Test Automation in Open-Source Android Apps: A Large-Scale Empirical Study

Jun-Wei Lin, Navid Salehnamadi, and Sam Malek
School of Information and Computer Sciences
University of California, Irvine
ASE 2020

UCIrvine

Automated Testing of Mobile Apps

Received attention in recent years from both researchers and developers

Researchers

- Automated program repair, automated test transfer, mutation testing, …
- The extent mobile tests exist
- The type and quality of these tests
- Whether the tests are adopted in a particular way

Developers

- Why and how to adopt automated testing
- The impact on user satisfaction or project popularity in developer community

Automated Testing of Mobile Apps

Received attentio in recent years from both researchers and developers

Researchers

- Automated pr
- The extent
- The typ
- Whether

Developers

- Why and how to adopt automated tes
- The impact on user satisfaction or popularity in developer community

Holistic view from three complementary perspectives: **apps**, **developers**, and **impacts**

Differences from Prior Work

Scale

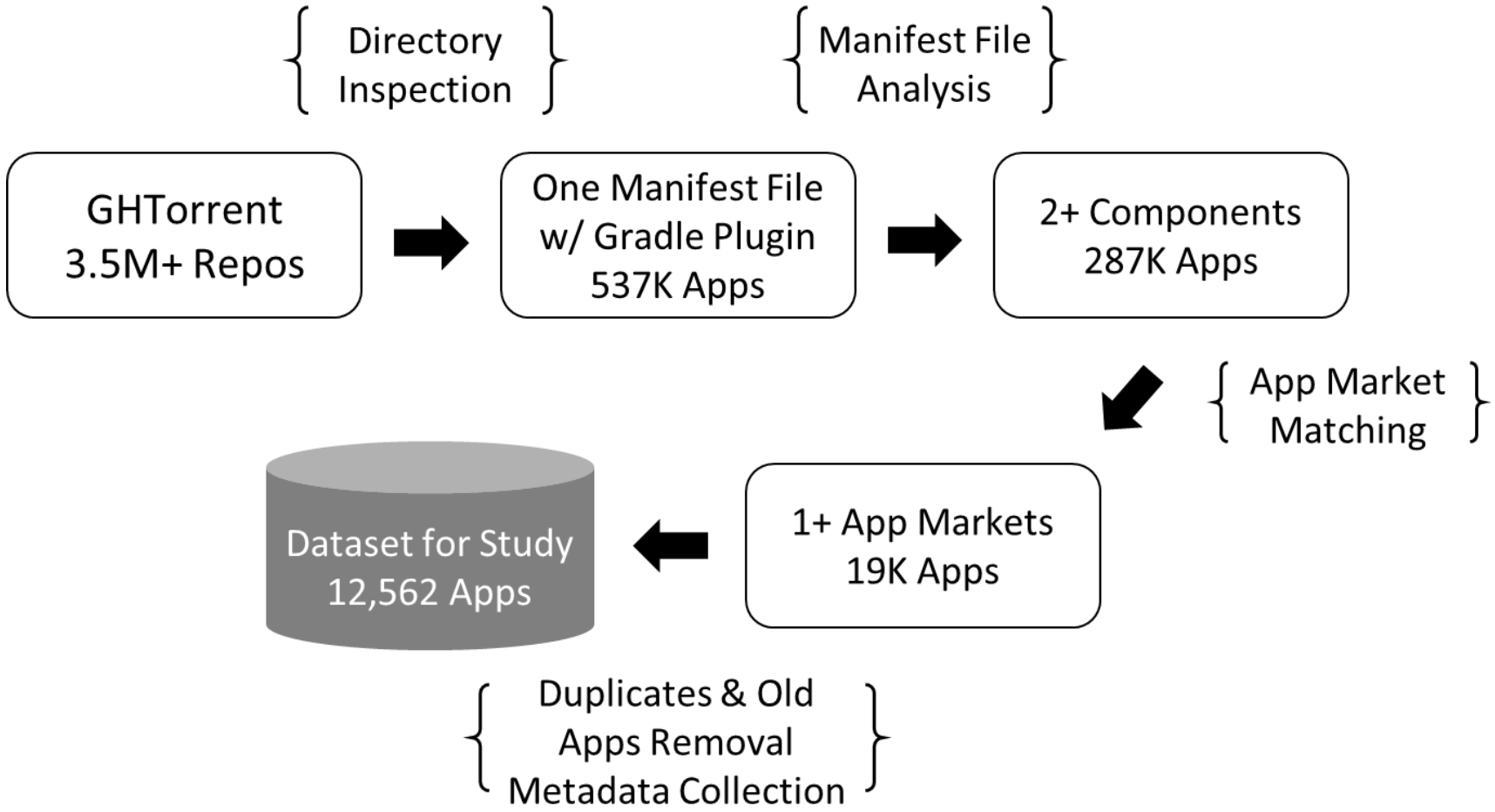
- 3.5M+ non-forked repositories on GitHub were scanned
- 12,000+ real-world apps across 16 app markets were identified
  (prior work: ~1000 apps on F-Droid/Google Play)

