

Mobile Testing Tips

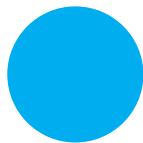
Experiences & Realities

Bařış Sarıalioğlu

Keytorc Software Testing Services, Turkey

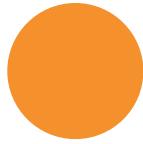


Is This Book Right For Me?



INTRODUCTORY

Introductory content is for software testing professionals who are relatively new to the subject matter of the ebook. Introductory ebooks typically feature an overview of an aspect of testing or guidance on understanding the fundamentals of the subject at hand.



INTERMEDIATE

Intermediate ebooks are for software testers who are already familiar with the subject matter of the eBook but have limited hands-on experience in this area. Intermediate level usually focuses on working examples of a concept or relaying experiences of applying the approach in a particular context.



ADVANCED

Advanced ebook content is for software testers who are, or intend to become, experts on the subject matter. These ebooks look at more advanced features of an aspect of software testing and help the reader to develop a more complete understanding of the subject.

Some testers believe that mobile testing requires a completely distinct approach deployed by distinct techniques, methodologies and tools; while some others believe that it is nothing but a contextually different traditional software testing.

In Baris's Book, '*Mobile Testing Tips*' you can read thirty different tips to help you conquer your mobile testing challenges.

In this eBook we'll take a closer look at three of these tips:

- Tip 2: ***Grasp the Challenges of Mobile Testing***
- Tip 9: ***Mobile Testing Principles and Strategies***
- Tip 28: ***Mobile Usability Testing is Not Rocket Science***

Biography



Barış Sarıalioğlu

Keytorc Software Testing Services, Turkey

Barış Sarıalioğlu has over ten years of experience as an information systems professional. He is highly experienced in software development life cycle, project management, Agile development, quality assurance, and software testing.

Barış also has diverse experience spanning several industries, including telecommunications, defense, banking and finance, semiconductor manufacturing, and aviation. Based on this broad experience, he has been involved in several challenging projects and had the chance to work in several different countries, including Turkey, the United States, Russia, Germany, China, and Greece.

He has written articles and papers on software development methodologies, quality assurance, and software testing, and he regularly attends international and national conferences as a speaker, panelist, moderator, and contributor. In 2013, he published his first book, *Software Testing Tips; Experiences & Realities*, in which he shared his experiences and vision about various software testing issues.

Currently, he is one of the managing partners of Keytorc Software Testing Services, where he holds the responsibilities of delivering test consultancy, outsourced test management, and software testing training.

To the new member of our family,
my baby girl, Sırma.

CONTENTS

Tip 2: Grasp the Challenges of Mobile Testing

Tip 9: Mobile Testing Principles and Strategies

Tip 28: Mobile Usability Testing is Not Rocket Science

Keytorc People Answer the Question “What Is Mobile Testing?”

TIP 2

GRASP THE CHALLENGES OF MOBILE TESTING

The greater the obstacle, the more glory in overcoming it.

—Molière

Life is a challenge, meet it.

—Mother Theresa

Every problem has a gift for you in its hands.

—Richard Bach

Sorrow prepares you for joy.

—Mevlana Rumi

GRASP THE CHALLENGES OF MOBILE TESTING

Just after getting used to the terms “mobility” and “mobile context,” we need to spend some time on the challenges. Yes, you have heard it correctly: as you can guess, there are a number of challenges ahead of you before you can be successful, and these challenges are really tough.

