Playbook Methods Repository

# **Manual Testing**

Leverage product knowledge to perform the initial verification of new or updated functionality, or regression testing of existing functionality if automation is not possible.

### Remote Agility: • Medium

### Linked Tactic(s): [Test Automation](https://docs.google.com/document/u/0/d/1HKQP85Qj1gfh9CEhI4To063Hw1pDQlSReSMALOU4Q7A/edit), [QA Strategies](https://docs.google.com/document/u/0/d/1c5y3X6Fyh2GJo-Qs2Ssn4zFeIz9FLvA0EYONHgfMd2E/edit)

## Why we do it:

Manual Testing is the most effective method of verifying new product changes and functionality. It leverages the full product knowledge held by QA Professionals to examine changes holistically, taking into account interactions with all parts of the system. This enables unexpected side-effects of the changes to be uncovered, to account for scenarios missed during the planning phases. In addition to this, Manual Testing is most often employed where [Test Automation](https://docs.google.com/document/u/0/d/1HKQP85Qj1gfh9CEhI4To063Hw1pDQlSReSMALOU4Q7A/edit) is not feasible, often when a project involves custom hardware or technologies.

## 

## When to apply it:

* When changes are being tested for the first time: There is usually no test automation in place for changes when they are tested for the first time. Thus, the task of verifying the changes against their corresponding requirements is done manually.
* When test automation is impossible or prohibitively costly: In the case where existing Test Automation tools are not suited to the product being developed (if custom hardware is involved, for example), developing automated tests can be either prohibitively costly, or completely impossible. Thus, testing activities such as Regression Testing and Smoke Testing will be done manually.

## Best Practices & Considerations:

* Tailor Test Case Detail to Project Purpose and Duration: The level of Test Case detail required for a project building a new product will be much higher than a project building a short-lived prototype intended to validate an idea. In the former, the tests and execution results will be referenced multiple times not only during the project, but over the product’s entire lifespan. The tests will serve as documentation of the system’s functionality, and thus should be fully-documented, and stored in a Test Case Management Tool. For projects that will be much shorter-lived, the return on creating thorough test documentation may not be sufficient to justify the cost. In these situations comments on individual tickets summarizing the scenarios tested and the test environments used may be sufficient.
* Diminishing Returns of Repetitive Testing: Over time, there is a risk of team members becoming desensitised to defects when repeatedly executing the same test scenarios, due to their familiarity with the flows and expected results. This most often happens with Smoke or Sanity Tests, activities that are performed at semi-regular cadences, and coincide with product releases or updates. Fortunately, there are strategies for mitigating this:
  + Prioritize the tests most often executed for [Automation](https://docs.google.com/document/u/0/d/1HKQP85Qj1gfh9CEhI4To063Hw1pDQlSReSMALOU4Q7A/edit), so they are executed programmatically.
  + If Test Automation is not possible in the near or long-term, consider rotating the responsibility of performing the repetitive testing amongst the team. This has many benefits: It brings a fresh set of eyes each time the testing is performed, and also increases empathy for QA Team members amongst the wider project team.
* Ensure the whole team can participate: Expanding on the point above to rotate manual testing responsibilities amongst the team, this should be possible on every project with a manual testing component. There will be times when QA Team members have other tasks to complete, or are otherwise unavailable (on PTO, for example). In these situations both the test cases themselves and the execution process should be sufficiently documented to allow for any member of the team to jump in. To ensure that non-QA team members feel comfortable with this, consider performing testing rotations when QA Team members are present to offer assistance.

## Responsible roles:

* QA Engineer: Document test scenarios and testing processes, document all defects encountered during testing, and report on manual testing progress to stakeholders.
* Product Manager: Work with QA to define and prioritize test scenarios, primarily to ensure expected results are correct, and provide clarification when writing tests for ambiguous or previously-undefined flows.

## Tools:

* Online tools/platforms/services
  + TestRail: <https://www.gurock.com/testrail/>. Thoughtworks QA’s recommended tool for Test Case Management.
  + Zephyr Squad: https://marketplace.atlassian.com/apps/1014681/zephyr-squad-test-management-for-jira?tab=overview&hosting=cloud. One of two Test Case Management Plugins for Jira.
  + Xray Test Management: <https://marketplace.atlassian.com/apps/1211769/xray-test-management-for-jira?hosting=cloud&tab=overview>. The second of two Jira Test Case Management plugins.

## 

## Thoughtworks Examples - Linked

* Thoughtworks’ internal TestRail instance: <https://testrails.connected.io/index.php?/auth/login/>
  + Ask IT for login credentials
  + Often used as the initial source of test case documentation on a project, before transitioning to a client-hosted tool.