Curated dataset

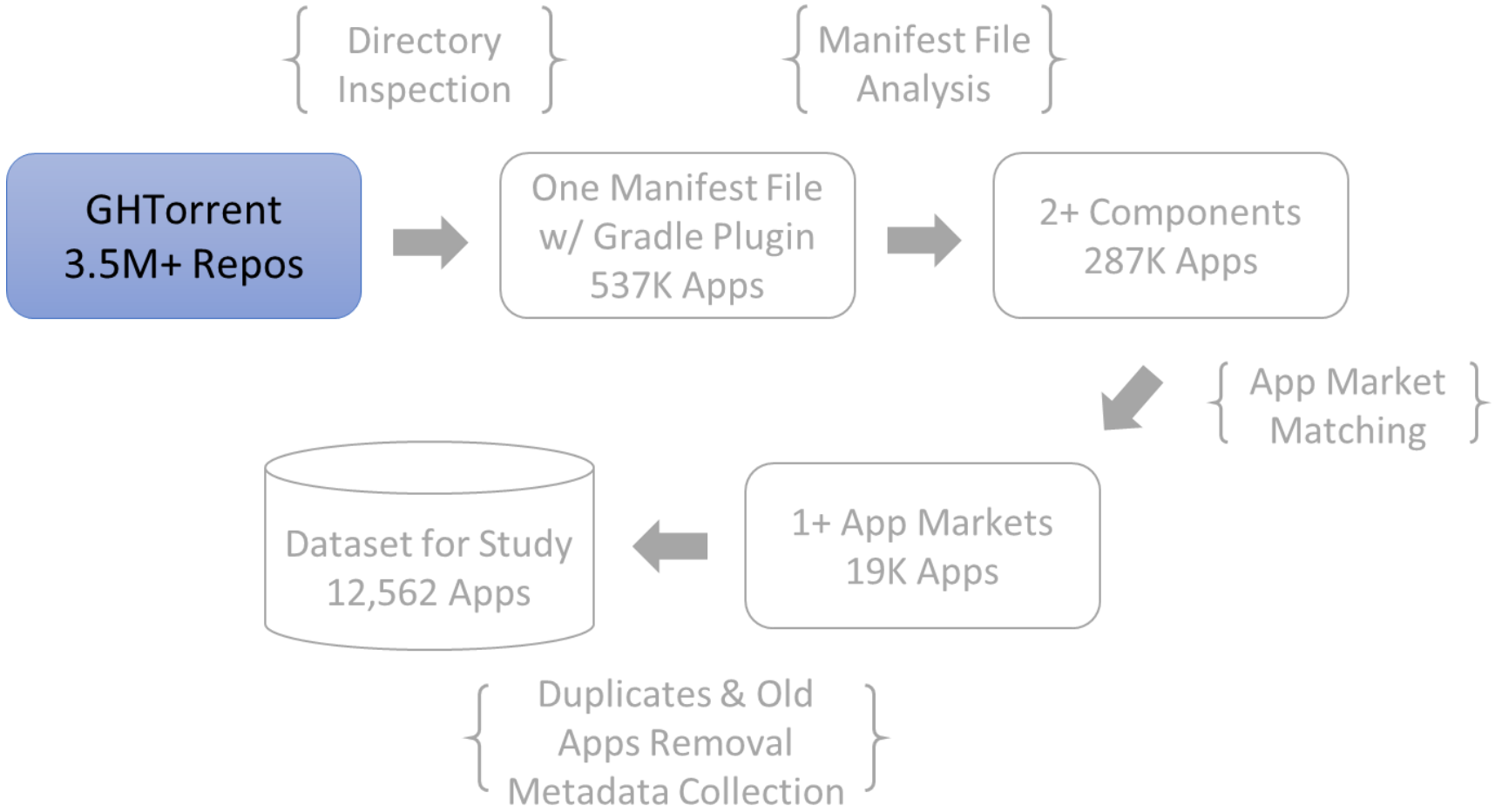
- Heuristics to identify non-trivial apps, e.g., apps published in an app market
- Excluding dummy tests generated by Android Studio, e.g., `ExampleUnitTest()`

Survey involving 148 contributors of the subject apps for the rationale behind our observations