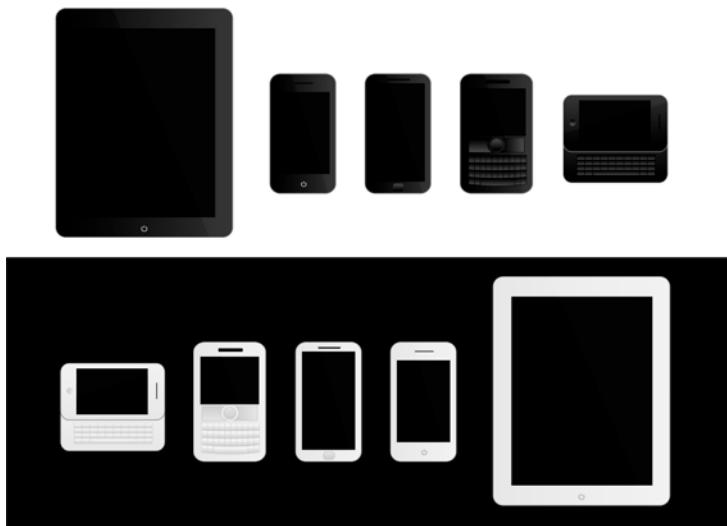


Mobile testers generally underestimate this fact. They need to know that without clearly defining the challenges, one cannot say that he or she is mastering mobile testing. Project management courses also underline this fact very clearly. Know your challenges to build up your strategy and a fitting approach. A trustful action plan is nothing but a good reflection/resolution of the challenges ahead.

Well then, let's now try to list the challenges of mobile testing. In order to be more clear here, I will divide these challenges into two areas. One will be the “generic challenges,” which include management, strategy, and human resources (HR) issues. The other area encompasses “test-specific challenges,” which are more related to testing itself.

Generic Challenges

Number of platforms—When we observe the timeline of the most recent mobile operating system (OS) releases, we see that iOS, Android, Windows, and BlackBerry dominate the market. They have the most recent versions running in numerous devices with frequent OS updates and device-specific features. As these common platforms are widely used, they should be tested accordingly. I advise every tester to regularly check the “Application Developers Alliance” website and stay informed about the recent releases, news, and statistics of mobile platforms.



Number of devices and displays—According to a research done by Application Developers Alliance last year, there were about fourteen thousand distinct Android devices in the market! Too many, isn't it? When we look at it from a tester's perspective, this can be an unreachable and unrealistic target. Since we, the testers, have limited time and budget, we need to be selective in our tests. In the following sections, I will propose some methods to cope with device selection and coverage.

Surely, distinct devices also mean different displays. Mobile testers should also take this into account. Different display size means different testing and different priorities. So I suggest you regularly search for “list of displays by pixel density” on Wikipedia so that you

can keep yourself up-to-date regarding market devices and their display specifications.

Mobile hardware is complex—Believe it or not, mobile devices are complex, especially the hardware. Within a very small space, there are numerous microelectronic devices, and all should be interoperable. There is a processor, a solid-state disk, backlights, loudspeakers and earpiece, a Bluetooth module, USB peripherals, a battery, a GPS antenna, a touchscreen, and several other sensors and microchips.

When you add these up, you have a complex formula. As a mobile tester, you need to understand the hardware complexity while you are testing any mobile device. In order to better understand the hardware side, you may find that regularly checking ifixit.com is useful in understanding the physical parts and their importance in testing.

Mobile software is complex—Not only is the hardware complex, but mobile software often possesses detailed features and architecture—especially if we are talking about server-client applications. Many distinct technologies should be taken into consideration. How should server and client talk to each other? How should the database be integrated and accessed? How are the web services oriented. How many screens are there? Can the application work offline? Does it have authentication/authorization? The list can be endless.

If you are testing a mobile application, you at least need a basic understanding of the technologies that have been used in developing such an application, the architecture that the app relies on, and the feature set that the application brings. Without this information, testing cannot be done effectively, especially the lower level tests such as unit and integration testing.

Mobile security is critical—Security has always been a headache for software producers. When we are talking about mobile applications, it is one of the biggest challenges. Every year, statistics indicate a common fact: mobile operating systems are being targeted by malware, and this equates to an increased security risk. When we consider the fact that each smartphone or tablet uses about one

hundred times more data than a standard feature phone, this equation becomes even more critical.

The risks associated with any software are a main driver of software testing; the mobile testers need to focus on this high-risk subject, that is, security testing.

Carriers/Operators are important—Due to lack of resources, know-how, and funding, mobile testing is usually done from noncarrier networks, for example, WAN, LAN, or Wi-Fi. This is a great problem since there are many network/carrier-related factors influencing mobile apps while they are running. Without moving some part of your testing into the wild by using real carrier networks, you cannot be 100 percent sure about your test coverage.

We all know that mobile users are on the go and not always connected to Wi-Fi. Therefore, we need to include the carrier network dynamics in our testing activities. We need to test while we are using 4G, 3G or 2G, we need to test while we are roaming, we need to test while we are tethering to the Internet, we need to test while we are having limited or poor access, and we need to test the interruptions caused by incoming calls, SMSs, MMSs, e-mails, and so on. All these carrier issues are important and are indispensable for mobile testing activities.

