Chapter 1

What’s Special about Mobile Testing?

Before I start describing the unique aspects of mobile testing, I’d like to share a true story with you.

What’s special about mobile testing? Someone asked me this exact question several years ago while at a testing conference. I started talking about mobile technologies, apps, how to test them, and what’s special about mobile testing. The guy simply smiled at me and said, “But it’s software just on a smaller screen. There’s nothing special about it.” He was really arrogant and didn’t see the challenges presented by mobile testing. No matter which arguments I used to convince him, he didn’t believe in the importance of mobile technologies, apps, and testing.

I met the same guy again in 2014 while at a testing conference where he talked about mobile testing. He spoke about the importance of apps and how important it is to test them.

As you can see, it’s very easy to underestimate new technologies. As a software tester it’s especially helpful to be curious about learning something new and exploring new technologies to broaden your skills.

So let’s come back to the initial question: What’s special about mobile testing? I think I can assume you have at least one mobile device, namely, a smartphone. Or maybe you have a tablet, or even both. If you look at your device(s), what do you see? Just a small computer with little shiny icons on its screen? Or do you see a very personal computer with lots of sensors and input options that contains all of your private data? Please take a minute to think about that.

My smartphone and tablet are very personal computers that hold almost all of my data, be it e-mails, SMS, photos, music, videos, and the like. I can access my data no matter where I am and use my smartphone as a navigation and information system to find out more about my surroundings. For that reason I expect my apps to be reliable, fast, and easy to use.

In those three sentences I described my personal expectations of mobile devices and apps. But you may have entirely different expectations, as does the next person. And this brings me to the first special characteristic or unique aspect of mobile testing: user expectations.

User Expectations

In my opinion, the user of an app is the main focus and main challenge for mobile teams. The fact that every user has unique expectations makes it difficult to develop and deliver the “right” app to customers. As several reports and surveys have shown, mobile users have far higher expectations of mobile apps than of other software such as browser applications.[1] The majority of reports and surveys state that nearly 80% of users delete an app after using it for the first time! The top four reasons for deletion are always bad design, poor usability, slow loading time, and crashes immediately after installation. Nearly 60% of users will delete an app that requires registration, and more than half of users expect an app to launch in under two seconds. If the app takes more time, it gets deleted. Again, more than half of users experience crashes the very first time they start an app. An average user checks his or her mobile device every six minutes and has around 40 apps installed. Based on those numbers, you can deduce that mobile users have really high expectations when it comes to usability, performance, and reliability. Those three characteristics were mentioned most often by far when users were asked about their experience with mobile apps.

Currently there are more than two million apps available in the app stores of the biggest vendors. A lot of apps perform the same task, meaning that there's always at least one competitor app, which makes it very easy for consumers to download a different app as it's just a single tap away. Here are some points you should keep in mind when developing and testing a mobile app:

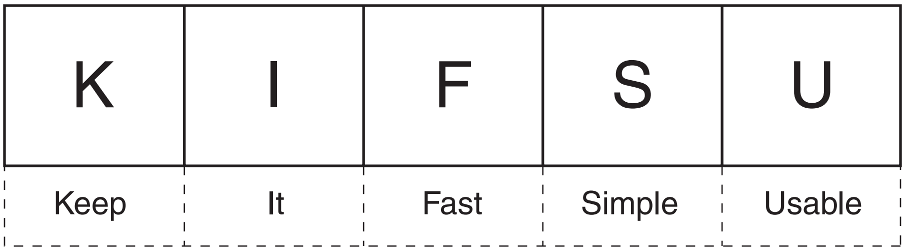
- Gather information about your possible target customer group.
- Ask your customers about their needs.
- Your app needs to solve a problem for the user.
- Usability is really important.

[1] http://offers2.compuware.com/rs/compuware/images/Mobile_App_Survey_Report.pdf

- Your app needs to be reliable and robust.
- App performance is really important.
- Apps need to be beautiful.

There are, of course, a plethora of other things you should take into account, but if you pay attention to these points, your users are likely to be happy.

You've probably already heard of the KISS principle.[2] KISS is an acronym for Keep It Simple, Stupid and is always a useful reminder—especially for software projects—to not inflate the software with just another function or option. Keeping it small, easy, and simple is best in most cases and is likely to make your customers happy. Inspired by KISS, I came up with my own principle for mobile apps: KIFSU (see Figure 1.1). This abbreviation is a good mnemonic to help you cover customer needs and a constant reminder not to inflate apps with useless functions.



K	I	F	S	U
Keep	It	Fast	Simple	Usable

Figure 1.1 *KIFSU*

Mobility and Data Networks

Another challenge mobile apps have to deal with more than software running on computers is the fact that users are moving around while they use apps, which often requires an Internet connection to fetch data from the backend and serve the user with updates and information.

