



CSMR-WCRE 2014: SQM 2014

Exploring Development Practices of Android Mobile Apps from Different Categories

By

Ahmed Abdel Moamen

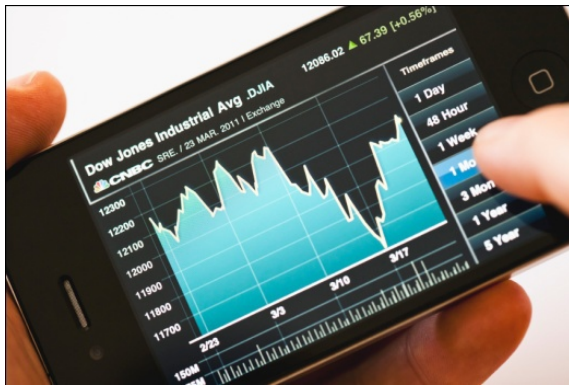
Chanchal K. Roy

Department of *Computer Science*



UNIVERSITY OF
SASKATCHEWAN

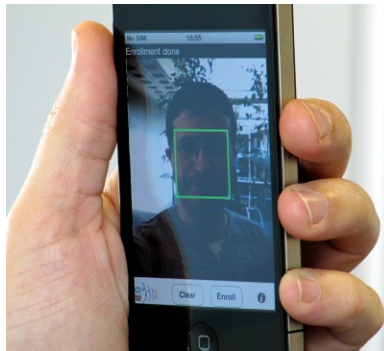
Mobile Devices are Ubiquitous



- In 2014, it is expected that 40% of all Internet traffic will be attributable to the smartphone [\[eMarketer, 2013\]](#)
- Smartphone ownership rose by an annualized rate of 64% in 2Q-2013 [\[Comscore in Wall Street Journal, 2013\]](#)
- One in four Americans owns smartphone [\[Engadget, 2013\]](#)
- Apple Inc sold 20 million iPads in 2013 [\[Engadget, 2013\]](#)

Categories of Mobile Apps

Utilities



M-Commerce



M-Learning



Mobile Healthcare



Games





Number of Mobile Apps in 2013

Company	Platform	Number of Apps
Apple	iOS	650,000
Google	Android	500,000
RIM	BlackBerry	100,000
Microsoft	Windows	50,000



Development Practices of Mobile Apps

- Existing software engineering knowledge may not hold in mobile apps.
- Mobile devices place a restriction on the developer
 - Memory consumption and availability
 - How to handle memory and how the UP components be displayed?
- Apps development are different from the traditional software systems.
- Understanding development practices may reduce the effort in developing mobile apps.
- **This paper explores the code characteristics and development practises of Android apps.**



Paper Outlines

- Contributions
- Findings
- Case Study I: Software Analysis for Mobile Apps
- Case Study II: Exploring the Development of Mobile Apps from Different Categories
- Conclusions



Main Contributions

- Explore and compare the development practises of mobile apps from different categories.
- Study more complex software engineering metrics
 - e.g., count of coupled and derived classes, max inheritance tree, and average cyclomatic complexity)
- The largest dataset (i.e., thirty apps).



Interesting Findings

- Game apps tend to be smaller in the size of code.
- There is an inverse relationship between the number of classes and the platform dependency.
- Android mobile apps rely much more on platform APIs.
- Most of mobile apps show very little growth of its size after the initial commit.
- The case study shows a nearly total absence of inheritance in mobile apps.



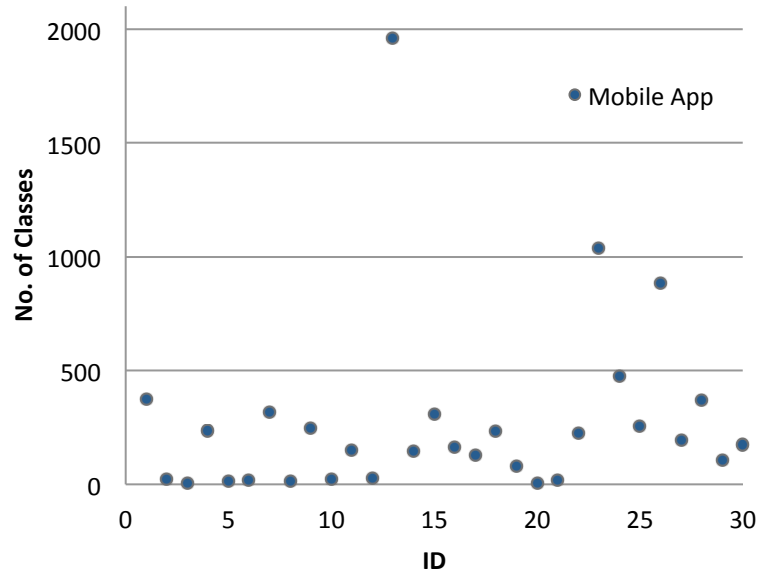
Case Study I: Software Analysis for Mobile Apps