Mobile devices have limitations—Some people think that a mobile device is a portable/miniature computer. I agree with this perception but only to a certain extent. We need to know that these devices have some significant limitations when compared to PCs. They have smaller screens and reduced bandwidth, they have no mouse or standard-size keyboards, their processors and memories are less capable, their power management is different, they have smaller disk spaces, they mostly have touchscreens, they generally run a single application from a single screen, they have shorter sessions, and their users are moving.

While you are testing a mobile device, you need to remember these limitations. Do not forget to define your limitations and differences before starting your tests. In this way, you can be more confident about your testing and its outcomes.

Native vs. mobile web—One common discussion in mobile application development is the following: “Should we have native apps, or shall we use mobile web?” The answer is not so easy. A political answer like “it depends” fits here.

Both ways bring advantages and disadvantages. For example, a native app may have more advanced UI features, and it has more access to device-specific features (e.g., vibration, localization, backlights, camera, sensors, etc.). On the other hand, mobile web has upgrade flexibility and code portability. As testers, we need to understand the needs of our users/organizations, and we need to test the mobile apps accordingly, especially by taking into consideration the related technologies and their differences.

We have just defined our framework, so now let's move on to the test-specific challenges.

Test-Specific Challenges

Need for shorter development life cycles—Since mobility translates into very demanding and ever-changing user behavior, software producers should adapt themselves to this context. In order to fulfill customers' needs and penetrate the market quicker, mobile application development should be done with shorter life cycles with fewer requirements and features. Having fewer requirements per cycle significantly reduces the risk of having unhappy users. In order to achieve this, companies try to adopt agile development and be more responsive. They do frequent releases with narrow scopes instead of a wide-ranging one. Mobile testing should be adapted to this very demanding and agile context, and testers should be more responsive and creative. As we always complain about the lack of remaining time for testing, we need to be careful here. We need to deploy a risk-based testing approach and try to focus our testing efforts on the most commonly used features and business rules.

Need for regression testing—After we talk about shorter life cycles, we need to talk about regression testing. As we will have more frequent deliveries, we should do more regression testing. With each and every change in the code, we need to be sure about the status of the legacy (predecessor) functions and/or features. These changes can be new features and/or defect fixes, and both should be very

carefully tested.

In the following sections, I will further emphasize the importance of mobile test automation and will show how it fits with regression testing by referring to several case studies.

Need for back-end testing—Unfortunately, mobile testers focus a lot more on the UI (front end) rather than APIs or any back-end layers. When I quantify this figure through several case studies, I observe a ratio of around 90 percent. Generally, mobile applications are parts of bigger systems. For example, consider a mobile banking application: the back-end code is likely to have dependencies, coupling, and shared code with the core banking system.

If we test a mobile application with a similar architecture, we need to test the back end as well as the front end. Especially, application programming interfaces (APIs) should be checked by test cases, and in this way, business and database layers should also be checked.



Need for performance testing—One other widely neglected test activity is performance testing. As nonfunctional requirements are rarely collected, developed, and traced in software projects, nonfunctional testing is unfortunately positioned as a low-priority issue. If you ask mobile software producers, none will agree with this, but you cannot hide the truth. Almost all the performance-related bugs are being resolved in production environments! In

other words, companies are not readily investing in performance testing unless they experience a hurtful performance failure in their live systems.

Mobile testers should be aware of this fact, and they need to promote performance testing. Lots of research is being done in this area, and the results show that mobile users are impatient. You cannot tell them to wait for ten seconds for your mobile app to launch and that afterward they will ascend to heaven. Industry benchmarks show that mobile users want apps to launch in two seconds! Yes, you are reading it correctly: just two seconds. Simply put, mobile users want us to do performance testing; otherwise they will do it for us and then just delete the apps we have developed.

Need for mobile testing tools—This is a generic point in any software testing problem and tool selection. When we shift our focus to mobile testing, it just grows bigger. In order to reduce the risk of mobile application failures and to improve traceability, coverage, reporting, and execution frequency, you need to select the right mobile testing tools at the right time.

