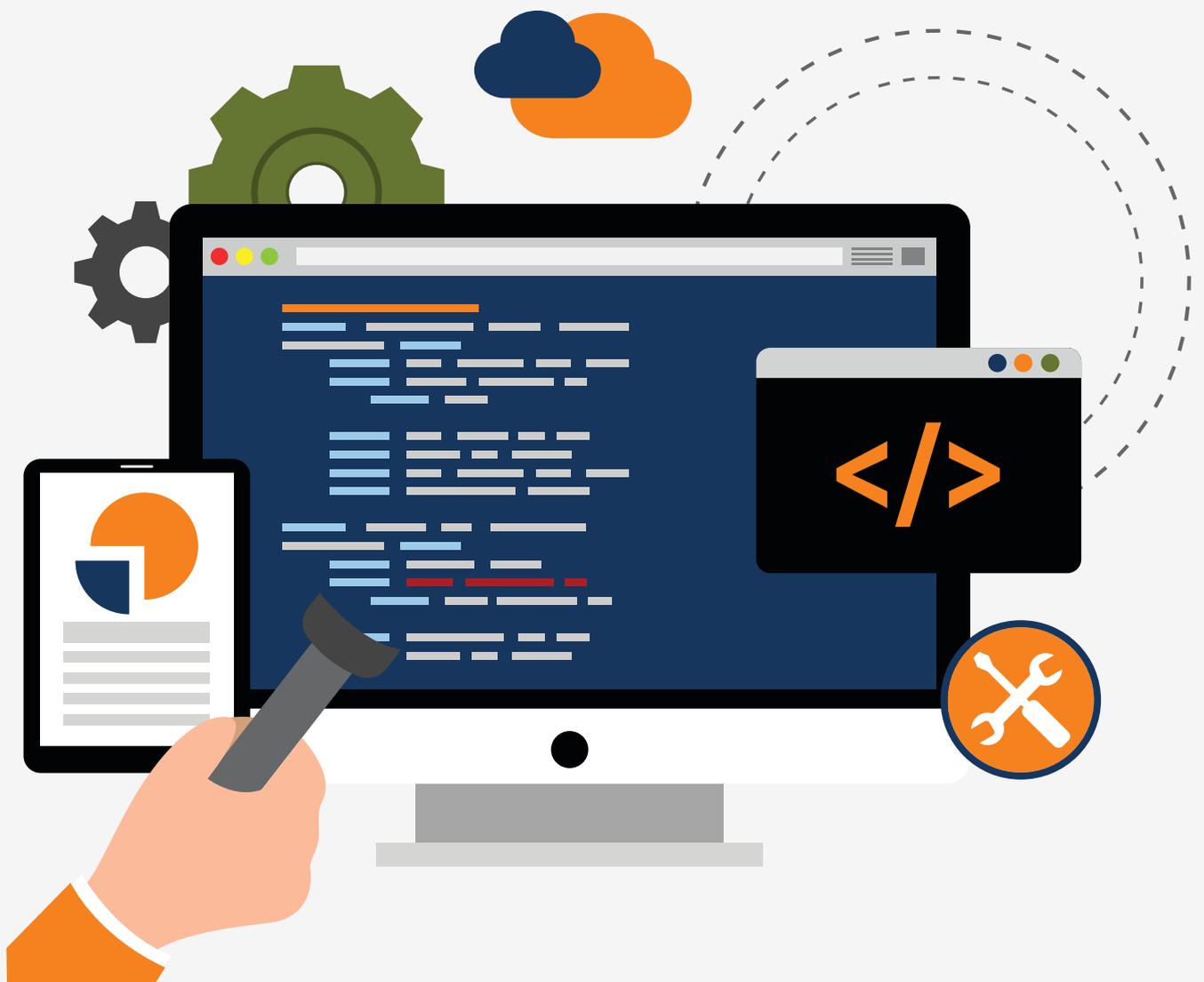


Overcoming API Testing Challenges

Anticipating And Resolving Common Issues In API Testing



Code-Level Functionality

The main goal of API testing is to ensure the functionality of an application at code level. It occurs prior to the testing of a GUI, which requires manual and automated testing procedures. Recognizing the common challenges associated with API testing and planning your approach will help reduce the number of issues throughout the testing process. It will put both development and QA teams at ease while expediting your development and testing cycle.

Adhere to Best Practices to Overcome API Testing Challenges

There are five best practices to keep in mind when embarking on a full suite of API testing. Implementing these practices will smoothen each step of the testing process, from initial understanding and planning to execution and reporting. By adhering to the below best practices, you will forego many challenges that can occur during API testing.