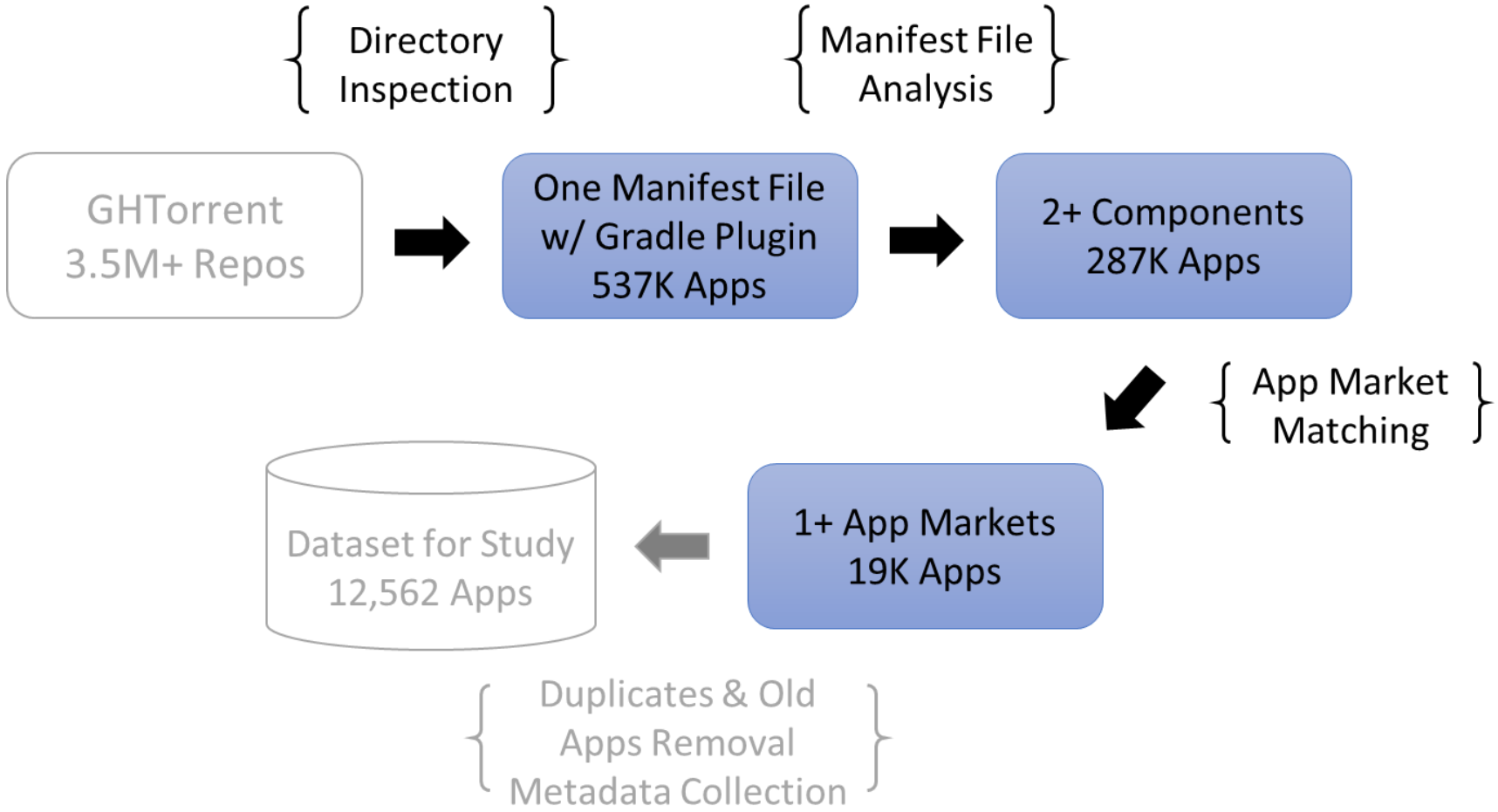
Data Collection



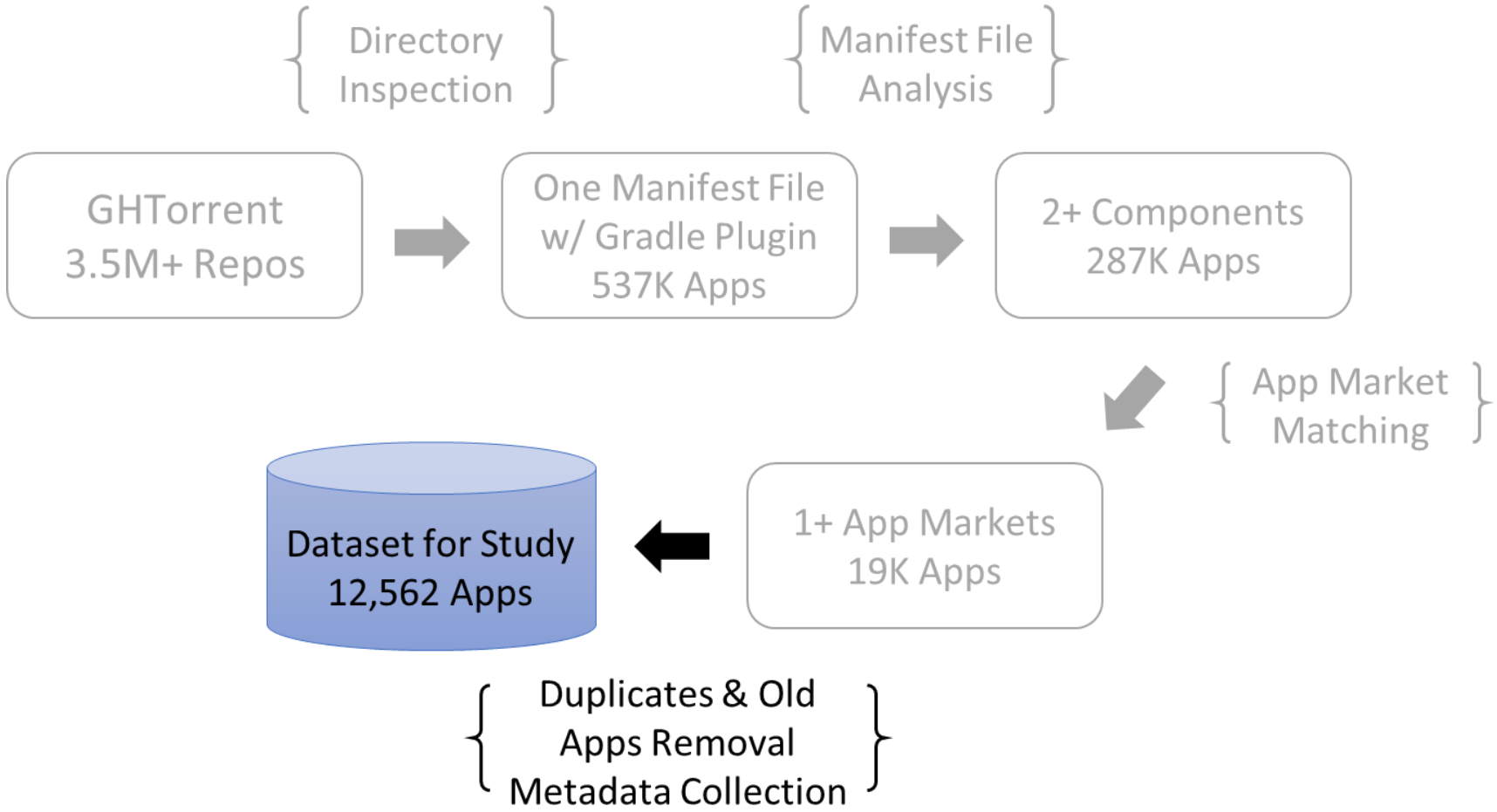
Data Collection



Data Collection



Data Collection



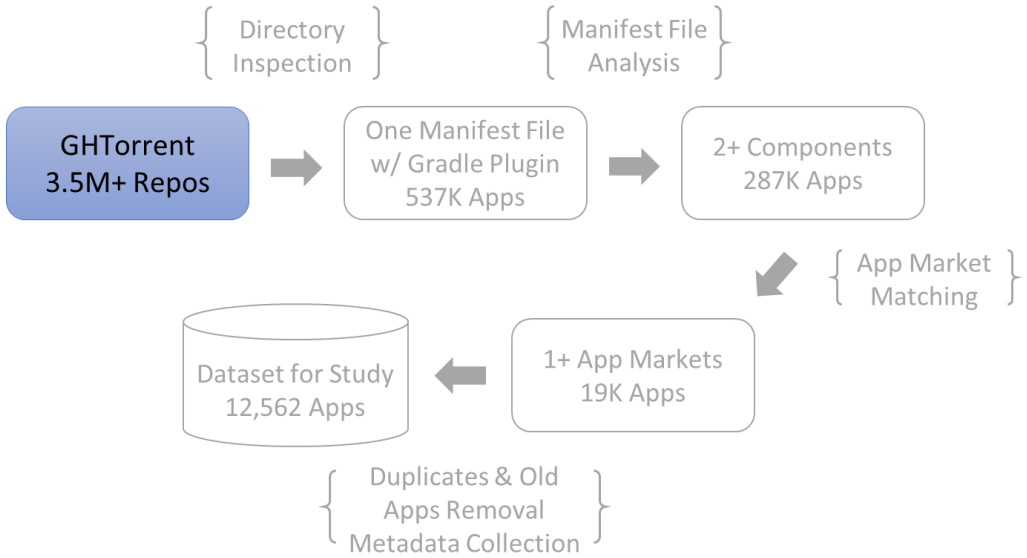
Initial List of GitHub Repos

GHTorrent DB

- A research project that monitors the GitHub public event timeline and populates the meta-data of the observed events