Recently, we have observed significant improvements in this area. Mobile testing tools are on the rise, and every year new players are being introduced to the market. It seems that tool scarcity, which was the biggest challenge two to three years ago, is now out of the picture. However, this does not mean that tooling is no longer a challenge. We still know that organizations are suffering when choosing the right tools. They have more shelfware (not used/utilized software) than usable testing software. Since lots of solutions exist, there is no one right option. And mobile testers should own the responsibility to solve this equation and find the test tool they will use.

Need more time to test—Another big challenge in any tester's life is the shortage of time. For several years, I have been delivering training on software testing and mobile testing and telling people to test early and often—but still, there's not much improvement! Especially when I speak to upper managers and C-levels, they tell me that they are impressed and promise to give more time to testing, but it never happens. The latest life-cycle activity seems to be software testing, and if you are dealing with mobile applications, this behavior can make you suffer more.

A responsible tester should estimate rational efforts and promote quality rather than time-to-market. So never hesitate to ask for more testing time if you don't feel comfortable. OK, you need to think twice if you are talking to your CIO, but you should know that thinking twice doesn't mean never saying a word!

Need more devices and environment—Whether you do in-house testing or outsource it, it is always useful to have an in-house test environment (a lab or a simple test place) and to have some real devices in place. Surely you can use cloud solutions and emulators instead of real devices, but you need to know that nothing compares to a real mobile device. If you are doing system or user-acceptance levels of testing and the devices are owned by you, testing becomes more and more real, and you capture more real bugs.

Need for skilled mobile testers—Mobile testing is a very big and a promising market, but companies still suffer from a lack of talented people. If you want to be famous, you can try your luck and go for a mobile tester position! Since traditional approaches and regular ways of testing will probably not make anyone happy, creative, talented, and educated testers are needed.

Need to be on the move—When people invite me to their mobile testing environments, I observe lots of tables full of mobile devices, and testers are running their tests from their chairs and marking their results on the test management tool. Most of the time they use a stationary testing approach with no movement at all! The sad truth is that they are not aware of what they are missing. Mobile applications are being used by people who are on the move. I don't see any people around me who stop walking and sit on a chair to watch a video, tweet, or send an e-mail. All these things are done on the go.

Since we mobile testers are defenders of mobility, we need to devote some time to doing at least some of our testing on the move. If a testing activity does not reflect, exercise, or contain any real user behavior, then we can not conclude that it is 100 percent real testing.

After reading this section, if you feel that being a mobile tester will be a real pain for you, you can try your chances at several other jobs. As a famous saying goes, "If you are afraid of failure, you don't

deserve to be successful." Alternatively, I can be more constructive and end this section with words from a famous U2 song, "Miracle Drug":

*Of science and the human heart, there is no limit
There is no failure here, sweetheart, just when you quit...*

TIP 9

MOBILE TESTING PRINCIPLES AND STRATEGIES

Obey the principles without being bound by them.

—Bruce Lee

Failure comes only when we forget our ideals and objectives and principles.

—Jawaharlal Nehru

MOBILE TESTING PRINCIPLES AND STRATEGIES

This section will focus on the principles and strategies of mobile testing. You may adapt these principles to almost every mobile testing project you deal with. Do not hesitate to use them if you are under pressure because of a manager who has no idea about mobile testing.

- Do early testing by getting involved early in the projects and try to do as much static testing as possible. Project managers have a tendency to reject one or two days of static testing because of the time it consumes. On the hand, the same project managers are willing to postpone the same project for twenty more days because dynamic testing is not properly finished due to unverified bugs that could have been identified with static testing!
- Do not test everything. Some testers are heroic and say that they need to test everything or test the product 100 percent. Please calm these people down and tell them that 100 percent testing is impossible. Rather, you may propose risk-based testing, which is a far better and achievable approach.



- Dedication matters, so let mobile testers test your apps. I find it difficult to understand the fact that business analysts get offended when you say you need a tester to test any product. But when the time comes to testing, they try to stay away

from it and say that they have lots of other high-priority tasks and it would be better if they focus instead on their analysis/design assignments. Ultimately, do not believe in crocodile tears; employ your mobile testers.

