

1place Release & QA Testing Processes

Last Modified on 08/31/2026 9:56 am EDT

Overview of our Testing & Version Release Procedures

- We ship a **Quarterly Stable Release** that has passed our full automated and manual test suite.
- Between quarterlies, we publish **Weekly Pre-Release builds** (opt-in “beta”) so select customers can access new features or urgent fixes sooner.
- Pre-releases are supported, but they’re **not** the final quarterly build; they may change before GA.

Release Channels

1) Quarterly Stable 'Final Release' Versions for General Availability (GA)

- **Who it’s for:** All customers, production environments.
- **Quality bar:** Must pass all quality gates (see “Testing & Quality Gates”).
- **Change control:** Release notes, upgrade guide, and rollback guidance provided.
- **Support/SLA:** Full production support.

2) Daily or Weekly (Untested) 'Pre-Release' Versions for Beta/Preview Users

- **Who it’s for:** Opt-in customers who want earlier access to features or time-sensitive fixes.
- **Purpose:** Validate fixes/changes in real-world scenarios ahead of the quarterly release.
- **Expectations:** May receive minor updates week-to-week; we may iterate quickly based on feedback.
- **Support/SLA:** Best-effort with prioritized bug handling; we recommend using in **staging** first, then production only if you’re comfortable with beta software.

3) Hotfix (as needed)

- **Who it’s for:** All customers impacted by a critical issue.
- **Scope:** Minimal, targeted fixes; released outside the normal cadence when necessary.

Testing & Quality Gates

We use a mix of **automated** and **manual** testing. Every quarterly Final Release must clear all gates; weekly pre-releases clear a subset focused on the changed areas plus safety nets (smoke/regression).

Automated (CI/CD-gated)

- **Static analysis & linting:** style, code smells, security patterns (SAST).
- **Unit tests:** function/method-level correctness; fast and isolated.
- **Integration tests:** components/services talk to each other correctly (DB, cache, APIs).
- **API contract tests:** request/response schemas, versioning guarantees.
- **End-to-end (E2E) UI tests:** user flows across the app (e.g., Selenium/Playwright).
- **Regression suite:** prevents re-introducing previously fixed defects.
- **Performance/load tests:** baseline throughput and latency; detect regressions.

- **Security checks:** dependency vulnerability scans; basic DAST on key routes.
- **Migration tests:** database schema changes apply/rollback cleanly.

Manual / Human-in-the-loop

- **Smoke test:** “Does it basically run?” quick, broad pass.
- **Sanity test:** targeted verification of areas touched by a change.
- **Exploratory testing:** creative probing beyond scripted paths.
- **Cross-browser/device checks:** where applicable.
- **Accessibility checks:** key workflows against WCAG principles.
- **UAT (User Acceptance Testing):** internal stakeholders (and, for pre-releases, select customers) validate workflows and outcomes.

Alpha vs. Beta:

- **Alpha** = internal testing only (engineering/QA/product).
- **Beta/Pre-Release/Preview** = limited external testing with consenting customers.

Our daily/weekly pre-releases are **beta** builds.

Versioning & Labels

- **Quarterly Stable:** YY.MMDD (Example: 25.0701)
- **Pre-Release:** YY.MMDD.X-beta (Example: 25.0523-beta, or 25.0523.2-beta)
- **Hotfix:** YY.MMDD.X (increment patch). (Example: 25.0701.1)

Each build comes with release notes (via [Version Changes Log](#)) that list **new features, fixes, known issues, and any breaking changes.**

Automated Quick-Tests (Executed on Pre-Release Versions)

Sometimes a new feature is needed BEFORE the General When particular users request the use of an un-tested / beta version (ed a preThe following is a list of the '

[illegible]