

Survey Questions: Debugging Tools for Static Analysis

Q1 How long have you worked as a software developer?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * < 1 year
 - * 1 – 2 years
 - * 2-5 years
 - * 5 – 10 years
 - * > 10 years
 - * Other...
- Mandatory.

Q2 At the moment, which programming languages do you develop with?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * Java / Android
 - * C / C++
 - * C# / .NET
 - * Perl
 - * JavaScript / TypeScript / NodeJS
 - * PHP
 - * Python
 - * Ruby
 - * Objective C
 - * Other...
- Mandatory.

Q3 Would you be willing to participate in a later interview? If so, please provide your email below.

- Free-text field.

Q4 Which code analysis tools do you use in your current projects?

- Multiple choice question with an “Other...” free-text field.
- Choices:

- * IDE notifications (e.g., dead code in Eclipse)
 - * FindBugs
 - * Fortify
 - * Checkmarx
 - * CodeSonar
 - * Coverity
 - * VeraCode
 - * AppScan
 - * Klocwork
 - * SonarQube
 - * Linters
 - * Other...
- Mandatory.

Q5 At which points of your projects are code analysis tools typically run?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * During coding (in the editor)
 - * During nightly builds
 - * At commit time
 - * At major milestones in the projects
 - * Other...
- Mandatory.

Q6 Who usually configures the analysis tools?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * Yourself
 - * A dedicated team
 - * No one. I use the default settings of the tools.
 - * My manager
 - * Other...
- Mandatory.

Q7 What kind of issues are typically detected by code analysis tools on your projects?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * Security vulnerabilities
 - * Functional bugs
 - * Coding style
 - * Memory consumption

- * Concurrency
- * Performance
- * Other...
- Mandatory.

Q8 In your opinion, what kind of issues *should* code analysis tools detect in your projects?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * Security vulnerabilities
 - * Functional bugs
 - * Coding style
 - * Memory consumption
 - * Concurrency
 - * Performance
 - * Other...
- Mandatory.

Q9 Do you usually review the analysis results yourself?

- Single-answer question.
- Choices:
 - * Yes
 - * No
- Mandatory.

Q10 Where are the analysis results typically reported in your projects?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * In my code editor
 - * In the build output
 - * In the code review
 - * In a dedicated tool
 - * In a pdf report
 - * By email
 - * Other...
- Mandatory.

Q11 Where would you prefer analysis results to be reported?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * In my code editor
 - * In the build output

- * In the code review
- * In a dedicated tool
- * In a pdf report
- * By email
- * Other...
- Mandatory.

Q12 If you are using multiple analysis tools, would you prefer all analysis results to be reported in a single interface or in multiple ones?

- Single-answer question.
- Choices:
 - * One single tool
 - * Multiple tools
 - * I am only using one analysis tool
- Mandatory.

Q13 How long does it usually take you to completely fix an analysis warning?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * A few minutes
 - * < 1 hour
 - * < 1 day
 - * < 1 week
 - * < 1 month
 - * < 6 months
 - * > 6 months
 - * Other...
- Mandatory.

Q14 After you have modified your code in response to an analysis warning, how long would you be willing to wait for the analysis to verify your change?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * < 1 second
 - * < 1 minute
 - * A few minutes
 - * < 1 hour
 - * A few hours
 - * < 1 day
 - * A few days
 - * > 1 week
 - * Other...

- Mandatory.

Q15 Any additional comments you would like to share about reporting tools?

- Free-text field.

Q16 Which analysis tool do you use most?

- Free-text field.
- Mandatory.

Q17 Why do you use this tool?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * Company policy
 - * It helps me code faster
 - * It helps me code better
 - * Other...
- Mandatory.

Q18 How often do you use the tool?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * Once a month or less
 - * 1 – 3 times a week
 - * Once a day
 - * 2 – 5 times a day
 - * > 5 times a day
 - * Other...
- Mandatory.

Q19 When do you usually use the tool?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * In the morning
 - * In the afternoon
 - * In the evening
 - * At night
 - * When I have a few minutes here and there
 - * During the work week
 - * During week-ends
 - * Other...
- Mandatory.

Q20 Where are you when you usually use the tool?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * At work, at my desk
 - * At work, with other colleagues
 - * At work, during meetings
 - * At home
 - * In the transports
 - * Other...
- Mandatory.

Q21 How long per week do you spend interacting with this tool?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * < 1 hour
 - * < 1 – 5 hours
 - * < 5 – 10 hours
 - * < 10 – 30 hours
 - * > 30 hours
 - * Other...
- Mandatory.

Q22 Typically, how long do you use the tool in one working session?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * < 10 minutes
 - * 10 - 30 minutes
 - * > 30 minutes
 - * Hours
 - * Other...
- Mandatory.

Q23 What is your goal when you open the tool?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * To fix all of the warnings it reports
 - * To fix all the warnings I can in the time I have
 - * To fix a set number of warnings
 - * To consult the list of warnings
 - * Other...
- Mandatory.

