CMSI 4071: Article Summary Choice Assignment — Extra Credit

Intro:

We chose the paper “Android Apps and User Feedback: A Dataset for Software Evolution and Quality Improvement” by Giovanni Grano, Andrea Di Sorbo, Francesco Mercaldo, Corrado A. Visaggio, Gerardo Canfora, and Sebastiano Panichella. The study is focused on how user reviews and app code data can be combined to help developers improve software quality and maintenance. The authors built a large dataset of open-source Android apps that connects user feedback with actual code versions and quality metrics.

Summary:

The paper presents a dataset of 395 open-source Android applications from the F-Droid repository, which is linked to over 288,000 user reviews from Google Play. Each review is categorized based on its purpose, like bug reports or general feedback, and its topic. The team behind the study collected the app versions and their reviews using web crawlers and automated scripts. They used text-mining tools and classifiers to automatically analyze what users were saying. To connect user opinions to the software itself, the researchers performed a static code analysis on each app version. They used tools like Paprika and custom Python scripts to detect code smells and calculate 22 code quality metrics. The result was a structured dataset that helps researchers explore how user feedback aligns with technical code quality and software evolution.

Key Findings:

- The dataset includes 629 app versions across 23 app categories, ranging from games and tools to social apps.
- The authors found that user reviews often guide app maintenance, as many feedback sentences focus on bugs or updates.
- Most feedback falls into feature/functionality and app-level discussions, which shows that users care most about how the app works and performs.
- Code analysis revealed thousands of issues like Blob Classes and Leaking Inner Classes. That means that poor code structure is still common in Android development.
- The dataset is publicly available on GitHub, allowing other researchers to study relationships between code quality, user satisfaction, and app success.

Reflection:

As a group, we found this paper valuable because it bridges user experience and software engineering. Instead of focusing only on technical improvements or only on user opinions, the authors used both to give developers better insight into what matters most to users. We liked how they automated data collection and labeling using real tools like PhantomJS and Paprika. A practical and scalable way to handle large-scale datasets in app research is important. The idea of linking user feedback to code smells also stood out to us since it could help developers prioritize fixes that have the biggest impact on user satisfaction.