Effectiveness of Multi-device Testing Mobile Applications

Sergiy Vilkomir, Katherine Marszalkowski, Chauncey Perry, Swetha Mahendrakar
Department of Computer Science
East Carolina University
Greenville, NC 27858 USA
vilkomirs@ecu.edu

Abstract—The paper evaluates methods of selecting mobile devices (i.e., smartphones and tablets) to test mobile software applications. Due to the number of such devices on the market and the variations in their characteristics, it is hard to guarantee that an application will work properly on all devices. Software faults found only in certain devices are known as device-specific faults. The goal of this research was to determine how many devices must be tested and which methods of device selection are best for revealing these device-specific faults. To collect the experimental data, 15 Android applications were tested on 30 mobile devices using real devices and remote testing services such as TestObject, pCloudy, and others. In total, 24 device-specific faults were detected. Using our data, we analyzed the different methods of selection based on the characteristics of each device. Our research showed that a random selection of 13 devices achieved 100% effectiveness. However, coverage of device characteristics in the selection process enables the achievement of an acceptable 90% level of effectiveness with sets of only five devices. The most successful approach was the coverage of different types of Android operating systems. Our results include recommendations that will help increase the effectiveness and decrease the costs of mobile testing.

Index Terms—Mobile testing, device-specific faults, Android, device selection.

I. INTRODUCTION

According to a recent survey, 91% of American adults own a cell phone and 50% of them use these devices to download and run mobile applications [1]. Mobile application testing is therefore important to ensure the quality of such applications (apps). One of the reasons that makes mobile testing a critical and challenging task is the so-called “fragmentation problem,” in other words, the large number of distinct mobile devices (i.e., smartphones and tablets) in use at the same time. This problem is especially important for Android devices [2]. A 2014 report by OpenSignal claims that 18,796 distinct Android devices were seen in that year [3]. Due to the huge number of devices on the market and the variations in their characteristics, it is hard to guarantee that an application will work properly on all devices without device-specific faults (i.e., faults found only on certain devices).

This problem is well understood by practitioners and researchers. Muccini et al. [4] included “new and rapidly evolving mobile operating systems” and “diversity of phones and phone makers” in the list of the peculiarities of mobile apps that have a direct impact on software testing. Many practical solutions to mitigate this problem were suggested, starting with Google Play Services that allow the customer to “use the newest APIs for popular Google services without worrying about device support” [5]. Commercial tools and research prototypes developed to support different aspects of multi-device mobile testing include Ghostlab [6], Adobe Edge Inspect [7], Silk Mobile [8], AppACTS [9], CTOMS [10, 11], and others.

Before applying any methods and tools for multi-device testing, mobile devices should be selected for testing. The general approach to this selection, which is quite obvious and commonly accepted, suggests maximizing diversity coverage and minimizing the number of devices (see, for example, [4]). However, practical application of this approach requires answering the following simple questions:

- How many devices to select?
- Exactly which devices to select?
- What is the algorithm for the selection?
- Which device characteristics should be covered in the first place?
- Which approach to this selection is the most effective?

Surprisingly, these simple questions are not adequately considered in the research literature. Of course, during practical mobile testing, each company or developer makes these decisions, but such decisions are based on their own understanding and available resources, rather than on a solid theoretical basis. Several examples of the practices of different companies are described in [12], and the number of devices used for testing varies significantly from 12 (Red Robot Labs), to 40 (Pocket Gems), to between 30 and 50 (Storm8), and up to 400 (Animoca).

Considering that research investigation of these questions would be very useful and promising, we previously took the initial step toward filling in this gap [13]. The current paper continues in this direction by using experimental testing to evaluate the effectiveness of different approaches to the selection of mobile devices for testing.
This paper is structured as follows: Section 2 presents the research questions and main methods of investigation. All aspects of our experimental testing (apps under testing, devices used, and detected faults) are described in Section 3. Sections 4 and 5 provide numerical results and an evaluation of the effectiveness of the different methods by which devices are selected for testing. Section 4 considers a random selection of devices, and Section 5 considers selection with coverage of different values for different device characteristics. We analyzed the coverage of different Android operating systems (OS), screen resolutions, screen sizes, manufacturers, and sizes of random-access memory (RAM).