- Testers are happy to find bugs because they are paid for that. Do not assign a different meaning to their pessimistic, suspicious, and questioning characters. They find defects for living, not for getting developers fired.
- If you cannot use a mobile app, do not blame yourself; blame technology. Don't forget that technology is and should be created for humans, and that humans are not created for technology.
- Defects do not stay in public areas; instead they like to hang out in forgotten places and live their lives on the boundaries. If you expect to detect a running horse mistakenly put on the splash screen of a mobile banking application, do not wait in vain. People would detect the horse before it appears to you.
- Found defects are an indication of the presence of many others. Never ever say, "I found three defects in module X, so let's test other modules. We are safe in module X." In such case, remember ant nests. When you look at the nest from the top you see only a few ants. But when you dig into the ground, you see hundreds of ants. Defects are like ants. Most of the time they are invisible, and they stay and act together.
- Defects are not always program malfunctions; sometimes they can be the things that just bother you. When you are testing a mobile app, you should not only seek functional deficiencies; sometimes defects can be the annoying, silly things rather than program malfunctions. Some developers like to say, "This is not a defect; it is working." Do not believe in such illusions. Report the bugs.
- In the mobile world, you need to do usability and performance testing more than in any other category. Users are impatient, and they are not primarily focused on functionality. Rather, they are in search of simple and user-friendly products.
- Do not try to automate everything; user behavior should also be given priority in most cases. In the mobile world, any test case can be automated, but if you are looking for real feedback, you need to do manual testing. Don't get upset; this is not the end of the world, and no one is promoting you if

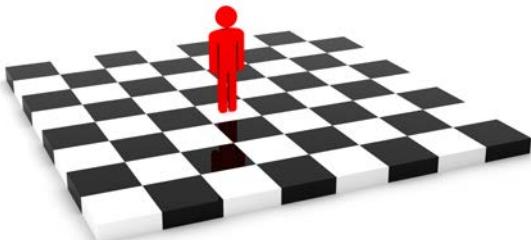
you automate some nonsense test case.

- Do not praise your testers by the number of test cases they prepare in a given period of time. Instead, you need to assess them by the defects per test cases ratio. In this case, you should remember that bigger is not always better (at least in testing, it is not!). While testers are preparing test cases, they generally race against time. Consequently, preparing defect detectable test cases is much better than preparing hundreds of passing (developer lover) test cases. Encourage testers to detect more defects with fewer test cases.
- Try to include some negative testing scenarios in your test suites. Happy-path testing will not find you defects after some time.
- While you are doing usability testing, you need to soliloquize the phrase “more is less.” If an app contains many fields, lots of unused/hidden functionalities, and numerous controls, there is a large probability that it will make its users unhappy. Mobile apps should not create any paradox of choice while they are being used by their users. By the way, I suggest you read the book *The Paradox of Choice* by Barry Schwartz. If you do, you will understand the “more is less” phenomenon better.



- Do not forget to perform claim testing, an activity checking whether the app store claims (app descriptions) are fulfilled by the app itself. We know many apps that claim they save the world and will show you the heaven, but that in reality suck. As a mobile tester, you need to reveal this as well.

- Be sure to stay away from “dark side” while you are testing. Remember the movie *Star Wars: Episode III—Revenge of the Sith* and the character Anakin Skywalker. He was born to be on the good side (Jedi force), but after he passed to the dark side, he became Darth Vader. If you are a mobile tester, you need to report the bugs (even though they are to be fixed in the afternoon release), announce valid metrics (despite that they are pointing you to some unattended test cases), and be truthful. If you falsify any data or do not report a defect because you are in love with a developer, you enter the dark side. Unfortunately rest will be easier. Before you know it, you will become the Darth Vader of your organization. As Master Yoda said, “Do not let your feelings towards developers cloud your judgment”. Be careful!



- Regrettably, the mobile world is evolving a lot faster than mobile testers. This means that talented and skillful mobile testers are needed more than ever, and the gap between the mobile world and testers will be widened unless organizations train their testers and make the necessary investments. However, if you are a mobile tester and you are waiting for your company to arrange the necessary training for you and continuously invest in your development, you are in an imaginary world. No one will invest in you more than you. So wake up and try to create your own challenge. Buying a brand-new smartphone is not always better than paying for mobile testing training. You need to do both.
- Be sure that you are checking your test objectives and their alignment with the business requirements before you start any test activity.

