

Name: Chun-Chi Huang
Course: Software Testing
Professor: Michael McKee
Date: 04/19/2018

Chapter 2: Thinking Like a Tester

The lesson I chose is “Exploring involves a lot of thinking.” The lesson states that testers should act as detectives, and they need to broadly observe the situation, carefully discover the clues of any suspected points, and find exact relationships between causes and effects. Three ways mentioned in the book are forward thinking, backward thinking, and lateral thinking. First, forward thinking means that testers explore from what they know to what they do not know. In other words, the test procedure is based on what testers see and do for the test. For example, there is a camera app, Snapspeed, which is a popular photo-editing application used on smart devices. When someone launches this application, there is a ‘tool’ option on the screen. As a tester, what I have to do is to click and see what will happen in this application. Moreover, backward thinking indicates that testers can begin from something they imagine or suspect to the things they already know. They attempt to verify the assumption or identify whether there is a bug. For instance, with Snapspeed, before I start utilizing this application, the knowledge of photo-editing functionalities already exists in my mind, such as quality-tuning, space-tuning, text-addition, etc. I already know that these functions should be embedded in this application. Consequently, I can start testing this application according to my previous knowledge and explore the things that exist in this application. The third one is lateral thinking, which suggests that testers should think from lateral sides. For example, what is the effect of this camera application on a bigger screen like a tablet?

There is one more idea that I want to mention not in the lesson. It is top-down thinking. Testers are usually assigned to test an individual function. However, if they know the concept of the whole system not only the particular application, it is more efficient for them to find issues. This reminds me of an experience in my previous project. The project was related to software development for tablets, and I was responsible for the integration of device drivers. The manager of QA team assigned his members to test drivers by function, which also meant that each QA member only tested one or two particular drivers. After their testing, all the test items passed, and no bugs were found. However, something I still suspected was a conflict between the camera and wifi drivers because they shared the same bus. Consequently, I informed the QA manager to arrange cross-function tests for these two drivers. Finally, the QA members found that when the end user used the camera, the throughput of wifi decreased. Fortunately, this issue was detected by our team, not by our customers. The more knowledge testers have, the more efficiently they can find issues. Currently, we are focusing on software. I hold the belief that understanding the concept of the whole software system is very important for not only developers but also testers. In the end, I agree with this lesson, “Exploring involves a lot of thinking” because testers should continue to learn different kinds of thinking to improve their job quality of software testing.