II. RESEARCH QUESTIONS AND METHODS OF INVESTIGATION

Two main research questions are investigated in this paper:

• RQ1: How effective is random selection of mobile devices for testing?
• RQ2: How effective is coverage of various device characteristics in the selection of mobile devices for testing?

We were interested in numerical evaluations of the different aspects of device selection. In particular, we investigated:

• RQ1-1: How much does the effectiveness of testing increase when the number of devices increases?
• RQ1-2: How many devices should be selected to reach an effectiveness level close to 100%?
• RQ2-1: Which characteristics of mobile devices should be covered during device selection to reach the maximum effectiveness of testing?
• RQ2-2: Is a small number of devices (four or five) sufficient for effective testing?
• RQ2-3: What is the effectiveness of small sets of devices (four or five) for the different selection approaches?
• RQ2-4: What is the difference between the effectiveness of random and coverage selection with the same number of devices?
• RQ2-5: How effective is “each-choice” selection (i.e., the coverage of all device characteristics at the same time)?

We used an experimental approach to evaluate the effectiveness of device selection that included the following steps:

Step 1. A set of mobile applications was formed for testing. We did some preliminary testing and analyses, and included an application into this set only if the app has at least one device-specific fault. In total, 15 apps were chosen.

Step 2. A set of devices was formed for testing. Our aim was to create a set large enough to permit the assumption that most of the device-specific faults would be detected after testing. Also, the devices in this set would represent a wide variety of device characteristics (e.g., type of OS, manufacturer, screen resolution, etc.). In total, 30 devices were chosen.

Step 3. All apps from step 1 were tested on all devices from step 2 (exhaustive testing). All detected device-specific faults were logged, and this information was used for the next steps in the evaluation of effectiveness. In total, 24 different device-specific faults were detected.

Step 4. To answer RQ1, we randomly formed subsets of devices of different sizes from 1 to 13 and evaluated how many faults could be detected by each subset. Data from step 3 were used and no additional testing was required. The effectiveness of each subset of each size was evaluated based on the number of detected faults, taking all 24 faults as 100%. For example, if testing on devices from some subset detected 12 faults, its effectiveness was considered as 50%. To increase the trustworthiness of the results, 15 different subsets were formed for each size of subset and the average effectiveness was calculated for each size.

Step 5. To answer RQ2, subsets of four and five devices were formed for each device characteristic. Each subset covered different values of the corresponding characteristic. Similar to step 4, 15 different subsets were formed for each size and characteristic. Effectiveness was evaluated for each subset of devices and for each characteristic on average. As in step 4, no additional testing was required.

The scope of our experimental investigation is summarized in Table I and the detailed description is given in Section 3. As with any experimental investigation, increasing the number of experiments (number of apps and devices in our case) would be useful and desirable, but this requires more time and other resources, and is more costly. We consider this investigation to be a pilot project that can be continued and extended in the future.

<table>
<thead>
<tr>
<th>TABLE I. THE SCOPE OF THE EXPERIMENTAL INVESTIGATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of</td>
</tr>
<tr>
<td>Applications</td>
</tr>
<tr>
<td>Devices</td>
</tr>
<tr>
<td>Different detected faults</td>
</tr>
<tr>
<td>Faults that occurred</td>
</tr>
<tr>
<td>OS coverage sets (4 and 5 devices)</td>
</tr>
<tr>
<td>Manufacturer coverage sets (4 and 5 devices)</td>
</tr>
<tr>
<td>Screen resolution coverage sets (4 and 5 devices)</td>
</tr>
<tr>
<td>Screen size coverage sets (4 and 5 devices)</td>
</tr>
<tr>
<td>RAM memory coverage sets (4 and 5 devices)</td>
</tr>
<tr>
<td>Each-choice coverage sets (5 devices)</td>
</tr>
<tr>
<td>Random coverage sets (1 to 13 devices)</td>
</tr>
</tbody>
</table>

