Usability Evaluation Framework for Mobile Apps using Code Analysis

by

Neeraj Mathur, Sai Anirudh Karre, Y.Raghu Babu Reddy

in

*The Evaluation and Assessment in Software Engineering (EASE) conference*

Christchurch, New Zealand,

Report No: IIIT/TR/2018/-1

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2018
Usability Evaluation Framework for Mobile Apps using Code Analysis

Neeraj Mathur, Sai Anirudh Karre, Y. Raghu Reddy
Software Engineering Research Center
IIIT Hyderabad, Gachibowli, Hyderabad, India
neeraj.mathur, saianirudh.karri(@research.iiit.ac.in), raghu.reddy@iiit.ac.in

ABSTRACT
The increasing usage of smart-phones has resulted in mobile applications replacing or supplementing traditional web-based applications. Given the limitations of the form factor in smartphones, usability can be considered as one of the important attributes that determine the success of a mobile application. The measures available for assessing the usability of mobile applications tend to focus more on human aspects and less on the functional aspects of usability. As part of this paper, we propose a usability evaluation framework to identify functional usability issues specific to mobile applications. This framework uses usability guidelines and code analysis to improve the usability of a mobile application. As a proof of concept, we have built an end-to-end system using the framework to validate and verify usability issues in Android mobile applications. We also generate code recommendations to implement failed usability guidelines.

KEYWORDS
Mobile Usability, Usability Guidelines, Code Analysis, Usability Evaluation, Automation, Mobile Apps

1 INTRODUCTION
A report published by a leading survey agency Nielsen titled "An Era of Growth: The Cross-Platform Report" ¹, found that the end users spend 90% of their time on mobile applications when compared to web-based applications on their mobile devices. Based on this earliest study, the exponential growth of smartphone usage coupled with decreasing cost of broadband data. This has influenced organizations to offer their products and services via mobile applications (apps).

1 https://goo.gl/1J5gEv

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EASE '18, June 28–29, 2018, Christchurch, New Zealand
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6403-4/18/06... $15.00
https://doi.org/10.1145/3210459.3210480

Given this rise in the mobile app market, there has been a significant push by various organizations to withstand their respective market competition and also to add value to customers by making their applications more usable (ease of use). Sheng et al.[17], in their study, observed that to compete in the market, it is now obligatory for mobile app firms to follow the usability guidelines and adapt them as part of their development practice. In reality, mobile apps have minimal generalized usability standards when compared with sophisticated usability standards of web-based applications.

Usability guidelines for mobile apps detail the ease of use, efficiency, etc. It is largely a human-centric activity and perception based. However, there are several guidelines that are functional in nature and can be directly correlated with specific parts of implementation at the code level. For example, an app developer might choose to implement the following guidelines at the code level: "Password field should reveal the password in clear text". This guideline can impact the assessment of usability from a user perspective. There are several such guidelines that can be automated at the code level but are not specified as requirements by the majority of the stakeholders. Such functional aspects of usability tend to be downplayed in focus groups and other human-centric perception based studies. Mechanisms that automatically evaluate existing mobile apps for such guidelines and/or recommend code-snippets that lead to enhanced usability levels of the mobile apps can certainly add value to the organizations.

In this paper, we present a "Usability Evaluation Framework" for mobile apps based on prevailing usability guidelines. The framework can be used to assess various mobile applications with respect to their specified guidelines. It is flexible and extensible in the following ways: (1) New guidelines can be added/deleted/modified (2) Validation cases to existing guidelines can be added/deleted/modified (3) Code-snippets for existing or new guidelines can be added/deleted/modified. As a result, an organization can create their own instance of the framework and perform a usability evaluation against their mobile app based on their business need.

The rest of the paper is organized as follows: Section 2 provides background on usability guideline and outlines the need for a usability evaluation framework, Section 3 describes the related work, Section 4 details the proposed usability evaluation framework, and the proof of concept based out of our framework, Section 5 details case study, results and industrial validation. Section 6 we describe the scope of our future work and limitations.
2 USABILITY GUIDELINES

ISO defines usability as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use” [13]. As per ISO - Learnability, Efficiency, Memorability, Errors and Satisfaction are considered to be the traditional usability attributes.

Most of the mobile usability guidelines [20] are generic and are non-domain specific by definition. With the increase in the number of unique usability guidelines, mobile software practitioners have become unclear on implementing these guidelines through code [2]. It is also difficult to validate and map these guidelines with usability attributes. Classifying the available guidelines based on usability attributes can help mobile software testers to validate the usability of their mobile apps. We have rigorously reviewed the available usability guidelines from Nielsen et al. [20] and Google [10]. We found that these guidelines can be broadly classified into two categories Functional and Non-Functional.

