

# Performance Testing Checklist:

## 9 Key Elements to Consider Before Testing

Performance testing is an essential step in the software development process that truly enables organizations to deliver high-quality applications. However, it is not uncommon for performance testing to be neglected and only executed prior to an application's deployment. When this happens, applications can fall victim to complex and costly performance-related issues that ultimately lead to a poor and unreliable user experience.

To improve your performance testing practices, check out our pre-test checklist that includes 9 points to consider during the test planning phase.

### 01

#### Establish the performance testing goals



To establish the goals or objectives of the performance test, determine what is the most important thing you or your team wish to know. This can include the application's performance during peak workloads, the application's stability under heavy workloads, how quickly the application responds to important requests, etc.



### 02

#### Identify the type of application that will be tested

To properly create a performance testing plan, it is essential to identify the type of application that will be tested. If not, the test plan will end up being too broad for the actual application that needs to be tested.

Important considerations include determining if the application is web-based, a mobile application, or desktop application. At the same time, be sure to identify the application's operating system.

### 03

#### Record the application's configuration/architecture



Before testing the application, it is important for the QA team to identify application's the baseline configurations with the hardware variables. These variables can include the Service Packs, Operating System, RAM, CPU, HDD, etc. Identifying these variables will allow the QA team to design performance tests that match the application's architecture and uncover performance related issues.



### 04

#### Verify product interaction involving any 3rd party vendors (credit card processing)

It is not uncommon for many applications to have third-party payment gateways. However, testing these gateways requires special considerations. To properly test third-party payment gateways, consider using a sandbox environment to create a secure test environment to assess key scenarios on an application with a fully integrated payment gateway.

### 05

#### Document any special software or plug-ins



It is not uncommon for many applications to have third-party payment gateways. However, testing these gateways requires special considerations. To properly test third-party payment gateways, consider using a sandbox environment to create a secure test environment to assess key scenarios in an application with a fully integrated payment gateway.



### 06

#### List browsers, connection speeds, or devices needed for the test

An application's performance can be effected by end-users' devices, browsers, network speeds, and more. Replicating these aspects is a better way to assess the application's performance during the test.

### 07

#### Identify any use cases or critical scenarios that need to be tested



While planning any performance test, it is crucial for the QA team to identify any critically important use cases or scenarios. These use cases or scenarios will often be those that are most commonly used or resource-intensive, and thus need to be tested. Evaluating these key scenarios will allow the QA team to uncover performance-related issues that can affect your major user paths or features such as searching for a product, creating an account, placing an order, etc.



### 08

#### Select specific metrics to monitor

Selecting which performance metrics or KPIs to monitor during the performance testing is rather important. Not only do KPIs help to guide your team, they also provide key insight into the application's performance.

### 09

#### Choose important benchmarks for the test



It is essential to establish benchmarks for performance testing. Benchmarks are metrics that allow testing teams to determine if the application, based on the results of performance tests, meets established quality standards. Comparing the results of these tests with the benchmarks established prior to testing is essential to maintaining high-quality applications.