Playbook Methods Repository

# **QA Testing Strategy**

Determine the types of testing that will be performed during a project, and the testing responsibilities across the project team. This facilitates prioritization of testing efforts, and the identification of work to enable testing activities.

### Remote Agility: **•** High

### Linked Tactic(s): Quality Assurance

## Why we do it:

A QA Testing Strategy enables the team to uncover, enumerate, and prioritize the testing efforts required over the course of a project. This is often done to establish responsibilities and processes, identify unknowns and potential blockers to testing, and ensure that testing is not limited to the verification of individual tickets. Some concrete examples of elements in a QA Strategy include:

* How testing fits into the development workflow, i.e., is manual testing performed in a temporary environment after a pull request is opened, or in a staging environment after changes are merged into the Main code branch?
* Whether Load or Performance Testing will be conducted, and at what stages of the project.
* The types of tests (Unit, Integration, Automated UI, Manual, etc.) that will be created at each phase of development, and which role(s) are responsible for their creation and maintenance. Are developers responsible for creating Unit and Integration tests, for example, and QA Team members create Automated UI and Manual Tests?
* The testing processes for releases to each component of the system.

## 

## When to apply it:

* After initial technical decisions have been made: In order to determine the activities to be performed and questions to answer, QA Team members need a high-level architecture of the overall system: The different components that will be built, the technologies that will be used to build them, and the scale at which the system will be expected to operate.
* Projects with multiple components, or limited QA resources: It’s easy to lose track of testing activities as the amount of testing in a project increases, especially if QA resourcing will be limited. A project that is building a backend service, web application, and cross-platform mobile applications is more likely to have performance and security testing tasks forgotten or deprioritized than a project building a single component.
* Projects with custom hardware or technologies: The unknowns involved in working with custom hardware or technologies make such projects the perfect candidates for QA Strategies. Custom technologies do not have performance or behaviour data from years of use, and thus require in-project testing to identify gaps or unforeseen issues.

## Best Practices & Considerations:

* Make the Strategy immediately actionable: A common pitfall of QA Strategies is similar to that of Waterfall development: The strategy is written, then never consulted or used again as changing requirements or other factors result in only the basic level of testing being performed due to reprioritization of work. To avoid this, the strategy should contain items that can be immediately acted upon (along with assigning responsibility, where possible), with guidance for follow-up tasks at set points or intervals during the project. Some examples of these items are:
  + Evaluating tools for specific purposes, such as Test Automation, Test Case Management, or Load Testing
  + Setting up cloud infrastructure for the desired test environments
  + Determining the feasibility of test automation where custom hardware is involved
* Do not make assumptions about aspects that are “obvious”: As experts in their field, QA Team members can sometimes assume that certain practices or responsibilities are intuitively obvious. For example, the fact that those in QA roles will write and maintain the manual test cases for the various components of the system. These practices may not be obvious for team members in other roles, however, as they may have little or no experience working with QA Professionals, or desire a high level of process documentation and standardization. This reinforces the importance of strong collaboration with Client Stakeholders, to ensure the Strategy contains all of the expected material.
* Document tasks even if they will be performed by 3rd parties: On larger projects it is likely that there will be more work than the members of the project team can complete on their own, or require specialized expertise that the team does not have. These types of tasks, such as Security Testing, can be outsourced to 3rd-parties, and in such cases should still be documented in the Strategy. This ensures Stakeholders are aware that the tasks will be performed, and the decision to outsource them if necessary.

## Responsible roles:

* Client Stakeholders: Provide guidance on the content of the Strategy, and work with QA Engineers and Engagement Leads to ensure the Strategy is aligned with expectations.
* QA Automation Engineer: Create the Strategy, and validate with Client Stakeholders.
* Engagement Leads: Facilitate discussions between QA and Client Stakeholders to ensure the Strategy is aligned with expectations.

## Tools:

### Online tools/platforms/services

* + Any online document-creation or Wiki tool is suitable, e.g. Google Docs, Confluence, etc.

## 

## Thoughtworks Examples - Linked

### Ford Canopy (formerly Sentinel) Production QA Strategy

* + Contained in multiple documents [here](https://drive.google.com/drive/u/0/folders/1iHRYB0iaSxCvVrfle2UYkduT8C83z3ov)
  + Developed prior to the project switching to development for production, in January of 2022.