Query the GHTorrent DB for projects:

- Written in Java or Kotlin
- Non-forked, non-deleted
- 3.5M+ repos



Filtering Criteria

Goal: to identify non-trivial and real-world apps

Criterion 1

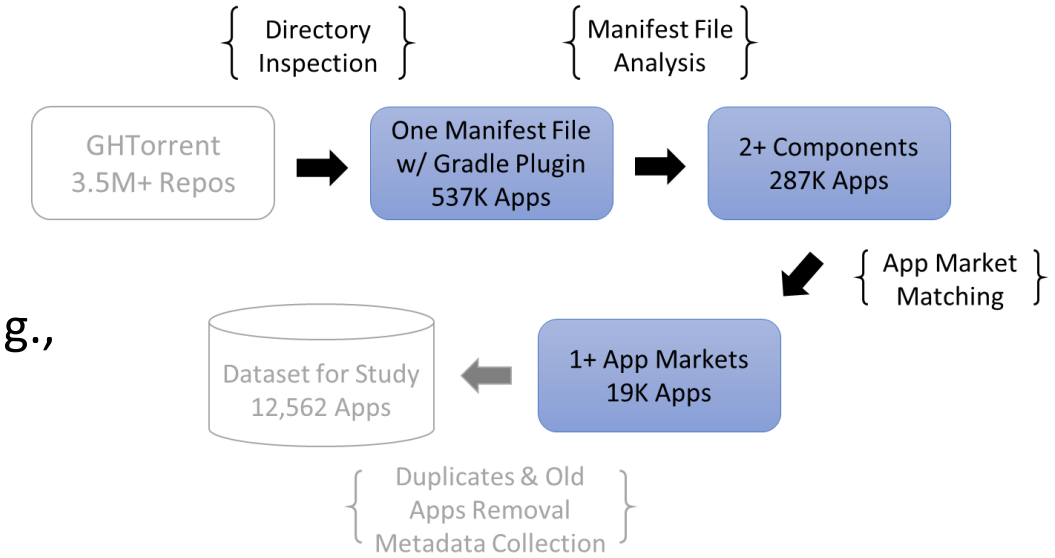
- Exactly one `AndroidManifest.xml`
- A task to build Android app in `build.gradle`

Criterion 2

- 2+ components declared in the manifest file

Criterion 3

- The package name must appear in an app market, e.g., Google Play, F-Droid, Anzhi



Data Cleaning and Meta-data Collection

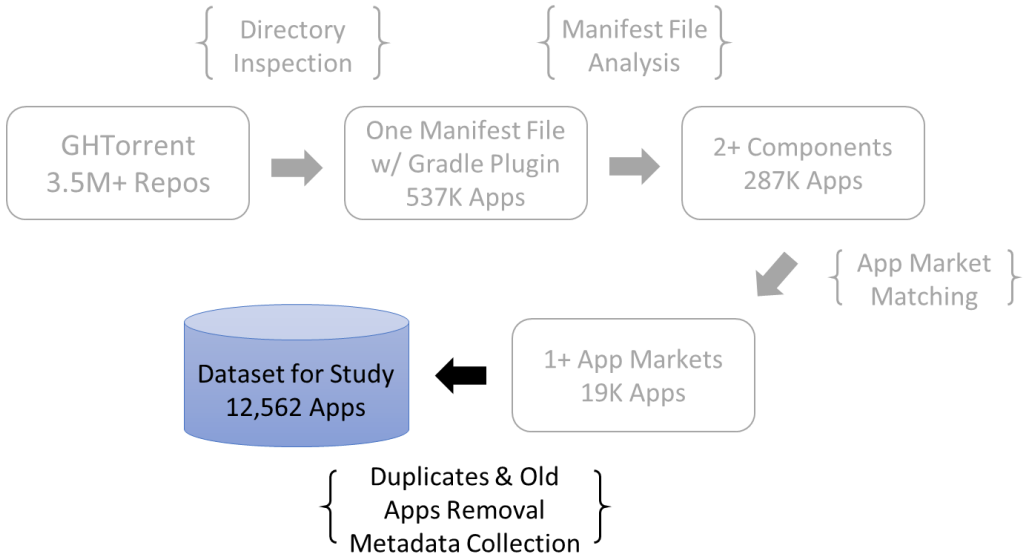
Remove apps

1) With duplicate package names
2) Before 2015

Collect additional meta-data

- Number of forks, stars, contributors, etc., on GitHub
- Category and user ratings on Google Play

