# Future Work

The future work ideas discussed here.

With given scope and time limits of present thesis work, we have picked up important among the research questions formed while brainstorming on the thesis topic theme. So, here are some list of research questions that we consider for future work.

1. Would having tabs help scale the tools visibility with bugs results?
2. Do graphs help in understanding the bugs reported?
3. Will the user need to have graphs for separate tools or one graph combining the tools selected?
4. In scenarios of having unique (single) graph for multiple tools, does user expect to have it shown all the tools integrated by default or need of filters for tools selection?
5. From a code perspective, does showing bugs with short description for different tools with different icons will be scalable?
6. From project files, a file having reported by multiple tools; if a bug of more files, then would it be helpful to show related files in description?
7. Are the choices of feedback for analysis process dependent on the assumption of time by user?
8. Do developers wish to stay until analysis of current bug fix gets done?
9. Will rough time estimation help developer as feedback to take his next decision?
10. How to deal with waiting time for long updates from some tools and short updates for other tools, i.e., responsiveness of static analysis tools?
11. Does having graphs (example: histograms) at particular part of code with commits related bug fixes help the user to trace?
12. Do user need traceability in real-time scenario, i.e., analysis the previous changes or prefer to work right away on current bug fix?
13. How could traceability help the developer in coming up with a solution that satisfies tools reporting same part of code?
14. Does user prefer one-page summary of traces (changes) be better in comparison of multiple short windows showing one at a time be better for user to conclude on decision for a task?
15. How to show the bug notifications which resulted because of change in other part of code?
16. To what level (depth) of tracing, does it suffice for users to keep the state of flow with work?
17. Does traceability feature scale better with increase in usage of tools?
18. How is teamwork facilitated in bug fixing?
19. What feedback works to improve quality of bug fixings?
20. How to minimise in terms of presentation for long updates, i.e., bug findings provided by analysis tools?

As mentioned in limitations chapter about the design tool, it would be good to consider if there is an alternative tool that could help with animation features in order to make it easier to understand than the present approach.