Evaluating Maintainability of Android Applications: Mooncascade Case Study

Mustafa Ogün Öztürk

Supervisor: Jakob Mass

MSc. Software Engineering Curriculum

Tartu 2021

Outline

- Introduction
- Problem statement
- Research methodology
- Results
- Conclusion

Android

- In the last decade smartphones and mobile applications became people’s primary way of interacting with technology.
- Mobile apps became a vital part of daily and business life, from ordinary people to large companies.
- As a successful mobile OS, Android has been a core element of this change, and the demand for Android applications has increased.

Mooncascade

- Product development company based in Estonia.
- Provides different software development services.
- With the increasing demand for Android applications, the company has been facing the some challenges when providing Android application development services to different customers.

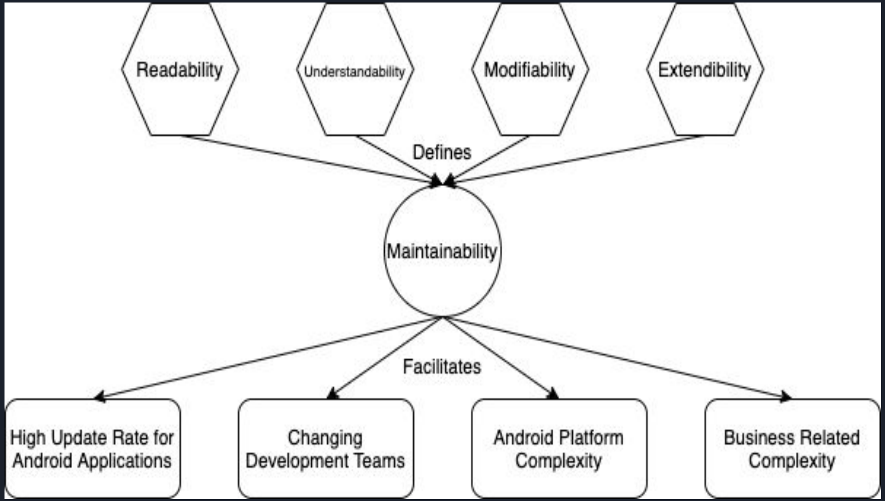
Challenges of Android Development

These challenges can be examined under four major topics.

- Android nature and its platform-specific components
- Sophisticated business needs
- The frequent update rate of Android applications
- Changing development teams.

Maintainability

- To overcome these challenges, "maintainability" emerges as one of the most important non-functional requirements.
- It is the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed.
- Developing maintainable Android applications is the way to survive in the competitive market time and cost-effectively.


Readability
Understandability
Modifiability
Extendibility
Defines
Maintainability
Facilitates
High Update Rate for Android Applications
Changing Development Teams
Android Platform Complexity
Business Related Complexity

Android Development at Mooncascade

- Kotlin (Programming language)
- SOLID Principles (Software design principles)
- Clean Code Principles (Coding conventions)
- Clean Architecture (Architectural pattern)
- MVVM (Presentational design pattern)
- RxJava (Asynchronous event handling library)
- Dagger 2 (Dependency injection framework)
- Retrofit or Apollo (Networking library)
- Android Architecture Components (Android libraries for maintainable apps)

Research Questions

Although these methods and technologies are widely known by the Android community and are believed to be useful for maintainability, their impact on maintainability is empirically unknown. To find a suitable method for maintainability evaluation and then evaluate the impact of these tools, techniques, and technologies used by Mooncascade's Android team in terms of maintainability, this study will answer the following research questions.

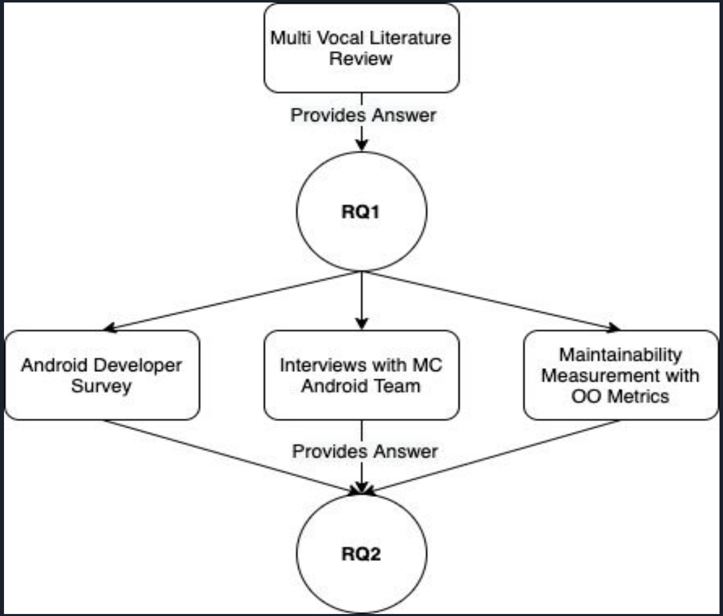
- RQ1: What is the proper way to evaluate the maintainability of the Android applications?
- RQ2: What is the impact of the methods and technologies used by Mooncascade to develop Android applications on the maintainability of Android applications?