Two months for the entire process



Top 10 App Market Distribution

Market*	#Apps
Google Play	11265
PlayDrone	539
fdroid	434
anzhi	408
appchina	294
mi.com	70
VirusShare	62
angeeks	41
1mobile	26
freewarelovers	12

Year Distribution

Year Created	#Apps
2015	3614
2016	2330
2017	1731
2018	2898
2019	1989
Total	12562

Counting Tests

Assumptions for the apps:

- Using JUnit-based testing frameworks, e.g., JUnit, Espresso, Robolectric, Mockito
- Developed with Android Studio

A method annotated with "@Test" is considered as a test case

- Used by JUnit-based testing frameworks

Exclude the placeholder tests automatically generated by Android Studio, e.g., *ExampleUnitTest.java*, *ExampleInstrumentedTest.java*

Developer Survey

Goal: To understand the rationale behind our findings from the dataset

Participant: the main contributor of each subject app

- e.g., project owner, contributor who has made the most commits, etc.

Questions

- Demographic, e.g., country, Android development experience
- Current practices of Android app testing
- Opinions about our findings in the correlation analysis between the adoption of test automation and the popularity of apps

148 complete responses from 6,837 unique emails sent successfully

Research Questions

App

- How prevalent is test automation in open-source Android apps?
- Is the prevalence of test automation varied across different categories of apps?

Developer

- How prevalent is test automation and what are the reasons for not adopting it?
- The biggest challenges in Android testing?
- The most useful criteria for evaluating Android tests
- Do developers prefer unit or UI testing and why?
- Do the same developers have the same testing habits across apps?
- Is the practice of Test Pyramid followed by developers?

Impact

- How does test automation relate to project popularity?
- How does test automation relate to user satisfaction?

Research Questions

App

- How prevalent is test automation in open-source Android apps?
- Is the prevalence of test automation varied across different categories of apps?

Developer

- How prevalent is test automation and what are the reasons for not adopting it?
- The biggest challenges in Android testing?
- The most useful criteria for evaluating Android tests
- Do developers prefer unit or UI testing and why?
- Do the same developers have the same testing habits across apps?
- Is the practice of Test Pyramid followed by developers?

Impact

- How does test automation relate to project popularity?
- How does test automation relate to user satisfaction?

Research Questions

App

- **RQ1. How prevalent is test automation in open-source Android apps?**
- Is the prevalence of test automation varied across different categories of apps?

Developer

- How prevalent is test automation and what are the reasons for not adopting it?
- The biggest challenges in Android testing?
- The most useful criteria for evaluating Android tests
- Do developers prefer unit or UI testing and why?
- Do the same developers have the same testing habits across apps?
- Is the practice of Test Pyramid followed by developers?

Impact

- How does test automation relate to project popularity?
- How does test automation relate to user satisfaction?

RQ1. Prevalence of Test Automation

Group	#Apps	Percentage
Apps with any tests	1002	7.98%
Apps without tests	11560	92.02%
Apps with unit tests	766	6.10%
Apps with UI tests	502	4.00%
Apps with both unit and UI tests	266	2.22%

Only 7.98% of the subject apps contain tests. Much lower than previous findings (14%-40%)

56% (7017/12562) subject apps contain placeholder tests (and were excluded)

Research Questions

App

- How prevalent is test automation in open-source Android apps?
- Is the prevalence of test automation varied across different categories of apps?

Developer

- **RQ2. How prevalent is test automation and what are the reasons for not adopting it?**
- The biggest challenges in Android testing?
- The most useful criteria for evaluating Android tests
- Do developers prefer unit or UI testing and why?
- Do the same developers have the same testing habits across apps?
- Is the practice of Test Pyramid followed by developers?

Impact

- How does test automation relate to project popularity?
- How does test automation relate to user satisfaction?