- Try to be dynamic and prepare yourself for time pressures, frequent changes, and budget limitations.
- Adapt your test methodology to your SDLC approach. Testing is a purely context-driven activity, so it makes a significant difference if you are doing agile, waterfall, spiral, or v-model development.
- A defect-free app does not mean that you are successful. The main success factor of any mobile app is the user satisfaction it creates.
- And finally, there is the illusion of 100 percent testing. In no circumstance ever say, “we have tested the app 100 percent.” This is not possible! Even though you have tested a mobile app for years, no one can be sure that the product is tested 100 percent. During the training that I deliver, people sometimes object to this principle and say, “We can be sure that all the requirements are covered by our test cases. Current tools and technologies allow us to achieve 100 percent requirement coverage.” My answer is simple; 100 percent requirement coverage does not mean that the product is tested 100 percent. This is because the requirements are still being written by humans, and no one can be 100 percent sure that these artifacts are covering all the functional and nonfunctional aspects of any mobile product. So once you say, “All the requirements are covered,” it does not mean that 100 percent of the product is tested. At least wise people will distinguish the difference.

I think that the foregoing principles and strategies are important in mobile testing. Whether you agree 100 percent or not, a majority of them can help you in establishing your ideas, organizations, and processes, and they can assist you in making better decisions.

As Victor Hugo once said, “Change your opinions, keep to your principles; change your leaves, keep intact your roots.”

TIP 28
MOBILE USABILITY TESTING IS NOT ROCKET
SCIENCE!
(CONTRIBUTED BY VEHBI KOKSAL)

*Software testers do not make software; they only
make them better.*

—Anonymous

MOBILE USABILITY TESTING IS NOT ROCKET SCIENCE!

For some reason, people not involved with the field of UX (usability) have the idea that usability testing in general is as difficult as rocket science—especially for mobile and the constraints it comes with.

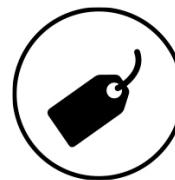
We hear from people that the main reasons for not conducting usability tests are that they don't have the know-how, "enough" time, and, most commonly, the budget. Of course, some approaches can be very difficult, take a lot of time, and be very expensive. But they don't have to be. In this chapter, we will discuss some of the approaches we use for mobile usability testing to keep all these factors as low as possible. We'll also address how mobile usability testing differs from desktop usability testing.



ROCKET SCIENCE



TIME-CONSUMING



HIGH PRICE TAG

Let's start easy

Being a usability or software tester doesn't have to be a requirement to conduct usability tests on mobile devices. If you are a designer, project manager, marketer, or even an enthusiast, you can start doing a simple mobile usability test on your own. Just launch the app (or website) and start using it. However, the biggest mistake we make when doing this is that we do not take into consideration that mobile devices are being used in different settings. It surprises me that many people have the tendency to conduct mobile usability tests at locations that do not represent the actual environment where mobile devices are used. Instead of using (read, testing) your mobile app at the office or in a usability testing lab, start using your app on the go. Use it when you are on the train, when you are cooking, when you are waiting at the dentist, or when you are having lunch. This will be a much more realistic setting than testing in your Zen-inspired office with reliable Internet connectivity.

You can expand this approach by including your relatives, friends, and coworkers to this simple mobile usability test. Just like you did,

let them use the app, and observe them while they interact with it. You will learn a lot just by looking at the screen and listening to what they have to say about the app afterward.

This simple, low-cost, and effortless approach for a mobile usability test might not provide you detailed metrics or a good basis for a substantiated conclusion that can affect big design decisions, but it does give you insight about how your app responds in different settings and how people use and think about your app.

Kick it up a notch

Testing on your own or with the closest people around you might not be the mobile usability testing approach you are looking for. You might want to include people who are actually representing your target audience, and at the same time keep the “quick and dirty” usability testing approach we discussed in the previous paragraphs. If this is the case, hallway testing is most likely the approach you would want to use.

For one of our clients, we designed a mobile app from the ground up. During the early stages of the design phase, we had to make a decision regarding the navigation of the app that would influence the entire design and structure. Before making such a big decision, we decided to test the navigation we had come up with first on usability. In order to do this, we created a low-fidelity prototype in just a few hours using an online prototyping tool, and we installed this on our mobile devices.

With a team of three, we went to one of the busiest shopping streets of the city—not to look for new clothes, but because our target audience was there: females between the ages of eighteen and twenty-five. Our team of three consisted of a greeter, a facilitator, and an observer. The role of the greeter was to spot people who would fit our target audience and persuade them to participate in our mobile usability test. Our greeter explained to the participants that the usability test would take only ten to fifteen minutes and that in return, the participants would be rewarded with a multistore gift card worth ten euros.