Motivation

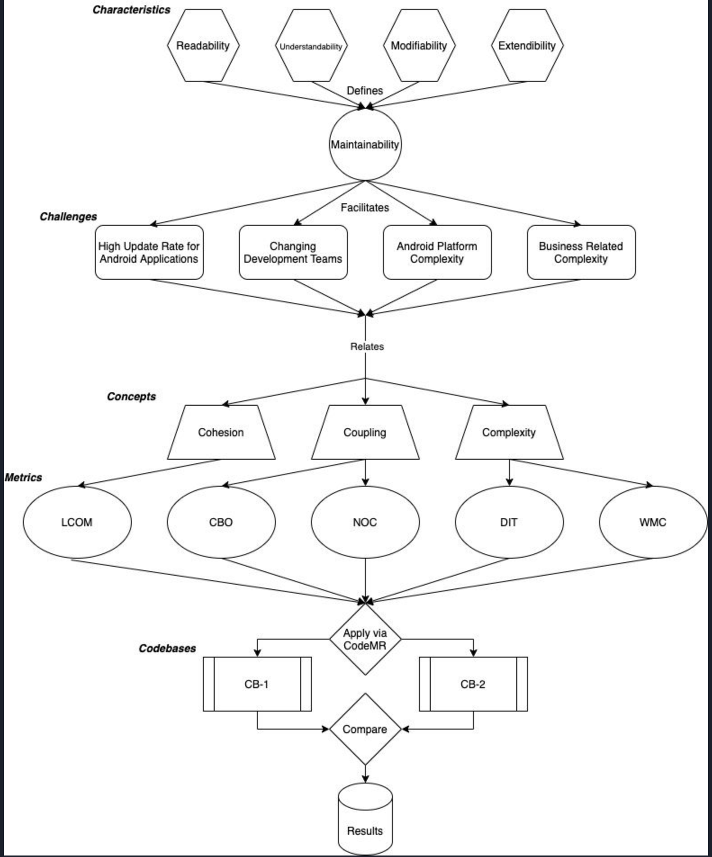
- Facilitating the daily Android development issues for the developers
- Providing time and cost-efficient development process for Android applications

Research Methodology

- The study follows the triangulation strategy.
- This strategy is applied by conducting two different methods.
- These methods are quantitative and qualitative evaluations.


Multi Vocal Literature Review
Provides Answer
RQ1
Android Developer Survey
Interviews with MC Android Team
Maintainability Measurement with OO Metrics
Provides Answer
RQ2

Maintainability Measurement with OO Metrics



Android Developer Survey Results

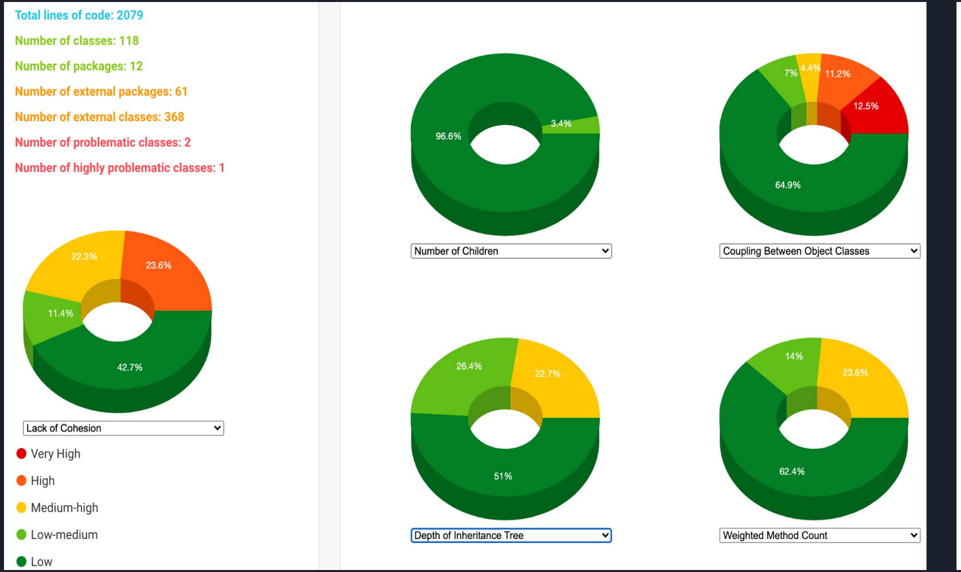
- The survey found 164 participants with different level of expertise.
- Preferences of the case company mostly match the participants' preferences.
- It was observed that a significant number of participants preferred the Kotlin Coroutines solution over the RxJava library.
- Participants mostly accept SOLID principles. Although some participants use Clean Code principles, some participants have been cautiously approached to these principles.
- Developer experience is in correlation with the architecture and design pattern selection.