Mobile apps need to be tested in real life, in real environments where the potential user will use them. For example, if you're testing an app for snowboarders and skiers that accesses slope information, one that is able to record the speed of the current downhill run and makes it possible for users to share records directly with their friends, you need to test these functions on a slope. Otherwise you can't guarantee that every feature will work as expected.

[2] http://people.apache.org/~fhanik/kiss.html

Of course, there are parts of an app that you can test in a lab situation, such as slope information availability or whether or not the app can be installed, but what about recording a person's speed, the weather conditions, or the Internet connection at the top of a mountain?

The weather conditions on a mountain, in particular, can be very difficult to handle as they can, of course, range from sunshine to a snowstorm. In such scenarios you will probably find lots of bugs regarding the usability and design of an app. Maybe you'll also find some functional bugs due to the temperature, which may have an impact on your hardware and, in turn, your app.

As I already mentioned, the speed and availability of Internet connections could vary in such regions. You will probably have a good network connection with high speed at the top of the mountain and a really poor one down in the valley. What happens if you have a bad or no Internet connection while using the app? Will it crash or will it still work? What happens if the mobile device changes network providers while the app is being used? (This is a common scenario when using apps close to an international border, such as when snowboarding in the Alps.)

All of these questions are very hard to answer when testing an app in a lab. You as a mobile tester need to be mobile and connected to data networks while testing apps.

As you can see, it's important to test your app in real-life environments and to carry out tests in data networks with different bandwidths as the bandwidth can have a huge impact on your app; for example, low bandwidth can cause unexpected error messages, and the switch between high and low bandwidth can cause performance issues or freezes.

Here's an exercise for you. Take any app you want and find three usage scenarios where the environment and/or network connection could cause problems.

Mobile Devices

Before you continue reading, pick up your mobile device and look at it. Take your device in your hand and look at every side of it without turning it on. What do you see?

You will most likely see a device with a touch-sensitive screen, a device with several hardware buttons with a charger, a headphone connection, and a camera. That's probably it—you're not likely to have more than five hardware buttons (except for smartphones with a physical keyboard).

In an era when the words *cell phone* have become synonymous with smartphone, it's important to remember that there used to be other types of cell phones, so-called dumb phones and feature phones that have lots more hardware buttons for making a call or typing a message. With a conventional dumb phone you are only able to make a call, type a message, or store a contact list; they're not usually connected to the Internet. The more advanced ones, the feature phones, have games, a calendar, or a very basic Web browser with the option to connect to the Internet. But all these phones are really basic in terms of functionality and expandability as users aren't able to install apps or easily update the software to a newer version, if it all. Both types of phones are still available, especially in emerging markets, but since 2013 more smartphones have been sold worldwide than dumb phones or feature phones,[3] and this trend is likely to continue as time goes on. In fact, in the next couple of years dumb phones and feature phones will be a thing of the past.

The phones we use nowadays are completely different from the "old" ones. Current smartphones are mini supercomputers with lots of functionality in terms of hardware and software. They're packed with various sensors such as brightness, proximity, acceleration, tilt, and much more. Besides that, all modern smartphones have both front- and rear-facing cameras, various communication interfaces such as Bluetooth, near field communication (NFC), and Global Positioning System (GPS), as well as Wi-Fi and cellular networks to connect to the Internet. Depending on the mobile platform and mobile manufacturer, you may find an array of other hardware features.

From a software point of view, smartphones offer lots of application programming interfaces (APIs) for manufacturers, developers, and users to extend smartphone capabilities with apps.

If you just focus on the major mobile platforms, iOS and Android, there are plenty of hardware and software combinations that mobile testers have to deal with. The fact that there are so many combinations is known as fragmentation. Mobile device fragmentation is a huge topic and yet another challenge when it comes to mobile testing.

You can't test your app with every possible hardware and software combination. And the fact that you should test your app in a real environment makes it even more impossible. Mobile testers need to find a strategy to downsize the effort of testing on different devices and to find a way to test on the right devices.

But how can that be accomplished? By testing on just one mobile platform? By testing on just the latest device? By testing with just the latest software version?

[3] www.gartner.com/newsroom/id/2665715

Before you define a strategy, you should keep in mind that every app is unique, has unique requirements, has other problems to solve, and has a unique user base. With these points in mind, you can ask yourself the following questions to find the "right" mobile devices for testing:

- Who is my user base?
- How old is the average user?
- How many men or women are in my target user group?
- Which platform is used most among that user base?
- Which device is used most?
- Which software version is installed on most of the phones?
- What kind of sensors does my app use?
- How does the app communicate with the outside world?
- What is my app's main use case?

Of course, there are lots more questions to ask, but if you answer most of the ones I suggest, the list of possible devices you should consider testing is much shorter.

In later chapters I will describe other techniques for selecting the right devices for mobile testing.