**Functional:** It is defined based on a behavior or a feature associated with usability aspect of an app, which can be directly mapped to code level implementation. Let us consider a Functional guideline [20] “Password field should reveal password in clear text”. In Android, this guideline can be achieved by decorating “EditText” of type “textPassword” with “TextInputLayout” as shown in listing 1. Thus by checking presence of “TextInputLayout” in code, we can validate the conformance of this guideline.

**Listing 1: Reveal or Hide Password**

```xml
<android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout_password">
  <EditText
      android:id="@+id/txtPassword"
      android:hint="Password"/>
</android.support.design.widget.TextInputLayout>
```

**Non-Functional:** It is defined as a statement not related to a functionality of a given app but it is indirectly associated with usability aspect of an app. It is not possible to map these guidelines to code level implementation. Let us consider a Non Functional guideline [20] “Label button in a descriptive way”. Checking conformance to such a guideline is difficult via a tool and requires manual validation.

2.1 Usability Testing in Mobile Apps

According to Chi et al. [6] there can be two ways to set-up an automatic usability evaluation. They are:

- Make use of conformance to standards or guidelines
- Try to predict the usage pattern of a system

We carried out a review on available automation tools for evaluating usability issues on Mobile apps. Majority of these tools are best fit for either black/gray box testing or manual testing on a post-release version of a mobile app. These tools offer User Feedback, Session recording and Event watching and analyze usage behaviour on a mobile app. These existing tools may not help developers solve usability challenges during design and development phase. Justine et al. [15] has developed a framework to conduct usability evaluation on websites based on usability standards. However, this framework was limited to websites only and was based on HTML code analysis which is beyond the scope for Mobile apps.

3 RELATED WORK

Usability Evaluation of mobile apps has become a wide research area for researchers in the past few years.

**Usability Models:** Harrison et al. [11] introduced PACMAD (People At the Center of Mobile Application Development), to evaluate the usability of the mobile app. Sanjeev et al. [2] have performed a systematic literature review on usability challenges in mobile web and identified the list of practices and usability models. Chen et al. [8] have investigated the differences between the usability tests on mobile phones in the laboratory and real-life situations. Nielsen UI research group has formulated an exhaustive list of heuristic evaluation methods [19]. Kutar et al. [12] have proposed a Goal Question Metric (GQM) based on Usability Guidelines for Mobile applications. Rahmat et al. [21] proposed a framework with an integrated approach to usability evaluation for a smart-phone app in view of abstraction levels of usability criteria and user interface elements of an app. All the above methods do not provide recommendations to developers to address usability challenges during development phase.

**Reviews and Field Studies:** Kim et al. [7] performed a meta-analytical review on existing empirical mobile usability studies and listed out contextual factors, the core and peripheral usability dimensions and key findings for future of mobile usability research. Singh et al. [4] came up with twenty-five dimensions based on a review model and proposed these dimensions as key attributes for designing a usable mobile app. Kortam et al. [14] used System Usability Scale (SUS) to collect data on usability applications on mobile platforms based on popularity and captured SUS rating to benchmark the design decisions. Ayob et al. [3] came up with UI guidelines identified based on a human evaluation of mobile apps using various interface design rules.

**Experimental Analysis:** Abran et al. [18] conducted an expert based usability study to understand the usability challenges in iOS mobile apps. Stringer et al. [5] proposed a mobile framework for two types of cognitive tests - Trail Making and Reaction Time and conducts an investigation of its performance. Lopez et al. [9] study have allowed identifying different interaction patterns and contextual factors not likely to be simulated in laboratory environment. Lalit et al. [16] performed a survey-based analysis to understand the usability challenges in mobile apps and proposed a usability index to usability evaluation. Kramer et al. [23] proposed a design-oriented usability evaluation questionnaire to validate the human interaction with business mobile apps and published their significant observations.

Overall, there are detailed usability studies published in the literature to address the usability challenges. However, practical implementation or case studies on addressing usability challenges through code analysis are negligible. Following are the unique contributions of our research:

- Create a usability evaluation framework for designers and developers to address usability issues through code analysis using usability guidelines during development phase.
Usability Evaluation Framework for Mobile Apps using Code Analysis

- Create a prototype tool based on the framework for software practitioners to conduct usability evaluation on their mobile apps during development and pre-release phase.

