Security scanners** — look for insecure code, accidentally exposed secrets, and known weaknesses in third-party software.

- **Supply-chain checks** — track and audit every outside software package the engine depends on, with a recorded inventory (a software bill of materials).
- **Patient-data and identity guards** — automated checks help keep real patient information and customer names out of the open-source code.
- **Editor add-on** — the companion VS Code extension is built and tested too.
- **Real-database and resilience tests** — full suites against real SQL Server and PostgreSQL, plus load and failure-recovery tests.
- **Windows service install** — a test that installs, runs, and removes the engine as a real Windows service.
- **Contributor agreement** — confirms every outside contributor has signed the contributor licensing agreement.

## Why two levels?

---

Running every check on every proposal would be slow and expensive — the database and Windows tests take the most time. So the fast, comprehensive checks run on every proposal, and the heaviest ones are saved for when a change is actually accepted. The trade-off: occasionally a problem that only the heavy tests can catch surfaces just after a change is accepted rather than before — and the team is alerted right away when that happens.

## How this fits the broader quality posture

---

Automated testing is one layer of a defense-in-depth approach. MessageFoundry has completed an internal self-assessment against OWASP ASVS 5.0 Level 3 (212 requirements met, 0 failed, 0 partial, and 133 not applicable, out of 345). This is a self-assessment rather than an external audit; an independent code review and penetration test are recommended at Level 3 but not required, and are planned. The testing described here is part of what underpins that posture and supports a HIPAA-compliant deployment.

---

*Figures reflect a recent measurement of the test suite. Exact counts vary slightly as the suite grows. The source of truth for what runs is the workflow files under [.github/workflows/](#).*