RQ2. Prevalence of Test Automation (Reported by Developers)

How Developers Test Their Apps

Way	#Respondents
Manually	130
With scripted/automated tests	85
With dedicated QA team or 3rd party testing services	43
With automatic input generation tools	
Other	
Not at all	

57% (85/148) **says** they are using, but only 8% **observed**

RQ2. Difficulties in Adopting Test Automation (Reported by Developers)

The app is not big or complex enough to require automated tests

Difficulty	
Cost to create and maintain automated tests	77
Time constraints	74
Size or maturity of the app	66
Lack of exposure or knowledge of existing frameworks	52
Cumbersome to use	50
Lack of support from management or organization	30
Other	11

Research Questions

App

- How prevalent is test automation in open-source Android apps?
- Is the prevalence of test automation varied across different

Developer

Popularity metrics that mobile developers care about

- Attention received from other developers, e.g., stars, forks, contributors on GitHub
- User ratings on the market

Whether these popularity metrics are impacted by the adoption of test automation

- Is the practice of Test Pyramid followed by developers?

Impact

- **RQ3. How does test automation relate to project popularity?**
- **RQ4. How does test automation relate to user satisfaction?**

Statistical Analysis

Two disjoint sets with equal size

- R_w: metric values (e.g., number of stars) from the **apps with tests**
- R_o: metric values from the **apps without tests**

Compute the mean and median of R_w and R_o, and the difference between the mean and median

Perform hypothesis testing to determine if the difference observed in R_w and R_o is statistically significant

Repeated for all the popularity metrics

RQ3. How does test automation relate to project popularity?

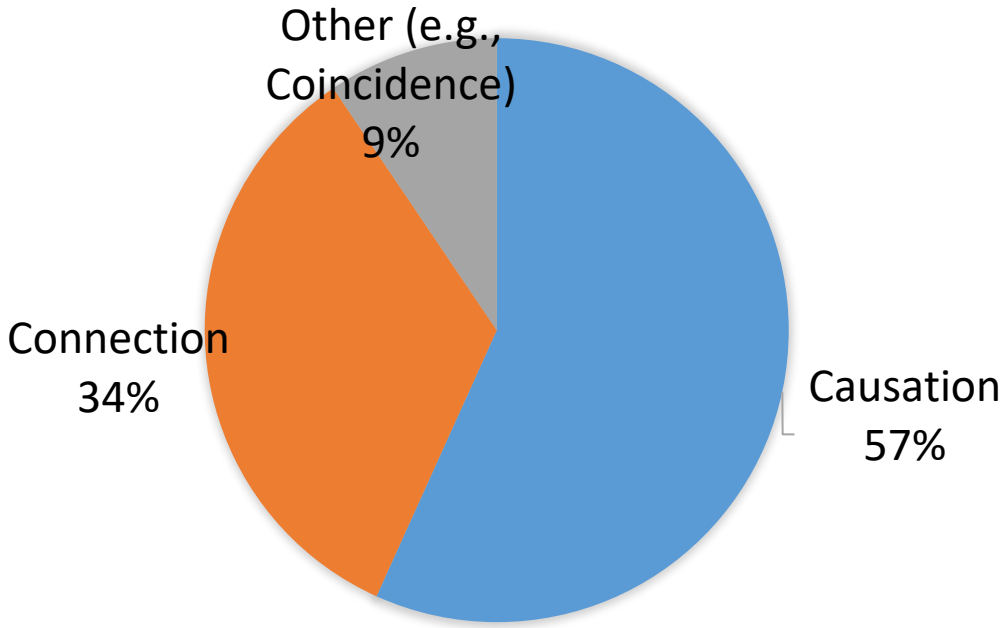
Impact of Having Tests on the Popularity of Apps.
R_w: Apps with Tests. R_o: Apps Without Tests.

	Stars*			Forks*			Contributors*		
	Size	Mean	Median	Size	Mean	Median	Size	Mean	Median
R_w	629	10.95	0	630	3.74	0	630	2.76	2
R_o	629	4.57	0	630	1.31	0	630	1.63	1
Δ		6.38	0		2.43	0		1.13	1

	Commits*			Issues*			Pull Requests*		
	Size	Mean	Median	Size	Mean	Median	Size	Mean	Median
R_w	628	147.21	84.5	635	10.39	0	633	8.76	0
R_o	628	39.93	14	635	1.35	0	633	0.85	0
Δ		107.28	70.5		9.04	0		7.91	0

*The difference is statistically significant.

Thoughts from the



*"**I would say they have a direct connection** since the quality and rigidness of the app's code can definitely influence an app's popularity."* **(direct)**

"First you build the app, then **it gets popular, then you get resources/motivation to increase it's quality**. That's when you go to UI tests." (reverse)

*"Projects that become popular end up writing more tests because they need to ensure the stability of the project. As the project becomes more stable (due to more testing) it **provides a positive feedback loop**."* (bidirectional)

RQ3. How does test automation relate to project popularity?

