# Motivation

Static analysis plays a prominent role in releasing bug-free software. In spite of that, important tools suffer from well-documented usability issues . Johnson et. al. found design flaws in current static analysis tools and the need for an interactive mechanism in assisting developers in fixing bugs. They conducted interview with 20 participants of which 16 are professional developers and 4 are graduate students. The interesting findings are like if the output of static analysis tool is user friendly and intuitive then false positives and high number of warnings could be less problematic for a developer, showing call hierarchies with which parts of code are affected by a bug, be able to share settings with predefined coding standards among the team, need of a web browser for reacting on the analysis output for instance adding comment to a bug which goes out of context to the developer. Christakis et. al. also did an empirical study on what developers want and need from program analysis. They did a survey by sending invitations to 2,000 developers within their organisation i.e., Microsoft and received 375 responses. The resultant data is analysed and found that there are some obstacles which hinder the usage of a static analysis tool by a developer such as ’Wrong checks are on by default’, ’Too many false positives’, ’Too slow’, ’Complex user interface’ etc. Being a user interface an obstacle for a developer along with other usability problems is noteworthy. The key takes away from the above-mentioned papers is the importance of Usability in the ongoing adaption of static analysis tools.  
  
In general, the setup of most of the recent research done in the area of Static Code Analysis is like assuming a single project in an organisation. Further, they assume there is a single person working on a single project with a single tool tackling a single type of problems. Somehow, the assumptions are made so singular to address a specific issue in their research. However, in practice i.e., in the real world of software engineering, there are numerous people working in teams for multiple projects at a time. Each project uses multiple tools in their software development. Even in the case of Static Code Analysis, multiple tools are used which are each capable of addressing several types of issues in order to find more code flaws .  
  
Habib et. al. did a study on static bug detectors about how many of the bugs are found by the tools. In their paper, they have mentioned that the tools used for their research on the topic are complementing each other in some bug findings and thereby expressed opinion that developers might want to combine the tools and so researchers could address how to reconcile the bug findings reported by multiple tools. This gives additional motivation for this thesis work.