

# Thinking Like a QA Engineer

by Devmountain Tutorials



## WHAT'S COVERED

This section will explore a sample QA project and how a QA engineer thinks by discussing:

1. EFFICIENCY IS IMPORTANT
2. UPDATING A TEST PLAN
3. CHALLENGES FOR QA ENGINEERS
4. LEARNING OPPORTUNITIES FOR QA ENGINEERS

## 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. UPDATING 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.



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.

---

## 3. CHALLENGES FOR QA ENGINEERS

While overall, I love my job, sometimes testing manually can get boring, especially if I need to do the same thing multiple times. If I get too bored, I know I'm doing it wrong. I can take a break, test something else, or change up my testing approach.

Taking the time to keep myself engaged helps keep me alert to spot problems. Managing and maintaining test cases can be a bit bland, but I've started having jam sessions with Jose (Product Manager) where we update our documentation together and it makes it more fun.

I love that we have the autonomy to control how we do our jobs. I'd rather take a proactive approach to my test cases so I'm not delayed by out of date information when I'm trying to execute a test.

My least favorite aspect of the job is when I have to speak up because I disagree with moving forward on a release. It can be intimidating to disagree when everyone else is on board to release a new feature. I make sure to let everyone know about the problem I found and the likely impact it will have on users. Usually, our team is able to have a conversation about the risks and come to a consensus, but that doesn't always happen. The more we work together, the more we are learning and the better we get.

Like many QA Engineers, I've found it challenging to test the different types of products we have. Even though the experience for users between the website and the mobile app feels very similar, the technology that powers them is different and that requires a different approach to testing.

---

## 4. LEARNING OPPORTUNITIES FOR QA ENGINEERS

Since Camilla helped me to see that there are many options out there, I've been doing research online to find tutorials on different types of testing.

Here is a list of topics I'm learning more about:

- UI Automation, Web, Mobile, and Desktop
- Continuous Integration and Delivery
- API Testing
- Database Testing
- Contract Testing
- Security Testing
- Performance Testing
- Accessibility Testing
- Usability Testing

And that's only a few of the topics I've found. An engineer can specialize in any of them. The more ways I can test out our product, the happier our customers will be, and the more valuable I am to my team. I've been focusing a lot on automation to speed up the testing process. I was surprised to learn that even great automation can't test everything. Usability issues are identified most effectively by manual testing. I've had a few peers transition from being QA Engineers into various development positions. An entry-level QA position was a great starting point for their careers.

---