4 USABILITY EVALUATION FRAMEWORK

Figure 1 provides an overview of our proposed usability evaluation framework. This framework can be extended to including all types of mobile apps (Android SDK, iOS). A developer who builds mobile apps or a usability tester who validates a mobile app can make use of our instantiated framework. The user should possess a mobile app in an Android application package (APK) format for Android Apps or iPhone application archive (IPA) format for iOS Apps in Debug mode as it generates all the dependent files related to a mobile app. Below are the four components involved:

![Usability Evaluation Framework Diagram]

**Decompiler/Source Code:** An executable Mobile app file or the source code of the Mobile App should be supplied for code analysis. In case of an executable mobile app file (e.g., APK), the framework will execute the decompilation process over the app file to generate the source code. For decompiling (APK) file we have used open source tools like dex2jar, apktool and jd-core. iOS app files are encrypted and not easy to decompile, therefore we need actual source code for iOS platform.

**Validation Test Case Generator:** A validation test case is a test case built based on a usability guideline. It consists of steps in XML format to detect the presence of a particular usability guideline, as shown in Listing 2. Here, the Usability guidelines can be considered as conformance rules. Validation test case generator facilitates the creation of test cases through a tool as shown in Figure 3, where a user can define what needs to be searched in the source code for analyzing presence of a guideline. We have built controls to search a specific type of artifact of a source code detailed architecture of validation test case generator is shown in Figure 2. We have built custom controls like Read Layout

![Validation Test Case Generator - Architecture Diagram]

**Validation Execution Engine:** The given validation case is loaded by execution engine and is invoked during execution. Here, all the attributes and constraints defined in a validation case are passed in a sequence for chosen guideline execution. Every validation case execution returns an individual result. These details are collated and are published in a database called MetricDB. These results are delivered to the external reporting system for visualization.

**Recommendation Engine:** It provides code snippet suggestions for failed guidelines. The code identified for a usability guideline will be added to a code snippet (live templates) central library by developer/tester along with guideline descriptions. Recommendation engine uses these descriptions to match (using Lucene search engine) the related code snippets for code suggestions during the evaluation process. They are stored in Code Snippet DB.

4.1 Incorporating guideline into validation case

The validation cases designed to evaluate the usability guidelines can be stored and re-used for multiple mobile apps. At any given time, a guideline can be added, removed, or modified for the overall evaluation. It is not required to rebuild the validation case for a given guideline every time we perform the evaluation. This essentially implies that a non-programmer can add or remove the guidelines to perform the overall validation by just choosing the guideline from guideline repository. This also implies there can be multiple

---

3https://lucene.apache.org/core/
instances of the same guideline and corresponding validation cases based on the various types of mobile apps. Below steps provide us an insight on how a guideline is converted in to a validation test case:

- **Step 1**: Select a mobile usability guideline specified in plain text form. For example “Password field should reveal password in clear text” [10].
- **Step 2**: Identify code related to usability guideline i.e. for example, if an EditText box of type password is wrapped inside the TextTemplate widget - it does enable Show Hide password button (Eye icon) as shown in listing-1. This code can be found in the Android SDK.
- **Step 3**: Identify steps which need to be checked to validate conformance of this guideline.
  1. Count total number of password field which are present in source code
  2. Count total password fields which are wrapped inside the TextInputLayout widget.
  3. Check if there is a mismatch in counts
- **Step 4**: Develop validation case using validation case generator Fig.3. Each validation case is saved in XML format as shown in listing-2, which validates the above identified steps.

### Listing 2: Validation Case

```
<ValidationCase Name="Check Show/Hide Passwords">
<Variables>
  <Variable Name="PwBoxListWithShowHide" Type=ListOfString/>
  <Variable Name="PwBoxListWithoutShowHide" Type=ListOfStrings/>
</Variables>
<ReadLayoutAttribute AttributeToRead="android:id"
  Result="[PwBoxListWithShowHide]
  XPath="/android.widget.TextInputLayout/
  EditText[android:inputType=typeTextPassword]" />
<ReadLayoutAttribute AttributeToRead="android:id"
  Result="[PwBoxListWithoutShowHide]
  XPath="/EditText[android:inputType=typeTextPassword]" />
<If Condition="[PwBoxListWithoutShowHide.Count > PwBoxListWithShowHide.Count]">
  <Response>
    <Passed>True</Passed>
    <Text>Validation Case Passed</Text>
  </Response>
</If.Then>
<If.Else>
  <Response>
    <Passed>False</Passed>
    <Text>Validation Case Failed</Text>
  </Response>
</If.Else>  </If>
</ValidationCase>
```

The validation case XML file is used to perform usability evaluation. The validation cases, once built, can be re-used based on the app chosen for evaluation. For example, in the above case, the validation case XML file generated can be used for every Android mobile app which has a password text field in any given form.

