✅ QA Entry & Exit Criteria – iOS Calculator App

Purpose

This document defines the conditions under which the QA process will begin (entry criteria) and the standards that must be met before a release is approved (exit criteria).

It is designed to minimize release risk, ensure the accuracy of core user-facing functionality, and preserve the long-term reliability of the iOS Calculator application.

QA Entry Criteria

QA activities will commence only when the following conditions are met, ensuring that testing is structured, results are traceable, and efforts contribute effectively to product quality—rather than serving as premature or reactive investigation.

1. 🔖 Requirements are clear and final

- All features and edge cases are documented, reviewed, and locked for the sprint or release.
- Behavior for error scenarios (e.g., divide by zero, sequential operators, copy without input) is explicitly defined.
- No active open questions or pending decisions block the definition of test cases.

2. 🔍 Designs & accessibility are final

- Final UI/UX specs available for all screens and interactive states.
- Accessibility labels are implemented for critical buttons (e.g., `=`, `C`, `AC`, `+`, `−`, `%`).
- Font sizes, contrast ratios, and tap targets conform to Apple's accessibility guidance.

3. 🛠️ Development build is stable and test-ready

- Functional build available via TestFlight or simulator and installable
- No known critical or blocker-level bugs at build time.

- Basic navigation and interaction flows pass a smoke test (e.g., opening the app, tapping numeric buttons, pressing “=”).

4. 🧪 Code quality gates are passed

- Unit test coverage exists for core modules: math engine, input handler, state transitions.
- Code reviews are completed and signed off.
- Static analysis and linter checks pass with no major warnings.
- App compiles cleanly without suppressing errors or warnings.

3. 📊 Analytics & crash reporting configured

- Firebase, Sentry, or equivalent crash monitoring is integrated.
- Key user interactions (e.g., successful calculation, copy action, error result) emit traceable analytics events.
- Test builds are distinguishable from production builds in monitoring dashboards.

7. 📱 Test environment is ready

- QA has access to a representative set of iPhones running iOS 15 - iOS 18+ (e.g., SE, 12 Mini, 13, 16 Pro Max).
- QA credentials, logging access, and required tools are configured.

8. 📋 Test plan and coverage established

- Manual test cases are documented, organized by functional area (core math, memory behavior, error handling, UI state).
- Regression scope is mapped for features previously validated.

QA Exit Criteria

A build will be considered eligible for release to the App Store when the following conditions are met. The focus is not only on functional correctness but also user experience, risk tolerance, and long-term maintainability.

1. ✅ Completion of test execution and validation

- Verification of all defined functional scenarios across supported devices and operating systems. No defined user flows or calculator functions may remain untested.
- Targeted exploratory testing beyond formal cases to identify unexpected behaviors or usability concerns not captured during planning.
- Any failed tests are to be addressed immediately. Failures must be either resolved, re-tested successfully, or formally accepted with risk justification by the product team.
- A clear audit trail of executed tests, results, and defect resolutions must be available as part of release documentation.

2. ❌ Resolution of defects

- No crashes, freezes, or corrupted states.
- No mathematical inaccuracies or result display issues (e.g., rounding bugs).
- No high-visibility UI misalignments, clipped buttons, or non-functional interactions.
- All medium/low issues have a clear triage decision: fixed or accepted with rationale.

3. 🧬 Regression suite is green

- Full regression run completed on latest build.
- Historical calculator behaviors remain consistent.
- Automated smoke tests executed against previous edge cases.

4. 📈 Performance benchmarks met

- No visible lag in button response or calculation display, even under rapid input.
- Memory usage remains consistent throughout extended sessions.
- CPU spikes and battery drain within acceptable range for utility-class apps.

5. 📐 Visual & interaction design is polished

- UI renders consistently in portrait mode across all tested screen sizes.
- Buttons are visually balanced, properly spaced, and responsive to touch.

- Operator state reflects the most recent selection clearly.
- No overlaps, truncations, or scaling issues across devices.

6. 📱 Cross-device and OS Validation

- Final tests passed on at least 3 real devices representing:
  - Small (iPhone SE)
  - Medium (iPhone 14)
  - Large (iPhone 16 Pro Max)

- Confirmed compatibility with all supported iOS versions (minimum: 15).

7. 📘 Documentation and sign-off complete

- Test results stored and accessible for audit/review.
- Release notes and known limitations reviewed and approved.
- QA, Product, and Design all sign off on the release.

These QA entry and exit criteria are intended to uphold a consistent level of quality and reduce release risk for the iOS Calculator application. By aligning QA activities with clearly defined readiness gates, the team ensures that functionality, performance, and user experience expectations are consistently met across all supported platforms.