

Sample QA Project

by *Devmountain Tutorials*



WHAT'S COVERED

In this lesson, you will learn how to analyze the steps that the QA engineer takes to make sure that the test plans and cases are sufficient to uncover all of the bugs early in the development process. Specifically, this lesson will cover:

1. Efficiency is Important

Software development is expensive. There are so many people working on so many different pieces of a project, so it is important to be as efficient as possible. For QA, this means finding bugs as soon as possible, or even better, to stop them from showing up in the first place! The quicker we find and solve problems, the less expensive they are.

Mori (UX) just met with the team and showed us his new designs for creating an account and we are getting ready to plan the changes. I am going to review my existing test plan and test cases to identify what changes I need to make and what additional clarification I'll need.



HINT

When changing or evolving software, it increases the likelihood of bugs because the code has to be refactored and it's easy to miss something.

2. Steps of a Test Plan

I start by looking at my test plan. The purpose of a test plan is to get an idea of what I'm going to need to test a feature. What resources do I need? What do I need to do to prepare to test this feature? How will I know when the feature delivers the expected results, not just for my test case but for the users and the company? My existing test plan doesn't need any major changes.

Next, I review my existing test cases. Remember that a test case specifies the inputs, conditions, procedures to test, and the expected results for a feature. With the change in design, I'll need to break up my existing test cases to match the new flow with additional screens. I analyze the requirements to make sure I understand all the changes we are making. We keep our requirements on a wiki page, which is great because it allows us to keep them updated with changes. The requirements are written to explain who is using the feature, what they want to accomplish, and how they should be able to get there. This is useful because it lays out the steps a user should be able to follow to get things done.

If I don't understand exactly what a feature is supposed to do, I won't be able to test effectively, so it's important here, more than anywhere else, to be willing to ask questions. My viewpoint on the team requires me to think from end to end so there is a good chance that if I have a question about something, someone else will have it too—even if they don't know it yet.

When reviewing the new requirements and my existing test cases, I came up with these questions:

- What happens if the user enters a Canadian postal code in the search by zip code field?
- Why does the user have to enter their zip code on the second 'Create Account' screen when they entered it already at the beginning?
- How does the 'Pet Info' screen let a user choose a different weight for each dog?

That last question threw Mori (UX) for a loop, so he is going to go back to that screen and make some revisions.

Once we've worked through the questions as a team, it is time for me to design and implement my tests. This is where I figure out how I want to test it, and actually getting my tests together. There are several test design techniques. Some are more formal than others, so there is some autonomy for each QA Engineer to do what works best for them and their team. I like to have all of my tests stored in a management system where I can go back and remember what I've tested before, and how I did it. This makes it easier for my team to cover my workload when I'm sick or on vacation.



BIG IDEA

You will never find a one-size-fits-all approach to testing—be willing to adapt to different situations and environments. This includes you! Your preferences and skillsets will change how you can best test a feature.



SUMMARY

This lesson explored a sample QA project by discussing the **importance of efficiency** in software development, particularly in the QA process, and the **steps of a test plan**. The lesson emphasized the need to find and solve problems as quickly as possible to reduce costs. The steps of a test plan include reviewing existing test cases, analyzing requirements, asking questions, designing and implementing tests, and storing them in a management system. The lesson also highlights the need for adaptability in testing and the importance of being willing to ask questions and collaborate with the team..

Source: This tutorial was authored by DEVMOUNTAIN and Sophia Learning. Please see our [Terms of Use](#).