Interviews with MC Android Team Results

- The participants (7) emphasized the areas of readability, understandability, modifiability and up-to-dateness for the maintainability of software systems.
- Maintainability is even more important for the case company due to the nature of Android apps and the way Mooncascade works.
- Choice of architectural pattern and the stability of the 3rd party libraries used are crucial for the maintainability of Android applications.
- The effects of the methods and technologies they use on maintainability are positive
- The points for improvement are the reassessment of some 3rd party libraries used, such as RxJava, and the improvement of the documentation of Android projects.

Measurement with OO Metrics Results

CB-1



Total lines of code: 2079

Number of classes: 118

Number of packages: 12

Number of external packages: 61

Number of external classes: 368

Number of problematic classes: 2

Number of highly problematic classes: 1

Lack of Cohesion

- ● Very High
- ● High
- ● Medium-high
- ● Low-medium
- ● Low

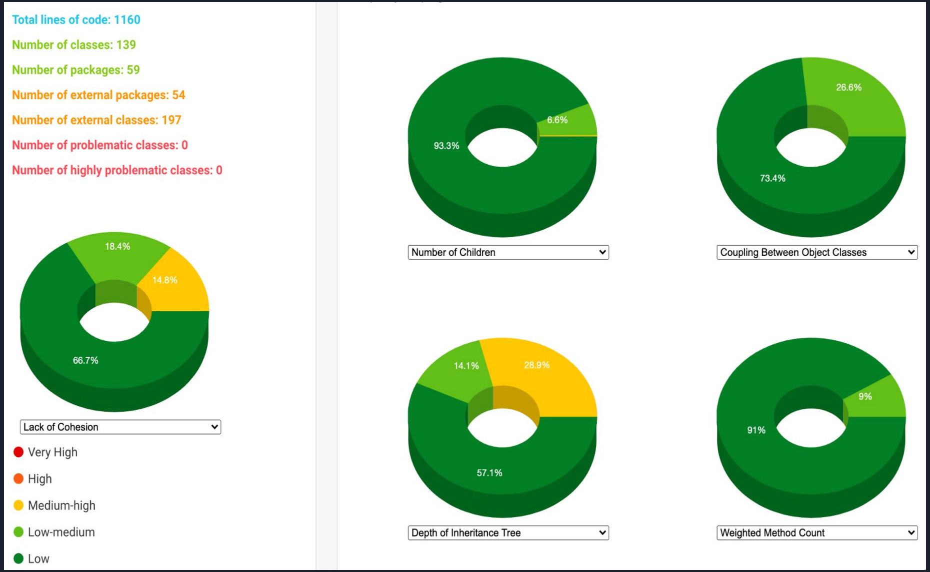
Number of Children

Coupling Between Object Classes

Depth of Inheritance Tree

Weighted Method Count

CB-2



Total lines of code: 1160

Number of classes: 139

Number of packages: 59

Number of external packages: 54

Number of external classes: 197

Number of problematic classes: 0

Number of highly problematic classes: 0

Lack of Cohesion

- ● Very High
- ● High
- ● Medium-high
- ● Low-medium
- ● Low

Number of Children

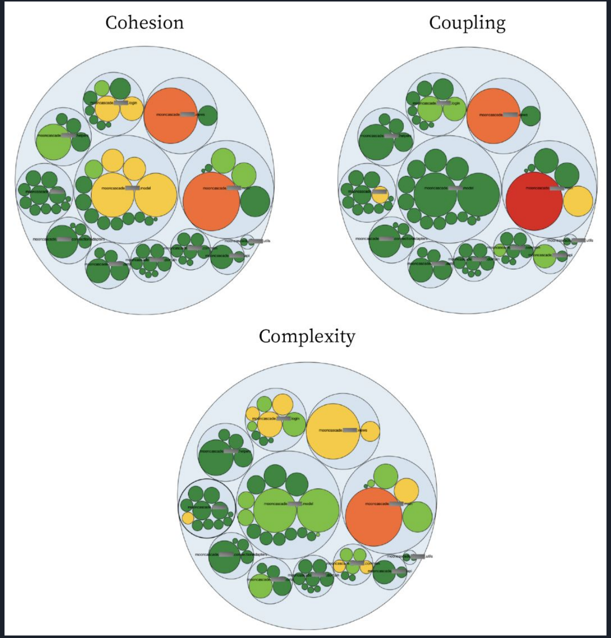
Coupling Between Object Classes

Depth of Inheritance Tree

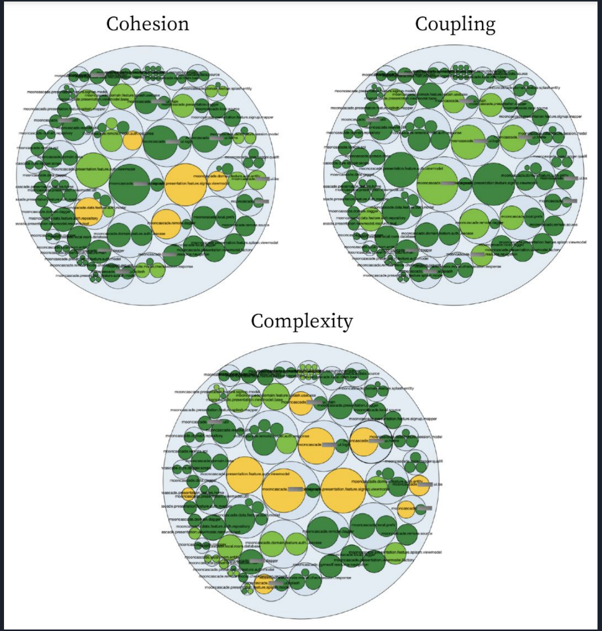
Weighted Method Count

Measurement with OO Metrics Results

CB-1

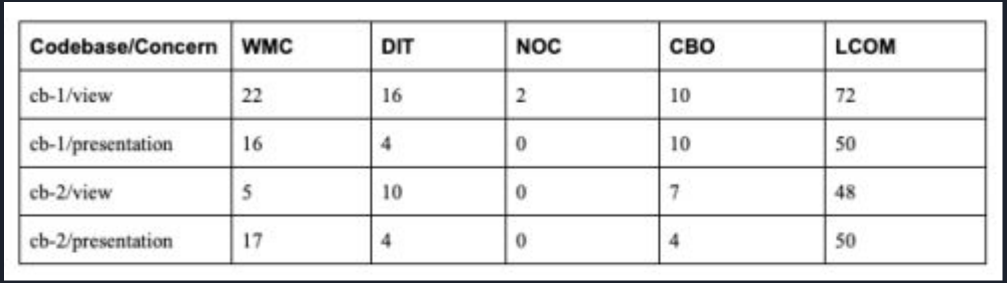


CB-2



Maintainability Measurement with OO Metrics Results

- Login feature metrics result comparison based on presentation and view layers



Codebase/Concern	WMC	DIT	NOC	CBO	LCOM
