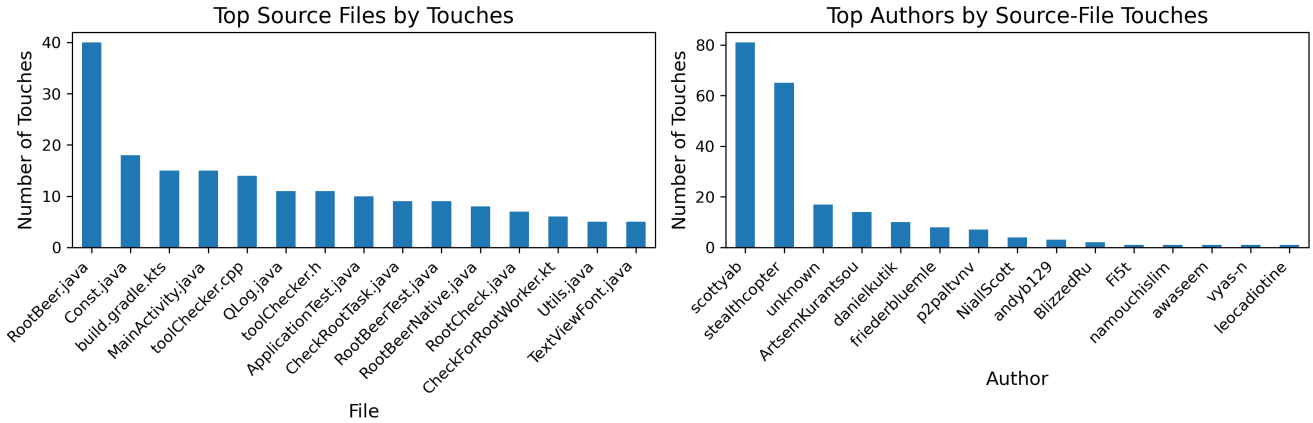
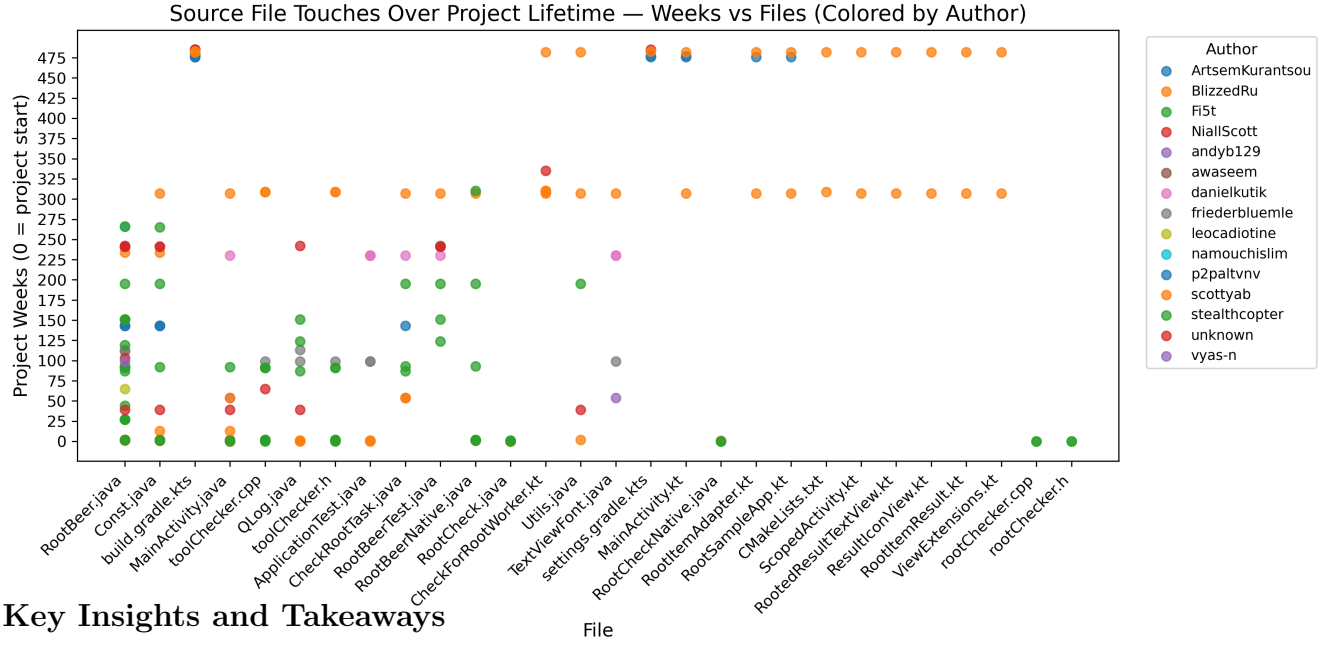
Executive Summary: Repository Maintenance and Ownership Analysis for the RootBeer Project From Project Inception to 2026

Overview This analysis provides a concise, evidence-based view of how the RootBeer codebase is maintained, who is doing the work, and where refactoring and ownership attention should be focused. The assessment is based exclusively on **changes in** the source code extracted from the project's GitHub history. We analyze the complete GitHub commit history and build a dataset that **captures every source code-file, i.e. Java, Kotlin, C, C++ and CMake** which was based on the detected languages of the repository, **the author responsible**, and **the time of each change**. From this data set, three visualizations were generated to highlight the main source files and authors by touches, and the touches of the source file over the lifetime of the project (evolution over time).





Key Insights and Takeaways

Analysis shows that maintenance effort is concentrated in a small core of files, led by `RootBeer.java`, `Const.java`, `build.gradle.kts` and `MainActivity.java`. Most changes to these hotspot files are made by a single primary contributor, `scottyab`, **indicating clear ownership but also a potential key-person dependency risk**. **Time-based analysis** reveals that these core files have evolved continuously from early project weeks through recent activity (weeks ∼450–480), **suggesting accumulated complexity and strong candidates for targeted refactoring**. **Focusing refactoring and knowledge sharing on this small set of files is likely to deliver the greatest improvement in long-term maintainability with minimal disruption**. Lastly, some of the **git commands** used during the analysis include; `git branch, git checkout -b <branchname>, git add, git commit -m "msg", git status, git clone, git push, git log, and others.` All analysis code and outputs are available in the project's branch: https://github.com/sserurich/group-8/tree/mine_repository/repo_mining.