Mobile Release Cycles

Now that you know how to find the right devices for testing your app, it doesn't mean that the process is over. To be honest, it's never going to end!

The main mobile manufacturers release a new flagship phone with more features every year. In and around those releases they bring out other phones for different user scenarios and user groups. This is especially true in the Android world where every new phone comes with a new version of the operating system packed with new features, designs, or APIs. There are multiple software releases within the course of a year, ranging from bug fixes to feature releases. You as a mobile tester need to be sure that your app will run on the latest hardware and software.

But how should you handle these situations? By buying every phone that appears on the market? By constantly updating to the latest operating system version?

Again, the most important factors are your target customer group and the app you're testing. When you know that your target group always uses the latest and fastest phones on the market, you need to buy those phones as soon as they appear. Regardless of whether or not your target group is up-to-date, you should always monitor the mobile market.

You need to know when the main vendors are due to release new flagship phones that a lot of people are likely to buy. You also need to know when the operating systems receive patches, new features, or new design patterns.

So the answer to the question of whether you need to buy every phone and constantly update the operating systems is yes and no. Of course you don't need to buy every phone that's on the market, but you should consider updating to the latest operating system version. When doing so, keep in mind that not every user will install the update. Many people don't know how to do that, or they don't care about new versions. You need at least some phones that are running older versions of the operating system to see how the app reacts in that environment. Older versions of the operating system are also needed to reproduce reported problems and bugs.

A good way to manage all this is to stick with the same operating system version on the phones that you have and buy new phones with the latest software version. This of course leads to another problem—it's really expensive! Not every manager wants to spend so much money on mobile devices when a phone is going to be used for only a couple of months. A solution for that is to rent devices. There are several providers and Open Device Labs where you can rent a device for a certain period of time (a list of providers can be found in Chapter 3, "Challenges in Mobile Testing"). Another way to rent devices is the mobile device cloud as there are a number of providers who give mobile testers exclusive access to the physical devices they have made available in the cloud. Just use your search engine and check them out.

In the mobile projects I've worked on, we always had the top ten to 15 devices used by our target user group in different variations for developing and testing. This was a good number of devices that covered nearly 90% of our target group. With those ten to 15 devices we were able to find most of the critical bugs; the remaining 10% of devices we didn't have were of no major consequence to the project or user expectations.

In order to handle the fast pace of mobile release cycles, you should keep the following things in mind:

- Monitor the mobile device and software market.
- Know when new phones will be rolled out.
- Find out about the new features of the operating systems.
- Keep an eye on your target customer group to see if new devices are showing up in your statistics.

- Think twice before updating a phone to the latest operating system version.
- Buy new phones with the latest operating system version.
- If buying is not an option, rent the devices.

Updating, buying, and maintaining all of your devices is a challenging task and should not be underestimated! At some point, depending on the number of test devices used within a project, this could be a full-time job.

Mobile Testing Is Software Testing

Let's come back to the story I told at the beginning of this chapter when the guy at the conference didn't believe in the importance of mobile testing. He had the attitude that mobile testing is not real software testing. In his opinion, mobile apps were only small programs with less functionality and no real challenges when it comes to software testing. But this is definitely not the case. If you look at the topics I described in this chapter, you should have an initial impression about the challenging job of a mobile tester. Mobile testing is totally different from testing software applications such as Web or desktop applications. With mobile apps, physical devices have far more influence over the software that is running on them when compared to other software such as Web applications. Because there are so many different smartphones available on the market, mobile testers need to focus a lot more on hardware during the testing process. In addition, users moving around and using different data networks force mobile testers to be on the move while testing.

Besides the hardware, user expectations play an important part in the daily business of a mobile tester and need to be taken seriously.

There are many more topics and issues mobile testers need to know about in order to help the whole team release a successful app. The rest of the chapters in this book will cover the following topics:

- More challenges for mobile testers and solutions to those challenges
- How to test mobile apps systematically
- How to select the right mobile test automation tool
- The different concepts of mobile test automation tools
- How to find the right mobile testing strategy
- Additional mobile testing methods
- Required skills for mobile testers

Keep the topics from this chapter in mind as a starting point. Keep your app simple and fast (remember KIFSU). Test while you're on the move, and test on different devices based on your target customer group.

Summary

The first chapter of this book mentioned some very important topics from the mobile testing world. As you have seen, mobile testing is completely different from testing on other technologies such as laptops or desktop computers. The biggest difference between mobile and other technologies is that the mobile user is on the move while he or she is using your product. Therefore, it is very important to know about the different data networks and the different types of mobile devices.

This chapter also provided a first overview of mobile users' high expectations. It is really important to keep KIFSU in mind when designing, developing, and testing a mobile app. It will help you to focus on the important elements and not waste time on unnecessary features that your users won't use.

And last but not least, this chapter should remind you to never underestimate a new technology. Be open-minded and curious to improve your daily work life.