### Figure 3: Define Validation Case

#### 4.2 Proof of concept
We have build a tool [1] for Android mobile apps, based on our framework to evaluate the usability guidelines. The technical details of our tool is available at [4].

### 5 EVALUATIONS

#### 5.1 Case Study & Results
To understand the robustness of the framework, we conducted a case study on enterprise android based mobile apps and captured results. Table 1 contains the list of 45 android mobile apps based on their business category and the app version used to perform the case study. The Contents of the table 1 are defined as ‘LOC’ - (Lines of Code), ‘SLOC-P’ - (Physical Executable Lines of Code), ‘P’ - Passed validation test cases per guideline, ‘F’ - Failed validation test cases per guideline and ‘%’ - Success Rate is the ratio between passed and failed validation test case(s)

**Case Study Setup** - We reviewed available enterprise apps in Android PlayStore and F-Droid. We have chosen 45 mobile apps which belongs to different domains like Writing Apps, Reading Apps, Money or Payment Apps and other General daily purpose Apps for our study. We reached out to the developer groups of these apps, obtained the debug mode version of their Mobile App. We conducted a 2 phase study as defined below:

- **Phase 1** - We used the debug mode mobile APK file and executed it against our Usability Evaluation Tool and captured Results.
- **Phase 2** - We performed a Cognitive walk-through experiment in the Usability Lab at IIIT Hyderabad, India with help of Usability Analysts to document the presence of these guidelines manually.

We have developed validation case(s) of 18 randomly selected usability guidelines for our case study [5], we are in process of

---

4[https://goo.gl/X68GKX](https://goo.gl/X68GKX)

5[https://goo.gl/6MVVhw](https://goo.gl/6MVVhw)
We conducted similar evaluations and obtained results for all the chosen mobile apps.

### 5.2 Industry Evaluation and Results

To understand the effectiveness of our framework, we have reached out to 16 premier Mobile App companies based out in India to deploy and evaluate our tool. These vendors build and manage mobile apps for Banking, Travel and Payment Gateway domain. We shared a survey consisting of 8 questions. The intent of the survey is to understand their current usability evaluation practices (Q1,2) and their experiences post usage of our tool (Q3,4,5). Below are the detailed observations of the survey results by respective Product Owners:

**Q1. Do you perform usability evaluation on your mobile app based on usability guidelines?** - 12 of 16 participants were not conducting usability evaluation based on usability guidelines. 4 of them were conducting usability evaluation based on usability guidelines.

**Q2. What challenges did you face while performing usability evaluation?**

expanding to more guidelines. Each guideline can consisted of a one or more validation test case(s). For example, the Fig 4 shows guidelines used along with failed and success validation test cases. We conducted similar evaluations and obtained results for all the chosen mobile apps.

### Table 1: Mobile Apps Used for Case Study

<table>
<thead>
<tr>
<th>App Name</th>
<th>Version</th>
<th>LOC</th>
<th>SLOC-P</th>
<th>p</th>
<th>f</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes</td>
<td>1.0.1</td>
<td>10358</td>
<td>9751</td>
<td>15</td>
<td>9</td>
<td>62.50</td>
</tr>
<tr>
<td>To-Do List</td>
<td>1.0.1</td>
<td>5711</td>
<td>5349</td>
<td>12</td>
<td>52.17</td>
<td></td>
</tr>
<tr>
<td>Omni Notes</td>
<td>3.2.2</td>
<td>117868</td>
<td>101868</td>
<td>13</td>
<td>12</td>
<td>52.00</td>
</tr>
<tr>
<td>Memento</td>
<td>1.1.0</td>
<td>12788</td>
<td>11831</td>
<td>16</td>
<td>9</td>
<td>64.00</td>
</tr>
<tr>
<td>Cloud Notes</td>
<td>0.11.0</td>
<td>59038</td>
<td>49326</td>
<td>10</td>
<td>10</td>
<td>50.00</td>
</tr>
<tr>
<td>Stotepad</td>
<td>1.2.1</td>
<td>6004</td>
<td>5712</td>
<td>11</td>
<td>11</td>
<td>50.00</td>
</tr>
<tr>
<td>SyncOrg</td>
<td>2.0.5.3</td>
<td>130466</td>
<td>109538</td>
<td>14</td>
<td>12</td>
<td>53.85</td>
</tr>
<tr>
<td>Kolab Notes</td>
<td>2.7.2</td>
<td>97644</td>
<td>68935</td>
<td>11</td>
<td>13</td>
<td>45.83</td>
</tr>
<tr>
<td>JustNotes</td>
<td>1.9</td>
<td>3112</td>
<td>2944</td>
<td>10</td>
<td>10</td>
<td>50.00</td>
</tr>
<tr>
<td>NoteBuddy</td>
<td>1.2.2</td>
<td>2950</td>
<td>2382</td>
<td>13</td>
<td>9</td>
<td>59.09</td>
</tr>
</tbody>
</table>

