



THE ULTIMATE SOFTWARE TESTING CHECKLIST

A complete QA verification guide for every phase of your release cycle

HOW TO USE THIS CHECKLIST

This checklist is organised by testing phase and testing type. Work through each section relevant to your release. Check off each item as it passes, and log a defect immediately if something fails - don't skip ahead. The release readiness section is your final gate before shipping.

☐ Not yet checked

☒ Pass

☒ Fail → log a defect

SECTION 1 - PRE-TESTING CHECKLIST

Complete before any testing begins

Environment & Setup

- ☐ Test environment matches production configuration
Same OS versions, dependencies, and data structure
- ☐ All environment dependencies are installed and running
- ☐ Test data is prepared - including valid inputs and edge cases
Anonymise any real user data before use
- ☐ Test data is prepared - including valid inputs and edge cases
Anonymise any real user data before use
- ☐ Build or version under test is clearly documented
Include build number, URL, or commit hash
- ☐ Test environment URL / endpoint is confirmed and accessible

Team Readiness

- ☐ Test plan has been reviewed with the team
Scope, environments, supported devices and browsers
- ☐ Design references and requirements are accessible to all testers
- ☐ Test cases have unique IDs and have been peer-reviewed
Check for missing coverage or unclear steps
- ☐ Preconditions for each test case are clearly defined
User state, data setup, required permissions
- ☐ Roles and responsibilities for this testing cycle are assigned

💡 Pre-testing is the fastest phase to skip and the most expensive one to skip. An unconfigured environment or missing test data will cost more time mid-cycle than the setup ever would.

SECTION 2 - TEST EXECUTION CHECKLIST

Use during active testing of every build

Functional Testing

- ☐ Core features perform as specified in requirements
- ☐ All user-facing forms submit correctly and validate inputs
Test valid inputs, invalid inputs, and empty submission
- ☐ Navigation paths work correctly across all pages and screens
- ☐ Business logic produces correct outputs for all scenarios
Include edge cases: zero values, max limits, special characters
- ☐ Error messages are accurate, helpful, and display at the right time
- ☐ Session management works correctly: login, logout, and timeout
Test idle timeout behaviour specifically - a common miss
- ☐ Data is saved, retrieved, and displayed correctly
- ☐ All integrations and API connections function as expected
- ☐ User roles and permissions are enforced correctly
Test what each role can and cannot access
- ☐ Payment flows complete successfully across all supported methods
Test success, decline, and retry scenarios

Usability Testing

- ☐ Navigation is intuitive - users can complete core tasks without instruction
- ☐ Labels, headings, and instructions are clear and unambiguous
- ☐ Error messages guide users toward correction, not just report failure
- ☐ Loading states and progress indicators are present where needed
- ☐ Interface is responsive and adapts correctly to different screen sizes
- ☐ Touch targets on mobile are large enough to tap accurately
Minimum 44×44px recommended

Performance Testing

- ☐ Page and screen load times meet defined performance thresholds
- ☐ Application responds within acceptable time under normal load
- ☐ Application maintains stability under peak concurrent user load
Coordinate with load testing tools
- ☐ Large data sets load without timeouts
Long lists, file uploads, search results
- ☐ No memory leaks or performance degradation during extended sessions

Compatibility Testing

- ☐ Application functions correctly on Chrome, Safari, Firefox, and Edge
- ☐ Tested on current and prior major OS versions
iOS, Android, Windows, macOS
- ☐ Layout and functionality verified on key screen sizes and resolutions
Mobile, tablet, desktop, widescreen
- ☐ Tested on real physical devices - not emulators only
Prioritise your top device/OS combos from analytics
- ☐ Platform-specific features work correctly on each platform
Push notifications, camera, location, offline mode

Security Testing

- ☐ Authentication and authorisation behave correctly for all user roles
- ☐ Sensitive data is not exposed in URLs, logs, or error messages
- ☐ Input fields are protected against injection and unexpected input
SQL injection, XSS, script injection
- ☐ Session tokens expire correctly and cannot be reused after logout
- ☐ HTTPS is enforced across all pages and endpoints
- ☐ File upload functions restrict file types and sizes appropriately

Accessibility Testing

- ☐ All interactive elements are reachable and operable by keyboard alone
- ☐ Screen reader announces all meaningful content correctly
Test with VoiceOver (iOS/macOS) and TalkBack (Android)
- ☐ Colour contrast meets WCAG 2.2 AA minimum
4.5:1 for body text, 3:1 for large text and UI components
- ☐ All images have descriptive alt text
Decorative images should have empty alt=""
- ☐ Form fields have visible labels associated programmatically
- ☐ Focus indicator is clearly visible on all interactive elements
- ☐ Content reflows correctly at 400% zoom without horizontal scrolling

 Log defects in real time - don't batch them for later. Always include: reproduction steps · expected vs. actual behaviour · screenshot or video · device / OS / browser / build number · severity level.