- We study thirty Android apps.
- Choosing criteria:
 - The popularity amongst users (measured by number of downloads).
 - Have accessible source code repositories and issue tracking systems.
 - Mobile apps are taken from a number of different categories
- Software engineering metrics:
 - Size of Code and Development Team
 - Platform and Third Party Libraries Usage

Selected Mobile Apps

ID	Name	Category	Rate	Installs
A1	ConnectBot	Communication	4.7	1-5 m
A2	ALogcat	Development	4.6	100-500 k
A3	AndLess	Multimedia	4.2	100-500 k
A4	Barcode Scanner	Utility	4.2	50-100 m
A5	Frozen Bubble	Game	4.3	1-5 m
A6	Mythmote	Multimedia	4.6	50-100 k
A7	Sipdroid	Communication	4.0	1-5 m
A8	Solitaire	Game	4.5	10-50 m
A9	Apps Organizer	Utility	4.6	1-5 m
A10	AppSoundmanager	Multimedia	3.8	100-500 k
A11	Android VNC	Communication	4.3	1-5 m
A12	Anstop	Utility	3.9	10-50 k
A13	Csipsimple	Communication	4.4	.5-1 m
A14	Diskusage	Utility	4.7	.5-1 m
A15	Mythdroid	Communication	4.1	100-500 k
A16	Open GPSTracker	Communication	4.2	100-500 k
A17	Opensudoku	Game	4.6	1-5 m
A18	Replicaisland	Game	4.2	1-5 m
A19	Ringdroid	Multimedia	4.6	10-50 m
A20	Search Light	Utility	4.7	100-500 k
A21	Share My Position	Communication	4.3	10-50 k
A22	Zirco Browser	Utility	3.8	10-50 k
A23	K-9 Mail	Communication	4.3	1-5 m
A24	KeePassDroid	Utility	4.7	.5-1 m
A25	Aedict	Utility	4.7	50-100 k
A26	Book Catalogue	Utility	4.5	50-100 k
A27	KeepScore	Game	4.7	10-50 k
A28	Wordpress	Communication	4.2	1-5 m
A29	Nectroid	Multimedia	4.7	.5-1 m
A30	ancal	Utility	3.2	10-50 k

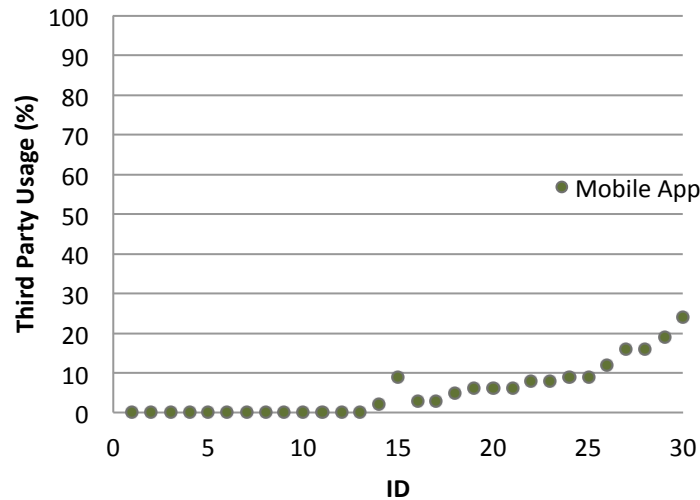
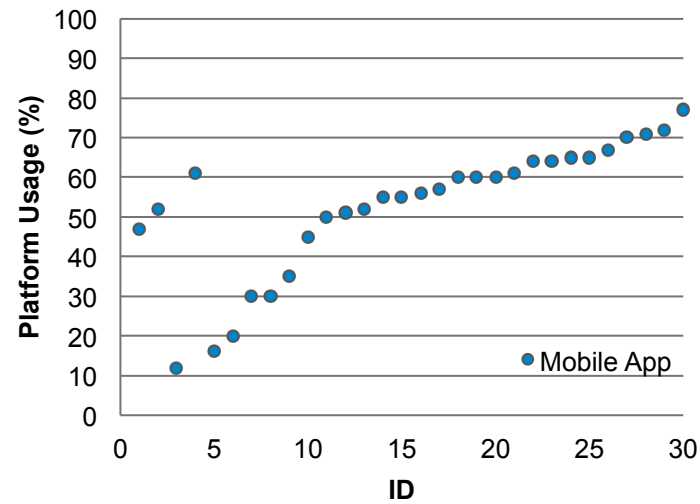
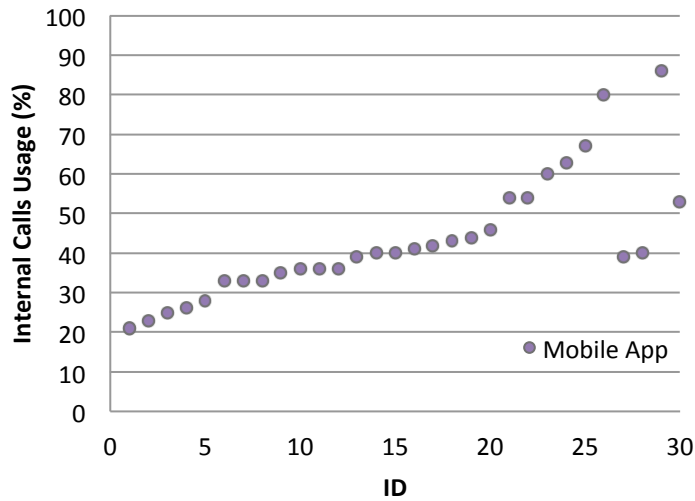
Size of Code and The Development Team



