

■ QE Testing Checklist

■ Understanding the Change

- What changes have been made?
- Where in the system were these changes applied?
- Why was this change made (bug fix, hotfix, enhancement, refactor)?
- Is this change temporary or permanent?
- Are there dependencies (configs, feature flags, 3rd-party services, infra)?

■ Test Scope & Coverage

- How should I test this change (manual, automation, exploratory)?
- Do we have updated requirements, ACs, or Jira tickets?
- What test data is needed? Do I need to create/generate it?
- What areas/features could be indirectly impacted?
- Should I consider non-functional testing (performance, security, accessibility)?
- Do we need new/updated automated tests?

■ Hotfix-Specific

- What exact issue does this hotfix fix?
- Was the issue reproducible before (steps available)?
- Can I verify the fix in isolation?
- Do I need to run quick sanity checks around the fixed area?
- Is targeted regression required?
- Was the hotfix deployed correctly (versioning/build integrity)?

■ Regression Testing

- Which modules/features are directly affected?
- Could this change break shared/common functionality?
- Do we need partial or full regression?
- What's the risk if regression is skipped?
- Are there backward compatibility concerns (APIs, data formats)?
- Have I prioritized regression tests by risk/criticality?

■ Smoke Testing

- Do we just need to confirm system stability?
- Which core flows must always be tested (login, checkout, payments, API health)?
- Is this a pre-deployment or post-deployment smoke?
- Are logs/monitoring metrics behaving as expected?
- Do we have a time window to complete smoke validation?

■ Release Readiness

- Has this been tested in all required environments (dev, QA, staging, prod)?
- Are rollbacks possible if the fix fails?
- Do I need to log exploratory testing notes for future test design?
- Is monitoring/alerting in place for the hotfix area?
- Do release notes/test cases/documentation need updating?
- Is stakeholder/PO sign-off required before release?

■ QE Mindset Add-ons

- Risk Analysis → What's the worst that can happen if this breaks?
- Shift-Left → Review ACs, PRs, design docs early.
- Continuous Improvement → Capture gaps post-release & improve.
- Data & Logs → Validate in DB, logs, monitoring, not just UI/API.
- Communication → Clarify assumptions with Dev, PO, Ops before closing.

Note:

Understanding the Requirements - > Firstly, comprehensively understand the project's requirements. Gathering all necessary information and clarifying and clarifying any uncertainties to ensure a precise understanding of the task at hand.

Estimating the Deadline - > After understanding requirements, it's essential to estimate the project's deadline based on the complexity of the task, the resources available, and the team's capabilities. Accurately determining the timeline will ensure a realistic and achievable project schedule.