III. EXPERIMENTAL TESTING

A. Mobile Applications

We selected 15 Android applications for testing. They were divided into several groups based on their functionality. The main three groups are (1) car/hotel/flight rental and booking, (2) health/fitness, and (3) maps/locations. Auto Europe, Bookit, and Hotwire fall into the booking and rental group of apps. These apps search for various hotels, vehicles, flights, or combinations of these services. The user must provide financial and other personal information in order to complete the process. Calorie Counter, Fitness Calculator, and Color Blindness belong to the health/fitness group. They provide users with some health information. Finally, FMCA, Mobile RVing, NoWait, and Sun Shield require the use of maps and
locations. A specific location is often used as a test input for each of these applications (we used “Greenville, NC”).

B. Devices

Each of 15 apps was tested on 30 different Android devices. The set of devices includes some modern popular models such as Samsung Galaxy S5 and some old models that are still in use. To make the set of devices representative enough, we included smartphones and tablets from different manufacturers (e.g., Samsung, HTC, LG, others), with different OS (from Android 2.1.0 to Android 4.4.2), different screen resolutions (from Low to Extra High), different screen sizes (from 2.8 inch to 10.1 inch), and different RAM memory (from 278 MB to 2.0 GB).

C. Testing

A small part of testing during our investigation was done on several real (physical) devices that were available to our research team. However, for the main part of the testing, we used special remote testing services. It is important to mention that all of these services provided access to real devices (and not to emulators). While the access was via a web browser, the results are the same as using the device with your own hands. Screenshots and screen videos are available and can be used for analysis and fault detection. The following mobile testing services were used during our investigation:
- pCloudy [14]
- TestObject [15]
- DeviceAnywhere (Keynote Mobile Testing) [16]
- Perfecto Mobile [17]

These services have similar purposes and main functionality, but they vary significantly in cost, available devices, additional functionalities, usability, etc.

D. Device-specific faults

During the investigation, each of the 15 apps was run on each of the 30 devices, so 450 test runs of applications were conducted. All detected faults were recorded. We were concerned only with device-specific faults, i.e., the faults of the same apps that occur on certain devices but not on all devices. Several of these faults occurred on each device. Also, each fault occurred on several devices (from 4 out of 30 devices for the fault of the Sun Shield app to 24 out of 30 devices for the fault of the Fox2News app).

Altogether, 24 different device-specific faults were detected. These faults occurred 317 times during testing. There were different types of faults among the detected device-specific faults. For example, the application crashed as soon as it started for some devices or the page loaded on forever. Some faults took the form of incorrect or incomplete information on the screen.

Investigation of the deep-rooted causes of device-specific faults is an interesting and important (though not simple) task. However, this was not an aim of our investigation. Instead, we used statistical information about the detected faults to evaluate the effectiveness of different approaches to the selection of mobile devices for testing. The results are presented in Section 4 for random selection and Section 5 for coverage selection.

IV. EFFECTIVENESS OF RANDOM SELECTION

The average effectiveness of the random sets of devices of different sizes is given in Fig. 1. The plot shows that 90% effectiveness is reached for the sets of seven devices and 100% effectiveness is reached for the sets of 13 devices.

![Fig. 1. Average effectiveness of random sets of devices of different sizes.](image)

The values for effectiveness are very stable among the sets of devices of the same size. Thus, testing each of the 15 sets of 13 devices detected all 24 faults, so the level of 100% effectiveness is the same for all random sets of 13 devices and does not depend on the selected devices.

V. EFFECTIVENESS OF COVERAGE SELECTION

In this section, we consider effectiveness of testing when the devices are selected not randomly but with the aim to cover different values of different device characteristics. We investigated the effectiveness of sets of four and five devices that are the most reasonable sizes for real practical testing. Because the number of different values of some characteristics is more than five, this requires more than five devices for testing, and the technique of restricting the number of options for each device characteristics is used, as we suggested previously in [13].