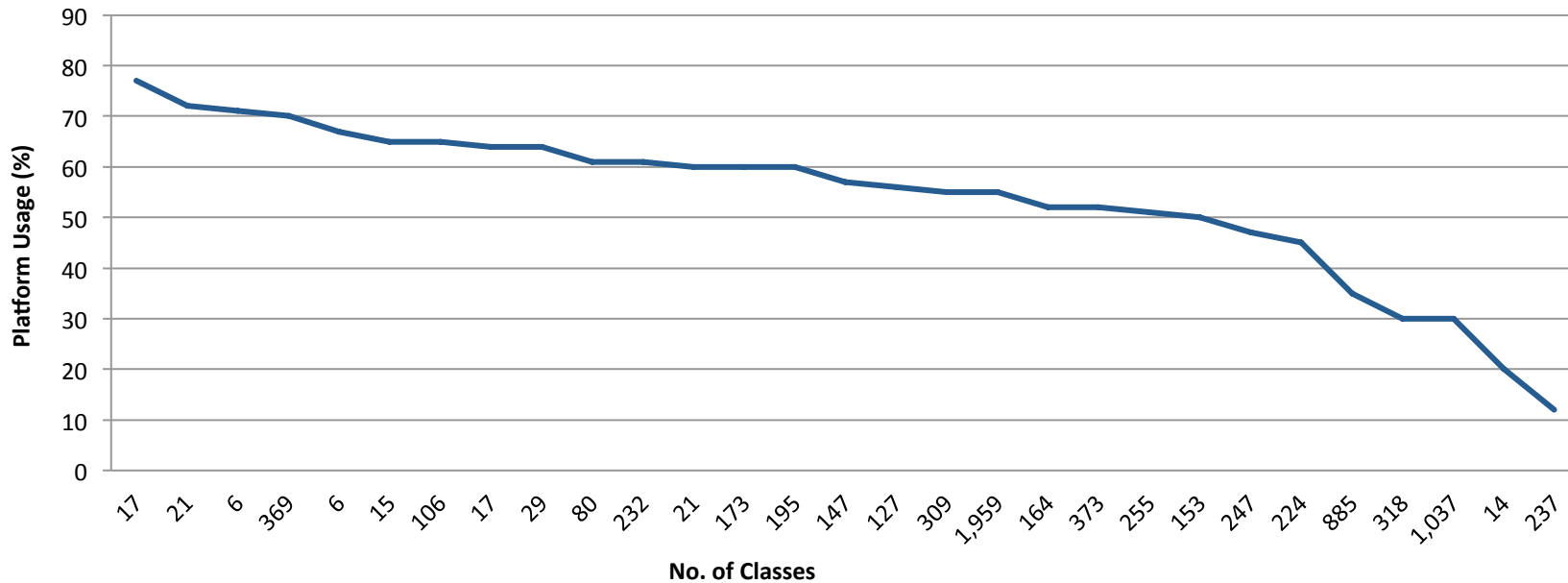
App ID vs. Number of classes

- Size of code ranges between 237 and 704,195 LOC
 - Median 12 KLOC
- Mobile apps have small code bases and development teams.
- Games have small size compared to other categories.
- Fewer developers contribute to mobile app projects.