SECTION 3 - REGRESSION TESTING CHECKLIST

Run after every code change to protect existing functionality

- ☐ All previously passing test cases re-executed against the new build
- ☐ Core user workflows pass end-to-end
Login, purchase, onboarding, and key actions

-
- ☐ Features adjacent to changed code have been re-tested
Ripple effects are the most common source of regressions
 - ☐ All previously fixed defects re-verified - no regressions
 - ☐ Automated regression suite has been run and results reviewed
 - ☐ New test cases added for any new features in this build
 - ☐ Test suite updated to remove any cases that are now obsolete
-

SECTION 4 - POST-TESTING CHECKLIST

Complete after each testing cycle

-
- ☐ All discovered defects logged with full details in the bug tracker
Steps, expected/actual, severity, device/OS/build
 - ☐ Defects assigned to the appropriate owners for resolution
 - ☐ Test results summary documented and shared with stakeholders
 - ☐ Pass/fail status recorded for every test case in this cycle
 - ☐ Test environment has been reset or shut down as appropriate
 - ☐ Feedback on the testing process captured for future improvement
What was missing? What took too long?
 - ☐ Checklist updated to reflect any new coverage areas identified
-

SECTION 5 - RELEASE READINESS CHECKLIST

Final gate - do not ship until every item is checked

Defect Resolution

-
- ☐ All critical and high-severity defects resolved and re-verified
No open criticals. Period.
 - ☐ All medium-severity defects resolved or have accepted risk sign-off
 - ☐ No regressions introduced by defect fixes
 - ☐ Known issues / accepted risks documented and communicated to stakeholders
-

Quality Gates

-
- ☐ Full regression suite passed against the release candidate build
 - ☐ Performance meets defined thresholds
Load time, response time, throughput
 - ☐ Security checks complete - no critical vulnerabilities open
 - ☐ Compatibility confirmed across all supported devices, browsers, and OS versions
-

-
- ☐ Accessibility testing complete - no critical WCAG violations open
 - ☐ End-to-end user journeys pass on the release candidate build
-

Release Logistics

-
- ☐ Release notes or changelog prepared and reviewed
 - ☐ Deployment plan confirmed with engineering and operations
 - ☐ Rollback plan defined and tested
Know exactly how to revert if something goes wrong post-deploy
 - ☐ Monitoring and alerting confirmed active for post-release observation
 - ☐ Stakeholder sign-off received
-

SECTION 6 - SPECIALISED CHECKLISTS BY PRODUCT TYPE

For Websites

-
- ☐ All forms submit correctly and validate inputs
 - ☐ Navigation works across all pages on desktop and mobile
 - ☐ Pages load correctly in Chrome, Safari, Firefox, and Edge
 - ☐ Images load correctly and have appropriate alt text
 - ☐ Keyboard navigation works throughout the site
 - ☐ 404 and error pages handled gracefully with a path back to content
 - ☐ Sitemap is accurate and all linked pages return 200 status
 - ☐ Analytics and tracking scripts fire correctly on key pages
-

For Mobile Apps

-
- ☐ App functions correctly on current iOS and Android versions
 - ☐ Tested on a range of real physical devices and screen sizes
 - ☐ Push notifications deliver correctly and navigate to the right screen
 - ☐ Offline / degraded connectivity behaviour handled gracefully
 - ☐ Camera, location, microphone, and permissions behave correctly
 - ☐ App passes App Store and Google Play submission requirements
 - ☐ In-app purchases and subscription flows complete successfully
 - ☐ Performance is acceptable on older / lower-spec devices
 - ☐ Gestures (swipe, pinch, long-press) behave as expected
-

For E-Commerce / Payment Flows

- ☐ Product search, filtering, and sorting return correct results
- ☐ Product pages display correct information, pricing, and availability
- ☐ Add to cart, update quantity, and remove items work correctly
- ☐ Checkout flow completes successfully with all supported payment methods
- ☐ Payment declines, retries, and error messages work correctly
- ☐ Discount codes, gift cards, and promotions apply and calculate correctly
- ☐ Order confirmation emails triggered and contain accurate information
- ☐ Order history and account management functions work correctly
- ☐ Returns and refund flows work as designed
- ☐ Tax calculation correct for all supported regions

COMMON MISTAKES TO AVOID

X Scope is too vague	Unclear scope means testers spend time on low-impact areas and miss critical workflows. Define exactly what is in and out of scope before testing begins.
X Steps written as assumptions, not instructions	Ambiguous steps produce uneven coverage. Every step should state the action AND the expected outcome.
X No clear pass/fail criteria	Subjective results slow release decisions. Define measurable expected results for every check.
X Checklist never gets updated	An outdated checklist creates false confidence. Review and update after every release cycle.
X Poor defect reporting	Insufficient details make fixes slower and retesting harder. Always include: reproduction steps · expected vs. actual · screenshot · severity · device/OS/build.

Document based on plusqa.com/post/ultimate-software-checklist-guide.