For this purpose, if the number of options is more than four, then we consider the three main or most popular options and combine all other options into one artificial option, “other.” This approach allows us to cover all selected options during testing with four or five devices.

We formed 15 sets of four devices and 15 sets of five devices for each of these five characteristics to cover all values. The example of such a set for coverage of the OS is given in Table 2.

<table>
<thead>
<tr>
<th>Num</th>
<th>Device</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motorola Moto G</td>
<td>Kit-Kat</td>
</tr>
<tr>
<td>2</td>
<td>Sony Xperia J</td>
<td>Jelly Bean</td>
</tr>
<tr>
<td>3</td>
<td>Samsung Galaxy Y Plus</td>
<td>Ice Cream Sandwich</td>
</tr>
<tr>
<td>4</td>
<td>Samsung Galaxy Pocket</td>
<td>Other (Ginger Bread)</td>
</tr>
<tr>
<td>5</td>
<td>LG Optimus G Pro</td>
<td>Kit-Kat</td>
</tr>
</tbody>
</table>

We also formed 11 sets of five devices according to each-choice coverage, which is coverage of all characteristics at the same time. The results are presented in Fig. 2.
The most successful approach is the coverage of different types of Android operating systems, which achieves practically 90% effectiveness on average for sets of five devices. Table 3 compares the average results for each characteristic with a random selection and each-choice coverage.

TABLE III. EFFECTIVENESS OF THE DIFFERENT TYPES OF COVERAGE

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Effectiveness %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 devices</td>
</tr>
<tr>
<td>OS</td>
<td>85.6</td>
</tr>
<tr>
<td>Resolutions</td>
<td>77.5</td>
</tr>
<tr>
<td>Manufacturers</td>
<td>78.3</td>
</tr>
<tr>
<td>Screen Sizes</td>
<td>81.7</td>
</tr>
<tr>
<td>RAM</td>
<td>81.1</td>
</tr>
<tr>
<td>Each-choice</td>
<td>-</td>
</tr>
<tr>
<td>Random</td>
<td>81.4</td>
</tr>
</tbody>
</table>

Table 3 shows that the results for the coverage of screen resolution, screen size, and manufacturer do not exceed the results for random selection. For the first two characteristics, these results are unexpected to a certain extent because some faults become apparent when the same information is displayed differently on different device screens, and it could be assumed that these faults depend on screen size or resolution.

Coverage of different types of OS shows that the results are significantly better than random selection and can be recommended as a reasonable and practical approach. It is interesting to compare OS coverage with each-choice coverage. It could be assumed that, because each-choice coverage includes OS coverage and also covers other characteristics, the effectiveness of each-choice coverage would be higher. However, our results demonstrate the opposite, showing that OS coverage is slightly more effective than each-choice coverage (89.2% vs. 87.5%), though the results are close to each other. This could be considered one more confirmation that covering OS is more important than covering other device characteristics.

VI. CONCLUSIONS

This experimental investigation evaluated the effectiveness of mobile testing on different devices. The goal was to determine how many devices must be tested and which methods of device selection are best for revealing device-specific faults. Fifteen Android applications were tested on 30 mobile devices each and 24 device-specific faults were detected. Different methods of device selection were analyzed. Random selection of five devices provided 85% effectiveness and achieved 100% effectiveness for 13 devices. Selection based on coverage of different types of OS shows better results (near 90% for five devices). However, coverage of other device characteristics does not demonstrate any benefits compared with random selection. OS coverage looks promising as a reasonable, practical approach that gives a high level of effectiveness of fault detection with testing on a relatively small number of mobile devices.

ACKNOWLEDGMENT

This work was performed under the following financial assistance award 70NANB14H264 from U.S. Department of Commerce, National Institute of Standards and Technology. The research is also partially supported by the National Science Foundation REU Site Award 1262933.

REFERENCES