**Legends:** p- Passed, f- Failed (validation cases) and % - Passed Rate

**Figure 4: Usability Validation Test results**

**[Results]** - This section comprises of results from both the phases in detail.

#### Phase 1 - Table 1 provides us the evaluation results of 18 usability guidelines with the counts against pass (p) and fail (f) use cases along with the success percentage. List of 18 usability guidelines and detailed use-case results[22] are made available for review. We could see that our tool was able to identify the use-cases based on code analysis for each guideline. It was able to provide us the compliance of the respective usability guideline. We sould that three mobile apps namely ‘Froogal’, ‘Vlabs’ and ‘MyExpense’ has the highest conformance to most of the guidelines.

#### Phase 2 - Based on the above observations, we also conducted a Heuristic Cognitive Walk-through experiment using human subjects to examine these 18 usability guidelines (each app has multiple individual use-cases per guideline) on above 45 mobile apps. Here human participants were requested to evaluate the presence of the usability guideline manually. The results of this experiment were in line with our tool results and provided us a confidence to proceed with a wide-spread study in Industry.

6https://goo.gl/forms/dEDwDj2KvVMaUMrE2
usability evaluation on your mobile app? - Release deadlines, automation challenges, time consuming task, manual effort causing human errors and frequent changes in user requirements were top challenges. Lack of usability testing expertise and no clear usability strategy are few minor challenges.

Q3. How would you rate the ease of use of tool based on your experience? - Here observations were captured based on factors like Readability, Understandability, Deployment, Validation case implementation and overall setup experience with a Likert scale of “Very Easy”, “Easy”, “Somewhat Difficult” and “Difficult”. Overall 81% of the participants found that the tool setup process was easy to follow.

Q4. What are the additional automation strategies you would like to suggest? - Event watching, Event capture, Screen capture and User-Key store analysis were few of the automation strategies suggested to us. Code analysis with a feature to encrypt and decrypt the source code was also suggested. This was to avoid exposing the undesired source code developers who were not part of a particular module of the mobile app development. This requirement was intended to secure the source code logic across developers.

Q5. Do you plan to use tool for usability evaluation in future? - 13 out of 16 participants found that framework helped them address the usability evaluation requirements. They continue to practice this approach over their future apps to addressed usability issues. They were planning to build custom validation cases as per their custom usability requirement. Three participants were skeptical about future usage of the framework as it requires building their own set of validation case(s), they would use if framework provides an extensive list of existing validation case(s) out of the box.

6 LIMITATIONS & FUTURE WORK

Our usability evaluation framework is extensible i.e. it can be instantiated for any given mobile app as it is independent of any instantiable frameworks. As usability guidelines evolve over a time, any individual or an organization can adapt our framework to create their own instance to evaluate domain specific mobile applications. As part of our future work, we have a focused approach to accommodate our future research plan:

- Perform case-studies using .ipa files of iOS and .xap/.appx files of Windows for evaluation
- We have plans to integrate this tool with a Mobile Integrated Development Environment (IDE) for dynamic usability code analysis during development phase
- Build validation execution engine using open source platform to avoid dependency of .Net framework.

The purpose of any such tool is to assist human evaluation. It should be used with caution and one should not rely solely on them for results. Particularly because of the technology limitations, it is difficult to exhibit human-like behaviour like common sense. We have also encountered difficulty in incorporating certain guidelines into validation case because of their abstract nature. Few of our participants found creating validation case is a time-consuming task. However, if the framework is supplied with an exhaustive list of validation case using guidelines - it will be a great contributor for usability evaluation.

REFERENCES