Platform and Third Party Libraries Usage



The number of classes versus the platform dependency



- ❑ There is an inverse relationship between the number of classes and the platform dependency.
- ❑ Smaller mobile apps depend more on the Android platform?



Case Study II: Exploring the Development of Mobile Apps from Different Categories

- Study four mobile apps from different categories in more depth.
 - Communication and networking, game, utility and multimedia
- Software engineering metrics:
 - The project metrics
 - Source Code Dependency Metrics
 - The growth of lines of code over time
 - Code Churn Properties
 - Class Metrics



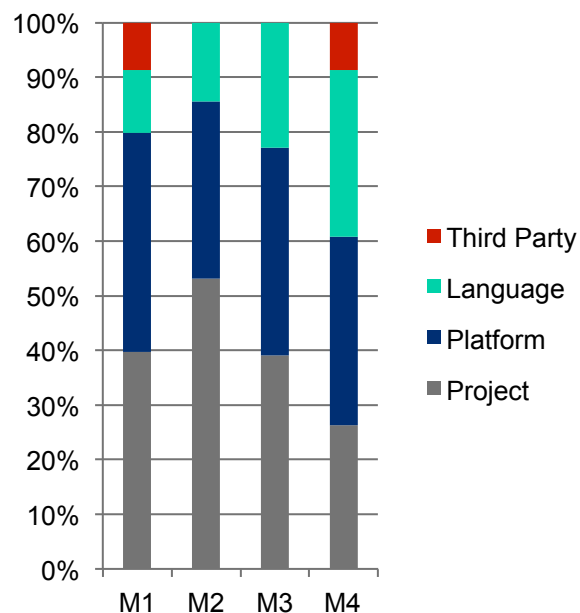
Selected Mobile Apps

ID	Name	Category	First Commit	Files	Classes	Functions	LOC	Dec. Stmt.	Exe. Stmt.	CCR
M1	ConnectBot	Communication tool	16/11/2007	331	373	2,052	34,004	8,213	15,011	0.35
M2	Replicaisland	Game	18/03/2010	242	232	1,581	25,181	6,366	10,067	0.17
M3	Apps Organizer	Utility	15/08/2009	167	247	1,471	14,198	4,315	4,980	0.6
M4	Nectroid	Multimedia tool	15/07/2010	106	106	615	12,711	3,142	4,828	0.51

■ Choosing Criteria:

- Category: one app from each category
- Simplicity: the code base for the mobile app can be easily identified
- Significant code base: mobile apps must have at least 100 source code files

