# A. Artifact Appendix

## A.1 Abstract

We present here the artifacts that supports our ISSTA19 paper. We proposed QADroid. We implemented QaDroid using Flowdroid and tested it over multiple versions of Android apps. We compared the regression selection of events in QADroid against the total number of events. The artifacts supports all the claims we made in the Experiments section of the paper. We provide a way to run QADroid and reproduce the results of the experiments.

## A.2 Artifact check-list (meta-information)

- **Compilation:** Needs Java version "1.8.0_151", Java(TM) SE Runtime Environment (build 1.8.0_151-b12), Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
- **Data set:** 50 open source Android apps from GitHub (available on http://www.cse.iitm.ac.in/ rupesh/software/QADroid)
- **Hardware:** At least 32GB of memory with Linux operating system.
- **Software:** Android Development Environment, together with IntelliJ IDEA, Eclipse and Android Studio.
- **How much disk space required (approximately)?:** At least 50 GB free space.
- **How much time is needed to prepare workflow (approximately)?:** 2 hours
- **How much time is needed to complete experiments (approximately)?:** 10-12 hours
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** No
- **Data licenses (if publicly available)?:** No

## A.3 Description

### A.3.1 How delivered

Source code of the QADroid is included in the artifact and required data sets are uploaded publicly.

## A.4 Installation

Extract the QADroid Artifacts.tar.gz. After extracting, current di- rectory will have two directories - Offline, Online. Offline contains the code base for running the offline phase of the QADroid. It builds upon Flowdroid and generates the callbacks, call graphs for the first version of the app. Online phase contains the code base that compares the two versions of the app and list the events that are to be executed. The artifacts tar.gz also contains the utility jar that we have used to obtain the static metrics of the app. It also contains a script.sh file to execute the jar to obtain these statistics. ReleasesData.xlsx file is included that contains the manually evaluated data as presented in the $7^{th}$ to $11^{th}$ column in the Table 2 of the paper

## A.5 Experiment Workflow

1. The data in the first three columns (#Class, #Method and LOC) of the Table 2 in the paper were generated through a utility developed by Maurcio Aniche (SERG-Delft). The script "script.sh" executes over the tar.gz files of the source code obtained from GitHub and obtains the static metrics of the app. The utility's code base can be found here: https://github.com/mauricioaniche/ck.
2. The data in the columns $6^{th}$ to $10^{th}$ were obtained manually using the Github compare utility. The numbers in the columns indicate the number of commits, number of files changes, number of java files changes, Android Manifest file, and the layout files changes respectively in each of the releases of the apps. This data has been obtained manually as a part of the change-set validation task. The excel sheet "Releases-Data.xlsx" contains the compare urls of consecutive versions for all the apps. In order to manually validate the numbers against a particular release, please click on the url and verify the numbers.
3. First the code in the Offline directory needs to be setup in the Eclipse and run to obtain the callgraph, callbacks for the app. Next, the project in the Online directory need to be executed to obtain the number of selected activities and the number of selected callbacks. The script for obtaining the plots in Figure 8 and Figure 9 have not been included in the Artifacts.
4. **Software Requirement:** JDK 8 or later must be installed. To install Oracle JDK 8, please execute the following commands. (i) sudo add-apt-repository ppa:webupd8team/java, (ii) sudo apt-get update, (iii) sudo apt-get install oracle-java8-installer, and (iv) sudo apt-get install oracle-java8-set-default . Set the JAVA_HOME environment variable to the location of your JDK installation.
5. **Software Requirement:** Android SDK is needed for running this experiment. Android Studio for linux must be downloaded, installed, and extracted to home directory). Set the ANDROID_HOME environment variable to the location of your Android SDK installation. The following commands would help in the installation - (i) sudo gedit /.bashrc (ii) export ANDROID_HOME=/home/user_directory/Android/Sdk (iii) export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools You need to Install SDK Platforms. To install, open Android Studio and click on Configure -¿ SDK Manager to open Android SDK Manager. Install the packages for API 10 to 27. Android Studio can be launched by opening a terminal, navigating to the android-studio/bin/ directory, and executing studio.sh using the following commands - (i) cd android-studio/bin (ii) ./studio.sh