cb-1/view	22	16	2	10	72
cb-1/presentation	16	4	0	10	50
cb-2/view	5	10	0	7	48
cb-2/presentation	17	4	0	4	50

Limitations

- Features used in quantitative evaluations were relatively less complex features of the Android application. Therefore the efficiency of evaluation and comparison was affected by this situation.
- Choosing the most efficient metrics to measure the maintainability of software systems and Android applications is controversial, especially when the differences of Android applications are taken into account.
- Considering that there may be insufficiencies in the methods used in the evaluations, further research should be conducted to find the most efficient quantitative evaluation method.

Conclusion

THEORETICAL OUTCOME

- Quantitative measurements together with qualitative measures can increase the effectiveness of the study.
- Maintainability of OO software systems can be evaluated quantitatively by using many different metrics.
- Research showed that a new model is needed to measure the maintainability of Android applications. The main reason for this need is the differences of Android applications from traditional software systems and their update rates.

PRACTICAL OUTCOME

- Maintainability of Android applications improved by paying attention to matters such as understandability, implementing human-readable code, proper software architecture and design pattern selection and use of stable third-party libraries.
- The impact of the methods and technologies used by the case company on the maintainability of Android applications is positive even for the relatively simple application features.
- The areas open to improvement are re-evaluating the use of libraries that are in danger of becoming outdated, architectural scaling and selection according to the project, and making the coding conventions more standardized.

Thank you for your attention!

Questions?

Mustafa Ogün Öztürk

mustafaogun.ozturk@gmail.com

Tartu 2021