When you want to have a piece of software built you mostly look at the end result, which is a functioning product. However, when receiving a breakdown of the costs, many business owners question the part of product development called testing. Why? Usually because of its price. It may be (not always!) one of the most expensive processes that many would gladly cut out from the product development lifecycle. However, it is an absolutely vital and integral part. Today's article will tell you all about testing and why we will never advise you to skip this particular part.

Why do we test software?

The simplest answer to that question will be: to make sure the software works. If there are any bugs or errors in the code, testing helps to identify them before the software release. However, the reasons for testing are countless. If we were to give you the most important ones, we would start with the below.

Cost-effectiveness

You may think this is an absurd statement as you are trying to avoid testing for the very same reason. Unfortunately, you are risking a greater financial loss than the savings not testing will bring. To start with, catching irregularities in the early stages of the project makes them relatively easy and cheap to fix. If the errors are caught after the release, it involves such costs as:

- loss of revenue due to the software's unavailability for clients;
- loss of income due to the clients turning to your competitors;
- increased costs connected to finding and fixing the error;
- potential financial repercussions connected to legal proceedings resulting from your system failure.

Enhanced security

Especially in the 21st century, where cybercrimes are at such an extraordinarily high level, security plays one of the most crucial parts in any software development process. Many entrepreneurs still do not realize that faulty applications and software with errors are simply an invitation for user information and details leaks. However, more and more users pay attention to software reliability and sufficient testing. Very often, bugs are the main reason why users stop using the software. Moreover, it not only causes lower income but also destroys your reputation.

Remember that an unsatisfied user is a very dangerous weapon, shooting with unfavorable opinions left and right. You can be sure that a security threat will make a good enough reason for bad reviews.

High quality of your software

Software with errors and bugs is not only expensive and dangerous. It also is simply an off-putting experience. Even though the errors eventually get rectified, they leave a sour taste in one's mouth. Even though your software may be brilliant, errors and bugs significantly lower its quality.

Greater user satisfaction

One wise man once said 'Your product is as good as the customer satisfaction it brings'. Who would want to use software with bugs and errors? It only causes problems, frustration, and a horrible customer experience. Product quality is a major factor in customer satisfaction. And since we already established that an erroneous product is of low quality, your customers will be unsatisfied if they receive such a product.

What are the risks of not testing software?

Sometimes the benefits are just not enough to stress the importance of certain actions or behaviors. There are different people and methods of getting through to them. Let us get more realistic and mention a few real-life situations where the testing was 'too expensive' seemingly 'not necessary' or simply not sufficient to guarantee the product's success.

In 2016, a leader in the automotive industry, Nissan, suffered a massive setback. The company had to recall over 3 million cars from the U.S. market due to software failure in the airbag sensory detectors. However, it was already a third recall due to the same software error! The first one (2013) included more than 80 thousand cars and the second one (2014) involved almost one million cars.

In 1985, in Canada, a few patients overdosed on radiation as part of a therapy received from malfunctioning Therac-25. The lethal doses of radiation killed 3 patients and left another 3 people in critical state.

Software bugs also caused many other accidents, such as the China Airlines Airbus

A300 crash in 1994, which killed 264 people, and a failed military satellite launch. The latter was announced as the most expensive accident in history ($1.2 billion).

You may not think that the software you want to release can lead to such costly or grievous accidents. However, do you think any of the entrepreneurs in charge of the above projects thought such cases might happen? I bet they did not. That is why, at Sailing Byte, we value testing and will never skip on it to save money or time.

How do we test software?

As you can probably imagine, sufficient testing requires a thorough process and sequence to not miss any important part. Let's analyze the basic steps in the process of software testing.

Basic functionality testing

As the name indicates, such testing is responsible for checking whether all functions of the software work well. It will include clicking all buttons, entering text in every field, checking the basic API functionality (with features designed to be accessed via API), etc.

Code review

It is a good practice to ask for a peer review. Another pair of eyes and a fresh look can uncover many overlooked issues. Functionality errors must be fixed before code review can be performed.

Static code analysis

Static code analysis tools can identify weaknesses, vulnerabilities, and concurrency issues in source code or bytecode without executing it. Thanks to these tools developers can enforce coding standards, and run them automatically as part of the build.

Unit testing

Unit testing tests the proper functionality of a unit/component/method/class across a range of valid and invalid inputs. If the developer works in a continuous integration environment, such tests should be scheduled every time after the

change is made to the source code repository. They should also be run on the development machine. Developers also use mock objects and virtualized services to ensure the unit's independent testing.

The types of tests we run

Apart from the testing process, there are many types of tests developers conduct to ensure their software works as intended. Below, we list the ones we use in our everyday software development processes:

- Black box testing- tests the functionality without any knowledge of its implementation, and with no source code visibility. Testers only know what the software is supposed to do, but not how;
- White-box testing- tests the internal structures of a program, not the functionality given to the end user. For example, PHPUnit is a testing framework for PHP that focuses on the programmer;
- UI/UX testing- UI testing of the user interface of a software or a website to ensure intuitive and easy use; UX testing, on the other hand, focuses on how the product or website affects the user's whole experience;
- Hallway usability testing- tests software with the help of colleagues, and other individuals;
- BrowserStack- tool giving access to a cloud platform that allows developers to test software, websites, and mobile apps across more than 3,500 real mobile devices and browsers;
- Performance testing- conducted to test the speed, stability, and responsiveness of a software program. We in particular test memory load, CPU load, and data load;
- Penetration testing- this security testing focuses on discovering and exploiting weaknesses in software. We implement this testing with the use of tools, such as Kali Linux, SQLMap, and Metasploit;
- A/B testing- compares two versions of a product against each other to determine which one performs better;
- Google Analytics/Tags- by sending data from a given website to linked Google product destinations we can measure its effectiveness and the effectiveness of ads.

Many tests can be automated, which significantly helps in software development by:

- speeding up the testing process compared to manual testing
- allowing to conduct more testing with fewer analysts
- guarantee of consistency thanks to the reuse of the same test cases
- increasing test coverage but not the testing time.

Sailing Byte provides the highest quality thanks to constant validation

Do you want to know the truth about software development? The perfect piece of software… does not exist. It is a myth that many entrepreneurs fall for. Everyone makes mistakes. We are humans and it is one of our main characteristics. That is why every piece of software needs testing to catch as many issues and potential problems as it is possible. The highest quality, especially in such a complex and ever-evolving area as software development, can only be obtained by constant validation. Such constant validation assesses whether the solution is usable and serves planned needs.

At Sailing Byte, we test our software ourselves, and with help from separate testers (manual or automated). With us, you can be sure that your product will be as flawless as possible. Book a call today to discuss all testing steps and methods to which we will subject your piece of software.