Popular projects are more likely to adopt test automation practices. 57% of the developers believe it implies causality between them.

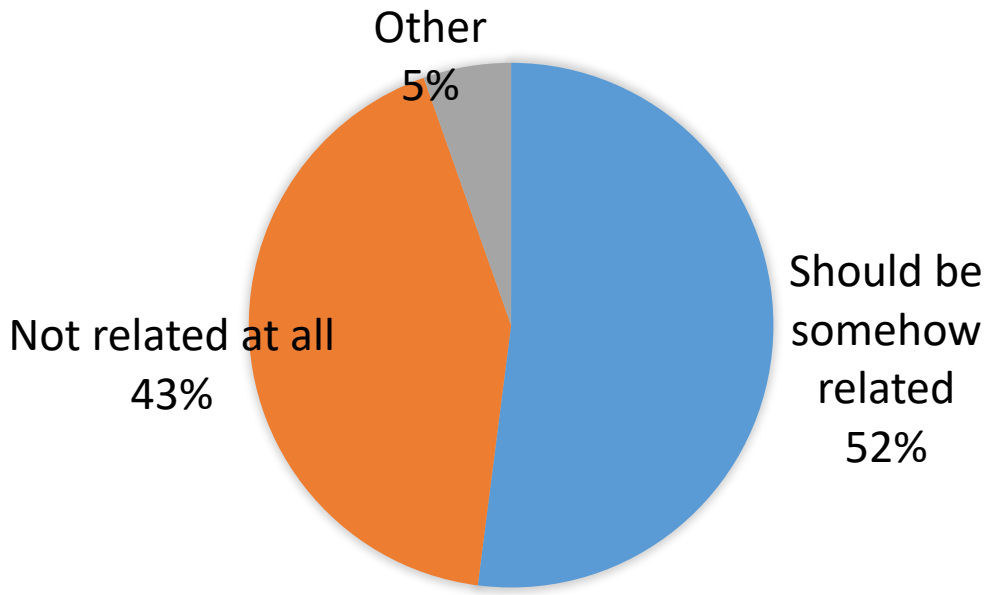
RQ4. How does test automation relate to user satisfaction (ratings on Google Play)?

Impact of Having Tests on the User Satisfaction of Apps.
R_w: Apps with Tests. R_o: Apps Without Tests.

	Size	Mean	Median	p-value
R_w	211	4.14	4.25	0.0689
R_o	211	4.2	4.33	
Δ		-0.06	-0.08	

No association found between them with statistical significance

Thoughts from the



*"Play Store ratings are a noisy metric of app quality and overall user experience, so the **no apparent correlation doesn't convince me** that app quality isn't impacted at least somewhat by automated testing"*

***"I think it would depend on the type of application.** Games and such are harder to test and the quality of test does not correlate with how fun the game is. For a banking application tests are essential and do effect the quality of the final product."*

RQ4. How does test automation relate to user satisfaction (ratings on Google Play)?

Users' satisfaction with apps appears to be unrelated to the adoption of test automation, while half of the developers think differently.

Discussion

Automated testing is not widely adopted

Only 8% of the subject apps have adopted automated testing

- Contradicts earlier studies that have reported higher adoption rate (14%-40%)
- But in line with the general perception that it is challenging to find open-source apps with tests for research purposes

Dataset of real-world and non-trivial apps with automated tests

- For emerging research topics, e.g., automated program repair, automated test transfer, and mutation testing

Discussion

Automated testing can be useful and important

Strong correlation between the adoption of test automation and the popularity of development projects

- Most of the survey respondents (91%) believe that the correlation is either causation or a connection

Users' satisfaction appears unrelated to test automation

- But a considerable amount of survey participants (52%) think that automated testing has impacts on users' satisfaction

Discussion

Tools and libraries have room for improvement

One of the difficulties reported by developers in adopting test automation is cumbersome tools

- Steep learning curve, poor documentation, usability, and compatibility issues

UI testing tools could be improved by addressing developers' concerns about speed, simpleness, and robustness

- "headless mode" (such as done by Robolectric) to accelerate execution of UI tests
- Better Interactive tools (such as Espresso Test Recorder) to help developers create UI tests
- Efforts to prevent or resolve flakiness of UI tests

Conclusion

A holistic view regarding how and why test automation is (not) practically adopted in open-source Android apps

- 12,000+ real-world and non-trivial apps on GitHub
- A survey of 148 developers of these apps
- https://github.com/seal-hub/ASE20Empirical

Findings to guide the current practices and future research directions

Future work

- More open-source projects, e.g., apps hosted on Bitbucket
- New research questions, e.g., continuous integration practices

Thank You!