» Acquaint Your Team with the API

Like other aspects of [QA Services](#), especially offshore QA, the amount of time that the API testing team spends with the API has a direct impact on the rest of the test project. Allow your team plenty of time to review and familiarize themselves with the functionality of the API.

» Choose a Suitable Testing Framework

Define the scope and makeup of the testing framework. Ideally, the selected framework will support GUI tests, as this will allow for easy integration between both GUI and API functions. The ideal framework will also incorporate API libraries, reporting tools, and build management tools.

» Test for Expected Usage

It is not feasible to test all possible scenarios, hence testing scenarios should replicate most likely usage. An example scenario would be one that involves interaction between several different modules of the application.

» Adopt Appropriate Test Techniques

To ensure efficient code coverage and to maximize the effect of your testing cycle, your API testing team should utilize techniques such as equivalence classes, boundary value analysis, and error guessing. Selected parameters should test functionality.

Valuable API Testing Techniques

Equivalence Class: Test scenarios are divided into valid/invalid groups. The most acceptable scenarios from the groups are picked and verified during API testing. For example, in a text field where only numbers 0 to 9 are valid inputs, only one test using a valid input is required. For invalid values (those below 0 and above 9), two test cases will be sufficient.

Boundary Value Analysis: The boundaries of a software application are likely failure points. To use the example above, checking input field length at 0 and 9 is a good validation, but checking again 10, will ensure it is not broken at the boundary.

Error Guessing: The tester's prior knowledge and experience is implemented in this methodology. Tests are designed using either functional documentation or customized to account for unexpected errors.

» Design Successful Test Cases

Test cases designed with the end-user in mind will yield best results. Successful test case design requires considering the full functionality of the system as well as how it will be used by the end-user.

Key Benefits of API Testing

- Less time-consuming than functional GUI tests
- Code-level testing for core functionality
- Lightweight XML and JSON data transfer modes
- Automation can be achieved using any core language
- Used in tandem with development and manual/automated testing
- Easy integration with GUI testing

For more info about the benefits visit our blog, [4 Advantages of API Testing](#)

Defining a Dynamic and Static Parameter

These terms are used primarily in API testing. A dynamic parameter is a value that will change during testing. In example, during a test, creation of a new user account would be a dynamic parameter. Static parameters are those that never change in value during a test cycle. A pre-existing user account would be an example of a static parameter.

Overcoming Specific Challenges of API Testing

Occasionally, challenges will arise before or during the testing process. These issues can be dealt with, time can be freed up, and the development cycle can be streamlined, if you keep in mind the best practices discussed earlier. Communicate closely with your development and [Offshore QA](#) or API testing team. Review these high-level challenges which are commonly noted during testing of API functions. Consider the listed approach that will help your team overcome them.

» Selecting and Integrating GUI/API Framework

Always aim to work in the same language in which the application has been built. Opt for framework that offers support for GUI tests to allow for better integration and streamlined testing. By adhering to these two guidelines you will improve communication between development and testing teams and expedite release.

» Selecting Data Structure for Handling Requests/Responses

Requests/responses during testing are best handled using data structures, as they provide a firm level of control on the overall flow of your test scenarios. Select test scenarios that mirror actual usage and can be successfully executed within the active time of the API function.

» Handling Timeouts and Other Errors During Testing

Timeouts and other various errors may become apparent during execution. These are most effectively dealt with using an exception-handling mechanism. Splitting your test data into dynamic and static states will assist in proper management of the scripts that have been developed. For example, an authorization token would be a dynamic parameter, whereas a developer key would be a static parameter. Once split, these can be used alongside other test data.

Reaping the Benefits of API Testing

API testing is worthwhile because it allows for a code-level functionality check of the application without GUI. It can be performed in tandem with development, and it can be used alongside functional automated and manual testing. This testing approach will expedite your development, QA, and release cycle whether you will be performing API testing in-house or with a dedicated offshore QA team.

For more free information from the QASource Knowledge Center, [Follow us on LinkedIn](#).

Quality That Creates Value

QASource employs a well-trained team of dedicated QA engineers who specialize in testing new software product development in mobile, cloud, and desktop applications. Our goal is to optimize our client's software quality assurance efforts while meeting their objectives and schedules. We know how to "go beyond the checklist" to deploy high-capability test practices resulting in high-quality software releases, time after time. Our key strengths have enabled us to complete hundreds of successful engagements for clients throughout the world and guarantee a level of expertise and knowledge that other firms don't provide.

QASource - Corporate Headquarters

73 Ray St, Pleasanton, CA 94566.

Tel: +1 925.271.5555, Fax: +1 925.226.3138