QA ESTIMATION GUIDE: How much software testing costs

UTOR

Introduction

In 2021 there's

Google Play

over 3.48 million mobile apps on Google Play

2.22 million on Apple Store

and 1.88 billion web apps out there.

These enormous numbers make for fierce competition on the market: with so many similar solutions available, product quality becomes the main differentiating point.

The importance of proper quality assurance shouldn't be underestimated; however, a poorly estimated software testing process can easily blow your budget. Depending on the project complexity, multiple stages of your delivery pipeline can require different aspects to be tested before it all goes into production.

Naturally, some things can get overlooked, resulting in hidden costs that you might not have counted in initially.

Some poor decisions made along the way can add up to unexpected expenses. Eventually, your break-even price will be affected to the extent that you'll need to alter your go to market.

UTOR has created this guide to help CTOs and CEOs tackle common approaches to estimating the QA budget to avoid the issues mentioned above.

How much software testing costs

Dealing with the software testing budgeting can bring up some reasonable doubts. Are my current spendings enough? When is too much too much? These questions will pop up in your mind fairly quickly.

There is no way to know for sure. To give you a rough estimate - you should probably be looking **at 15% to 25% (or even 50% in some cases)** of your overall project spendings.

It might seem like a lot at first glance. But consider this:

WHAT KIND OF BUSINESS LOSSES SHOULD I COUNT ON DUE TO THE DEFECTIVE PRODUCT?

Your business can probably get on with it, which isn't necessarily true for your reputation. As an aspiring business, you might get your only shot at this.

Pricing models in software testing

You have two ways you can go about doing this:

hire a freelancer

or

use an agency

Both can be valid options and provide various pricing models.

01. Hire freelancer

It will totally make sense to hire a freelancer if your project is pretty straightforward. You'll avoid over-paying and start testing quicker as freelancers typically require less time to get on with it.

However, taking on a new project might be somewhat challenging even for an experienced freelancer. It's not that much about the skill: starting a project from scratch requires team efforts for building effective workflows.

Most freelancers would offer you either **an hourly rate** or **a flat fee.**

An hourly rate is a preferred option for many freelancers. It is a safe way to go, particularly when it's hard to estimate the whole scope of work from the beginning. You will be invoiced upon completing a certain amount of tasks or when a certain threshold is reached (e.g., once per month).

With a flat fee, you agree on the project costs and a deadline, as simple as that. This option would be an excellent choice for hiring experienced freelancers for a one-time project. There's typically a certain amount of down payment involved as well.

All in all, you can use an hourly rate for long-term projects with shifting requirements or a flat fee for one-time projects with a fixed scope of tasks.

02. Use agency

Agencies like UTOR will come of great help for complex projects as they can pull off as many resources as needed, unlike freelancers. They are geared towards building effective operational processes and establishing testing priorities. As the result of this consistent approach, a client gets effective test coverage - something that can't be provided by a freelancer.

However, they take a more comprehensive approach when it comes to estimating the software testing costs.

We've covered a flat fee previously, so let's get a closer look at **time & material** and **dedicated team** models.

Factors for consideration	Time & Material (T&M)	Flat fee	Dedicated team
Project size	Small to medium	Medium	Large
Scope of tasks	Vary	Fixed	Fixed
Client's involvement	Significant	Little	Full
Adaptability	Great	Small	Great

T&M is much the same as an hourly rate approach with an addition of other spendings like licenses, subscriptions, etc.

The dedicated team option is tailored towards the needs of the enterprise companies. As a client, you get a team working on your project exclusively. Also, your dedicated team can be scaled up and down, depending on the direction your project will take.

There might be used hybrid models but, generally speaking, you'll be limited to the options as mentioned above.

QA budget planning criteria

Now you have your options on the table, and it's time to check what can bring your costs up or down

01. The project scale

That's the first and foremost factor that comes into play, and rightfully so. An enterprise and a startup-level application have very different testing needs: an enterprise typically leans towards using dedicated teams, while a startup project is likely to be based on T&M or flat fee model.

The dedicated team option implies sourcing various experts to cover complex project requirements. Here is who you want to see on your dedicated team:

QA team lead is the captain that steers the wheel of your QA team. A team lead doesn't get directly involved in testing (although he can) but ensures the whole team's productivity and deals with any staffing issues.

Test manager decides on the overall approach that needs to be taken as far as testing of the entire project is concerned. It's the test manager's job to distribute the scope of work between all the team members.

Test analyst is someone on your team who is ultimately good with the project documentation. Test analyst helps you navigate the whole testing process by defining priorities and testing methodology.

Test architect knows the app the best and acts as the source of knowledge for the other team members. That's the kind of person you would ask questions about how the product features are supposed to work.

QA engineer is your soldier on the frontline, directly responsible for testing projects and catching bugs.

These people are for a reason on your team, and they ain't cheap. However, it would cost you four times more to fix bugs that made it into production than if they were discovered during the pre-production stage, IBM says.

02. The expected release schedule

Frequent releases are good, but they can add up to your budget: every release requires additional preparation work and testing. With a flat fee, you agree on the project costs and a deadline, as simple as that.

03. The expected product quality

No product is perfect: as much as we want to make it at the top of its game, we're limited by the business constraints. So we need to establish our clear expectations towards the product quality, and how much time and effort we are willing to invest.

Set up your defect allowance: are you concerned only about critical bugs? Or do you want your entire application to be completely bug-free? Depending on your answer, you might need a different extent of test coverage.

04. Data security

How critical is data security to you? If your project involves sensitive data transferring, it's absolutely essential.

Security testing is somewhat different from your typical testing: you need to keep track of constantly appearing and evolving threats to know what you are up against. It's something that can be hard to automate. Therefore, security testing involves lots of manual work, which can be costly.

On the other hand, neglecting security can be devastating. Infinite Money, a trading app, had a production defect that allowed users to use the endless supply of borrowed money. It didn't take too long until someone discovered it and **made $1.7 million out of $3000.**

05. Accessibility testing

Accessibility testing helps your app to be inclusive, making it easier to use for people with additional needs. It usually goes by POUR (perceivable, operable, understandable, robust) principles.

It is also the law: failing to comply with the Americans With Disabilities Act can result in a $75,000 - $150,000 fine.

At UTOR, we perform WCAG 1.0 and WCAG 2.0-compliant testing.

06. The QA documentation

The good news is that the QA team documents everything related to the software testing process. The bad news is that producing documentation costs money. The QA team needs different documents at different stages of software testing, starting from the test plan, test design, checklist, and ending with defect reports.

Some documents are critical so that the QA team can do its job. The documentation can be also used as a reference for the development team, and that makes the interdepartmental interaction easier.

The other documents can result in overspending. At the end of the day, it's up to you to exercise judgment and decide whether you need that amount of status reports.

Allowance for the human factor

The QA processes must be designed with human factors and human capabilities in mind. That means allocating some budget to unforeseen obstacles as the result of human mistakes. Human errors are often simple mistakes yet can be catastrophic to those responsible.

All power of automation won't be able to guarantee that software testing is entirely risk-free. A great (and easy) way to improve the quality of your software – to truly impact the customer experience – is to look at your workplace culture and how you're engineers work day-to-day. Remove distractions, reduce pressure to check code against a deadline. Add a reflection and an evaluation period.

Your trusted QA partner is a phone call away!

Your product quality view is subjective, unlike your budget. Want to know how to strike a reasonable balance between the two? Schedule a call with the UTOR team right away.

QA budgeting can be tricky. UTOR makes it easy!

SCHEDULE A CALL WITH THE UTOR TEAM TO GET YOUR PROJECT QUOTE.