



Quality Assurance

Software QA-as-a-Service

Exevio QA Testing Service provides a detailed QA analysis of your product on a daily, weekly, monthly or per version basis. QA consists of in-depth preparation that includes getting to know the product, product research, competitor research (for use cases and UX), understanding the functionalities, setting up test cases, testing, logging, recording and re-testing.

We deliver a complete quality assurance report with a goal to solve issues like bugs, avoid mistakes or defects from happening within the product and making the product of higher quality long term.

Our QA team is experienced with up-to-date processes considering the variety of platforms, possible network conditions, constant rise of new technologies and with that - challenges.



QA: Coverage



1. Web

Generic websites, online stores, back-end systems, browser extensions, APIs, etc.

2. Mobile

iOS, Android, Windows, hybrid web/native apps

3. Desktop

Windows, MacOS, Linux executables

QA: On-boarding Process



1. Non-Disclosure Agreement

Your loyalty and privacy is of most importance to us. Having a Non-Disclosure Agreement protects company's trade and product secrets while getting valuable feedback through the use of Exevio Ltd. services.

2. Product Documentation

If existing, documentation would greatly help understand the product prior to the initial in-depth presentation. It helps us tailor the offer that suits your needs in better detail and gets us prepared for questions before the final quote.

3. Product Presentation

It is required to present the product to completely understand how it works, to define the needs and explore through demoing. We need to understand client's requirements, what are the expected goals to achieve with testing, defining testing types (explained in more detail below), the expected number of daily/weekly/monthly tests, etc.

4. Mandatory Use Cases

By defining the mandatory use case goals, scenarios and flows we can better target

contracted testers required for certain test types (explained below).

5. Final Quote & Agreement

After minimum criteria to form a quote is met, we deliver and present one: stages, required time per test, required number of internal & contracted (targeted audience) testers and final pricing. Upon confirmation and concluded agreement, we deliver the contract required to initiate the QA and to protect both parties.

6. Full Access & Credentials

Credentials with full access to the product is required to start a detailed research on every aspect of it before executing QA. Additionally, if needed for certain tests, we require credentials for 3rd party services and payment processing gateways e.g. to test the payment in the sandbox or simulated live/production environment.

7. Testing & Reporting

In-depth testing starts with the agreed timeline & deadlines. Issues are delivered through internal QA Management System, Trello, JIRA or any other platform you prefer.

Each issue is described with:

- a. Exact timestamp
- b. Steps to reproduce
- c. Screenshots
- d. Video if required

Detailed reports are delivered as agreed on weekly, monthly or per full test basis.

8. Updating & Re-testing

Each issue follows this process:

- a. Discovery + logging as a new entry (as described in Testing & Reporting)
- b. Fixing (your task / status change)
- c. Fixed (your task / status change with notes on what was the issue)
- d. Re-testing & validation (QA team)
- e. Closed (QA team)



The goal is to execute a number of various tests in order to ensure high quality results. Results being issues found by QA testers in 95% of cases, not spontaneously by end users.

Product = Web, mobile or desktop application.

Smoke Testing

This phase covers a preliminary set of tests that can reveal some basic failures and common issues severe enough to cause the product not working in most common cases, or to get rejected from the App/Play/Windows Store.

- Test cases that cover the most important functionalities and all the crucial functions of the product to work correctly and as expected.
- To determine whether the application is so badly broken that further testing is unnecessary until the code improvement and fixing is done.

Functional Testing

At this point, each possible case and scenario within the specific set of product functions gets tested, measured, commented and logged.

- To test if documented features are all implemented in the product and working as expected.
- To test and validate the functions at the user interface level (all of the aspects of usability, readability and consistency) using different browsers, devices, settings, etc.
- To validate each of the specific components (e.g. video/sound streams, e-commerce submissions), ensure they function as expected and that all of the available links are in function and working correctly.

Performance Testing

This is a phase of identifying any notable slow or strange performance.

- To examine the screens with slow performance in detail, e.g. long screens with a scroll.
- Screens influenced mostly by slow connection get tested under several different settings.
- Stress testing covers the scenarios of fast screen changes, wild usage of all the functional aspects. Any strange behaviors are recorded for easier resolving in future product development.

User Experience & User Interface Testing

- Overall and specific feedback related to how the product looks and feels.
- Suggestions on the possible visual or functional changes; wherever the user might get lost, confused or simply get a less effective experience.

Security Testing

- To examine if the user's data is properly protected.
- To detect and predict the risks related to the specific app.
- To find the security vulnerabilities that could either endanger users and their data, or allow risky or undesirable behaviors to happen.

Other: By Demand

Any other required test cases or testing outside the regular scope.

Advanced: Penetration Testing

In order to provide a high quality security assessment of the product and to detect vulnerabilities that can compromise the product owner, Exevio Ltd. offers a specialized service for web and mobile application penetration testing, carefully customized for each new app and tailored to all the popular platforms. The service includes hacking or cracking the product.

QA: Prerequisites



Testers

Each product comes with different technical requirements. Our employees and contractors are trained and optimized for any test type e.g. regular targeted testers/users paid to execute certain tests under NDA or employees with technical skills required for all tests.

If required, targeted testers can be video recorded for UX improvements.

Devices

We are equipped with all the generally required devices for tests e.g. all iPhone and iPad series and most popular Android & Windows devices. In case the product targets a specific set of devices, we will ensure to acquire them for testing.

QA: Pricing



Pricing is defined by many different factors and complexity of the project in general. Service is charged per hour per tester + initial setup.

- **Initial setup**
Covers a part of the on-boarding process up to agreed pricing, prior the QA execution.
2.000 EUR (15.000 HRK)
- **QA management**
Manager time for QA management, client communication, reporting and meeting time.
Pricing depends on the complexity of the QA project, starts from:
500 EUR (3.750 HRK) monthly or per full report
- **QA tester time**
Charged per hour per tester for estimated time in total monthly.
20 EUR (150 HRK) hourly

Flat-rate pricing available for certain conditions.

All prices are without VAT.

EU reverse charge is applied where applicable.

Example project: Mobile app QA estimation

Pricing factors:

- Initial setup (meetings, documentation, presentation, defining use cases)
- iOS and Android application
- Single pre-release version
- First time full testing
- Deadline: 2 weeks after agreement
- 3 people / testers
- 30 different devices with various OS versions
- QA work: communication, fixing, re-testing, etc.
- Full (detailed) report

Total pricing:

7.300 EUR (54.800 HRK)