Source Code Dependency Metrics

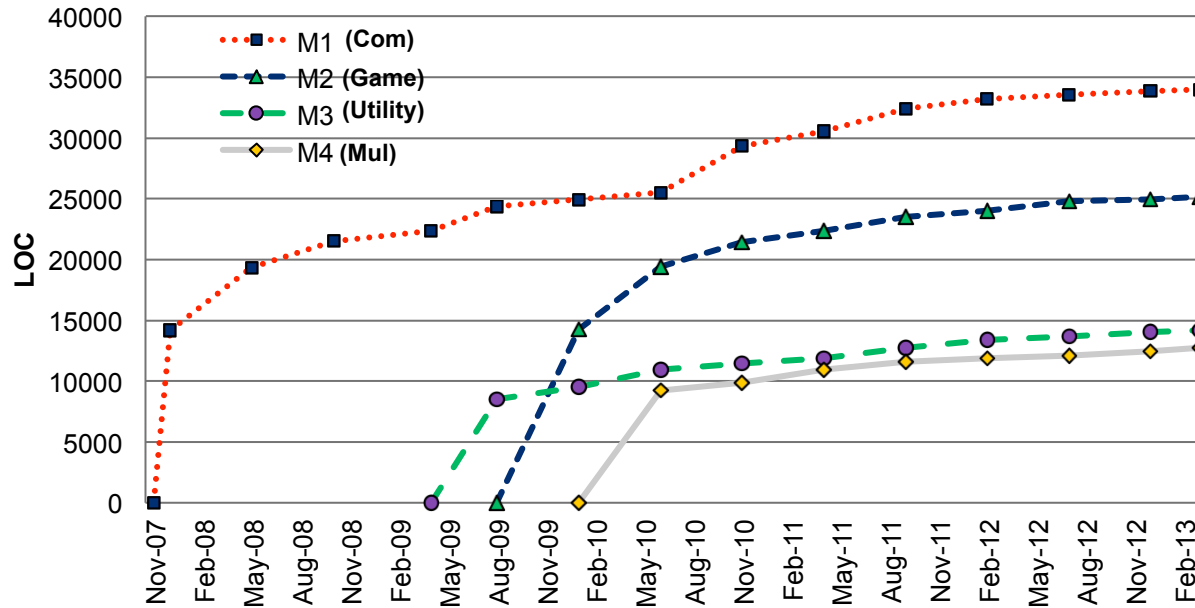


Source Code Dependency

M1: Com, M2: Game, M3: Utility, M4: Multi

- Android mobile apps rely much more on platform and language APIs.
- Third party dependencies account for fewer dependencies than any other dependency categories
- The Android and Java APIs appear to provide most of the dependencies of the mobile apps
- Over 56% of the Android mobile app dependencies are on the Android and Java APIs

The growth of lines of code over time



- The figure shows that source code files in communication and game apps change more frequently than other categories.

Code Churn Properties

Metric	M1	M2	M3	M4
Total No. File Changes	900	647	191	423
No. Project File Changes	863 (96%)	647 (100%)	191 (100%)	384 (91%)
No. Third Party File Changes	37 (4%)	0 (0%)	0 (0%)	39 (9%)
Total No. Line Changes	19,326	37,451	2,423	82,415
No. Project Line Changes	18,245 (94%)	37,451 (100%)	2,423 (100%)	74,235 (90%)
No. Third Party Line Changes	1,081 (6%)	0 (0%)	0 (0%)	8,180 (10%)

- The average LOC per change in all mobile apps is 72 LOC per change.
- This indicates that Android mobile apps see many small changes
- The third party source code has very little code churn compared to the project source code.

Class Metrics

M1: ConnectBot							
	Avg. LOC	Method Count	Variable Count	Count of Cou-pled Classes	Count of De-ri-ved Classes	Max Inheritance Tree	Avg. Cyclo-matic
Average	10.32	10.21	4.8	5.49	0.16	1.2	1.96
Median	7	2	0	3	0	1	1
Sum	4335	4083	1918	2048	59	449	823
Max	82	259	224	49	6	4	19
M2: ReplicaIsland							
	Avg. LOC	Method Count	Variable Count	Count of Cou-pled Classes	Count of De-ri-ved Classes	Max Inheritance Tree	Avg. Cyclo-matic
Average	8.27	13.24	3.54	5.65	0.53	2.47	1.72
Median	6	4	0	3	0	2	1
Sum	1981	1596	427	1357	128	592	412
Max	70	213	41	94	41	5	12
M3: Apps Organizer							
	Avg. LOC	Method Count	Variable Count	Count of Cou-pled Classes	Count of De-ri-ved Classes	Max Inheritance Tree	Avg. Cyclo-matic
Average	5.4	9.31	0.86	3.86	0.5	1.31	1.22
Median	4	1	0	2	0	1	1
Sum	1630	2785	258	1104	144	374	367
Max	32	893	42	25	13	4	8
M4: Nectroid							
	Avg. LOC	Method Count	Variable Count	Count of Cou-pled Classes	Count of De-ri-ved Classes	Max Inheritance Tree	Avg. Cyclo-matic
Average	6.26	6.91	1.2	4.21	0.14	1	1.34
Median	4	1	0	3	0	1	1
Sum	1077	968	166	577	20	139	231
Max	62	477	83	34	4	4	18



Class Metrics

- Android mobile apps' developers tend to write small size classes (in terms of LOC).
- The average number of local variables defined inside each class is too small (i.e., 3 variables).
- There are excessive coupled classes in Android apps that may prevent reuse of existing components.
- The average count of derived classes is 0.33, which shows a nearly total absence of inheritance.
- The average cyclomatic complexity for all nested functions or methods in each class is very low (i.e., 2).



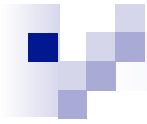
Threats to Validity

- The studied mobile apps represent a small subset of the total number of available apps.
- We have limited our study to a single mobile platform (i.e., the Android Platform).
- A number of third party tools were used in conducting the case study.
- The identification of third-party libraries in each app was done using heuristics and manual analysis.



Conclusions

- The goal of this paper is to understand how high quality and successful mobile apps can be developed and maintained.
- Mobile apps have unique code characteristics.
- We found that mobile apps of different purposes differ in different ways.
- Mobile apps bring a unique set of challenges to software engineering practice and research.



Thank You!

Questions & Answers