Once a passerby had decided to participate in our mobile usability

test, the facilitator stepped in. His role was to explain our goal and expectations from the usability test, and to introduce the mobile app without giving away too much detail because of our nondisclosure agreement with the client. Next, the facilitator handed over the mobile device and gave the participant the tasks he or she had to fulfill using the mobile app. While the participant was trying to complete these tasks, our observer took notes about the participant's progress. Since the goal of the usability test was to find out if our users understood the navigation of the app, we only kept track of whether the participant passed or failed a given task. To learn more about the participant's motives and behaviors, we conducted a short postsession interview.

We repeated this process with ten other participants. As we noticed a pattern when comparing the results of all the usability tests, we decided that we no longer needed to include more participants. With a limited amount of time and budget, this approach of mobile usability testing, or hallway testing, gave us insight about how users responded to the navigation we came up with for our mobile app. Even though this approach does not provide detailed metrics, it is a "quick and dirty" way to help you make substantiated decisions in the early stages of the design phase.



EQUIPMENT



SOFTWARE



ENVIRONMENT

Get more detailed

In many cases we find it more useful to have a more comprehensive approach when it comes to mobile usability testing. Usually, the mobile usability tests we conduct for our clients are as detailed as classic desktop usability tests. Even though this approach requires more know-how, has a larger time frame, and has a higher price tag, it does give us a more detailed and valuable view of how efficient, effective, and satisfying the product is from the user's point of view.

For these kinds of usability tests, we spend more time when it comes to preparation compared to the other approaches. During the test

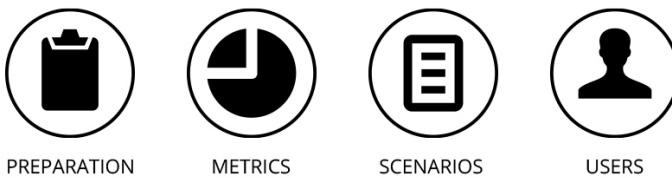
sessions, we capture every single detail using recording equipment and software. We capture screen activity, the user's facial expressions and voice, the input of the user (e.g., mouse clicks) while performing a task, etc. In order to run the test sessions without any discommodity, we test the hardware and software beforehand at the test location. We even perform connectivity tests if the usability test requires the use of an Internet connection. Running several pilot sessions is not an unnecessary luxury, so do not forget to include these into your preparation as well.

Another aspect that makes these kinds of usability tests more detailed are the metrics we are focusing on. We spend more time and put more thought into what kind of data we need to collect from the usability tests. Depending on the goal of the usability test, you can choose to collect quantitative data, qualitative data, or a combination of both. Without going into too much detail, quantitative data is the type of data that can be measured, like length, height, and volume. The other, qualitative data, cannot be measured but can be observed, like smells, tastes, and appearance. In the context of a usability test, quantitative data can be the measurement of how much time a user spends to complete a certain task, and qualitative data can be the observation of how a user reacts to a certain design.

Depending on the scope of the usability test, the scenarios the users need to walk though can also be quite long and more specific than the other usability testing approaches we have discussed in the early paragraphs of this chapter. For more complex scenarios, you may even need to provide supplementary information, like login credentials, dummy credit card details, etc. A good way to check whether the scenarios are being well interpreted by the participants is to run pilot sessions as mentioned previously.

Last but not least, finding users for these kinds of usability tests is much more difficult and time-consuming. It is not easy to persuade people to participate in a usability study that can take one to two hours of their time, especially when the target audience (or persona profile) encompasses those with a busy lifestyles. That is why we recommend that our clients attract participants with incentives in the form of cash, gift cards, coupons, etc. When the amount or type of incentive is right, which can differ per person, it usually is easier to attract participants. But keep in mind that this will increase the cost

of your usability test.



Mobile usability testing constraints

Even though we use the same approach for mobile and desktop usability testing, we unfortunately have to face two constraints when conducting usability tests for mobile. The first and foremost constraint concerns recording equipment and software. As mentioned before, during classic usability tests we use recording equipment to, for example, capture screen activity. For desktop usability testing we can do this by mounting an external video camera facing the screen, or by installing specific screen-recording software on the desktop. These are two techniques we can't use during mobile usability testing. Using a mounted video camera restricts the participant from moving the mobile device around freely while testing the mobile app. And installing and running recording software on a small device like a smartphone can slow down the performance of the mobile device, which may affect the progress of the usability test and lead to unrealistic test results.