Q24 Why do you stop using the tool?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * I cannot fix an issue
 - * I have to go to a professional obligation (meeting, etc.)
 - * I am distracted by office events (calls, coffees, etc.)
 - * I am waiting for the tool to update
 - * I finished fixing all issues
 - * Other...
- Mandatory.

Q25 Do you use the default layout of the analysis tool? (i.e., do you change the position of any component in the user interface?)

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * I use the default layout of the tool
 - * I am using a layout customized by the company
 - * I have customized the interface for my own use
 - * I change the interface regularly as I use the tool
 - * Other...
- Mandatory.

Q26 When you open the tool, what is the first thing in the interface that you look for?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * The list of warnings
 - * A dashboard
 - * Some code
 - * Other...
- Mandatory.

Q27 When you close the tool, what is the last thing in the interface that you look at?

- Single-answer question with an “Other...” free-text field.
- Choices:
 - * The list of warnings
 - * A dashboard
 - * Some code
 - * Other...
- Mandatory.

Q28 What are the components in the interface that you use the most?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * The list of warnings
 - * A dashboard
 - * Some code
 - * Other...
- Mandatory.

Q29 Once you finish fixing the issues, does anyone review your fixes?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * A colleague
 - * My manager
 - * A separate team
 - * No one reviews my fixes
 - * Other...
- Mandatory.

Q30 In the perfect analysis tool, how important are the following features to you?

- Multiple-choice grid.
- Categories:
 - * Responsiveness of the analysis (the time taken by the analysis to process my fix)
 - * Responsiveness of the UI (does the User Interface lag?)
 - * Bug information: how a bug works and what it is about
 - * Severity information: how severely the bug can affect your code
 - * Execution information: how the bug can be executed in your code
 - * Interaction information: how the bug is related to other bugs in the code base
 - * Fix information 1: how the bug can be fixed, on a high level
 - * Fix information 2: how the bug can be fixed in your code
 - * Quick fixes: fixes generated by the tools itself
 - * Visualisations: the tools provide you with a detailed visualisation of the warnings in the code
 - * Visualisations: the tools provide you with a general dashboard of the issues
 - * Sorting: the tools allow you to sort through warnings and search for them
 - * Prioritization: the tools allow you to select which bugs to fix first
 - * List: the tools allow you to keep a list of ‘your’ bugs
 - * Tracking progress: the tools give you a clear view of what you have achieved so far

- * Analysis configurations: configurations of the analysis before it runs
- * Feedback in the reporting tool: ability to tell if a warning is correct or not in the reporting tool
- * Customization of the analysis rules: writing my own rules for the analysis
- * Collaboration options: the tools allow you to collaborate with other colleagues to fix issues
- Choices for each categories:
 - * This should not be in an analysis tool
 - * Neutral
 - * Low importance
 - * Important
 - * Very important
 - * Indispensable
- Mandatory.

Q31 Which other features have we missed, and how important are they to you?

- Free-text field.

Q32 In a typical analysis report, what is the usual proportion of warnings that you personally investigate?

- Likert scale from 0 (0%) to 10 (100%).
- Mandatory.

Q33 How do you usually differentiate between a false positive and a real issue?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * False positives occur in particular places in the code I know are never executed
 - * Certain categories of issues are more likely to be false positives
 - * The path shown by the analysis tool is not executable
 - * The conditions along the path are never true
 - * Some constructs of the source code are not well handled by the analysis (e.g., static constructs, loops, etc.)
 - * Other...
- Mandatory.

Q34 How do you select which warnings to investigate first?

- Multiple choice question with an “Other...” free-text field.
- Choices:
 - * I follow the list of warnings from the top down
 - * I look at warnings affecting my code first

- * I prioritize warnings which I know I can fix
 - * I prioritize warnings with the most impact
 - * Other...
 - Mandatory.
- Q35** Among the warnings that you investigate, what is the usual proportion of warnings that you cannot understand / explain?
- Likert scale from 0 (0%) to 10 (100%).
 - Mandatory.
- Q36** What makes such warnings difficult to understand / explain?
- Multiple choice question with an “Other...” free-text field.
 - Choices:
 - * They span over too much of the code base
 - * They are issues that I rarely encounter
 - * I do not understand the explanation given by the analysis tool
 - * I do not understand the code base
 - * Other...
 - Mandatory.
- Q37** What do you typically do with warnings that you cannot understand / explain?
- Multiple choice question with an “Other...” free-text field.
 - Choices:
 - * I ignore them
 - * I suppress them
 - * I mark them and leave them for later
 - * I ask colleagues for help
 - * I escalate them
 - * Other...
 - Mandatory.
- Q38** In a typical analysis report, what is the usual proportion of warnings that you ask others to help you fix?
- Likert scale from 0 (0%) to 10 (100%).
 - Mandatory.
- Q39** What is your main motivation to ask others to help you fix warnings?
- Multiple choice question with an “Other...” free-text field.
 - Choices:
 - * They have more experience with the code base
 - * They have experience in this type of issues
 - * They have more experience with the analysis tool

- * I don't ask others to help me fix warnings
- * Other...
- Mandatory.

Q40 Any additional comments you would like to share about how you review analysis warnings?

- Free-text field.