To bypass this constraint, a mobile usability testing sled can be used. This piece of hardware enables us to capture the screen activity and the interaction between the user and the device by using a mounted webcam, all while providing the user with mobility. Unfortunately, since no additional recording software is being used, some features that might come in handy, like recording the input (e.g., screen taps), are not possible.

The second constraint we have to deal with when conducting usability tests for mobile is the testing environment. As explained earlier, unlike desktops, we use our mobile devices on the go. Therefore, it is advisable to conduct mobile usability tests in the environment where the app will be used. Depending on the context and target audience, you can conduct a mobile usability test in places

such as a train, the university, a shopping mall, etc. However, since we are on the go, keep in mind that connectivity, lighting, and distraction play an important role during the usability test, and that a sudden change to any of these factors can influence the user's behavior. So when determining a test location, conduct pilot sessions first to prepare yourself for what you can expect. In some cases, conducting an outdoor mobile usability test can be too risky. If that is the case, take a step back and try to find alternatives. Even though conducting mobile usability tests at the office or in a usability lab is not ideal, it is always better than not testing at all.

KEYTORC PEOPLE ANSWER THE QUESTION “***WHAT IS MOBILE TESTING?***”

“Done for using my phone seamlessly.”

Sera (HR manager)

“Checking if an application works or not on mobile devices.”

Merve (marketing and social media expert)

“What kind of testing do you mean? Performance testing or usability testing?”

Vehbi (user experience expert)

“Testing of cell phones.”

Esra (office secretary)

“A test executed on mobile devices.”

Berk (consultant)

“Testing reminds me of a process of controlling devices to check if they are working in a proper way or not.”

Emre (part-time engineer)

“The effort of making mobile applications more efficient and smoother.”

Bora (intern)

“I have no idea, sweetheart.”

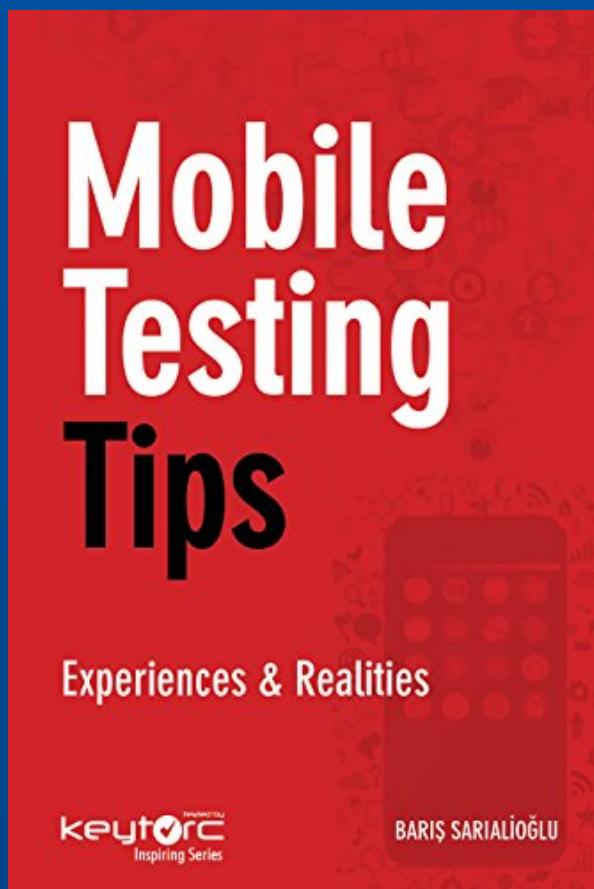
Hafize (cleaning lady)

“Does it have to do with a phone?”

Avni (the postman)

Note: Be aware that, these people are not randomly selected.
They are the heroes behind our successful company, Keytorc.

Purchase the full book



amazon kindle

BUY NOW

amazon

BUY NOW

Enjoyed this eBook and want to read more?

Check out our extensive eBook library on huddle.eurostarsoftwaretesting.com



Join us online at the links below

