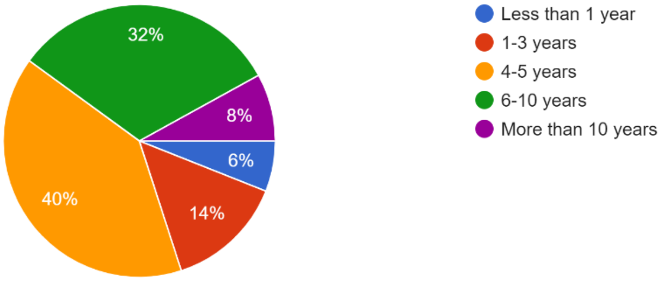
Questionnaire survey statistics results

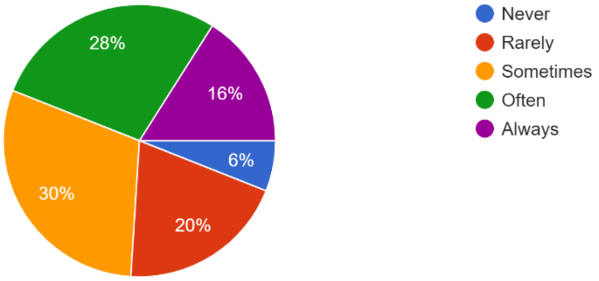
Q1.Years of Experience in Software Development

50 responses



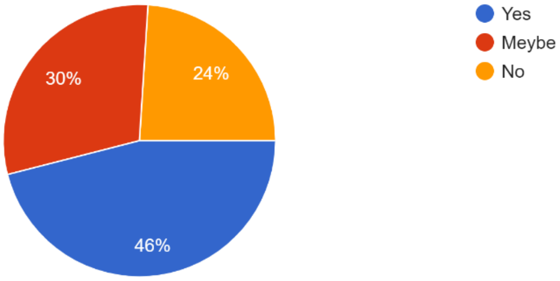
Q2.How often do you consider using Move Method Refactoring in your projects?

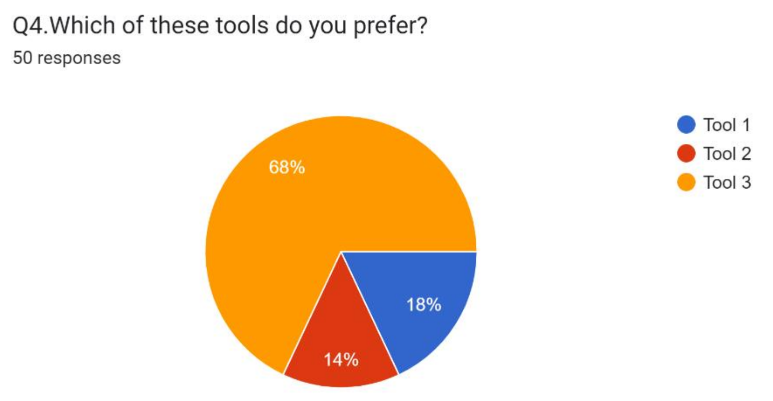
50 responses



Q3.Would you prefer to use tools for detecting Move Method Refactoring opportunities?

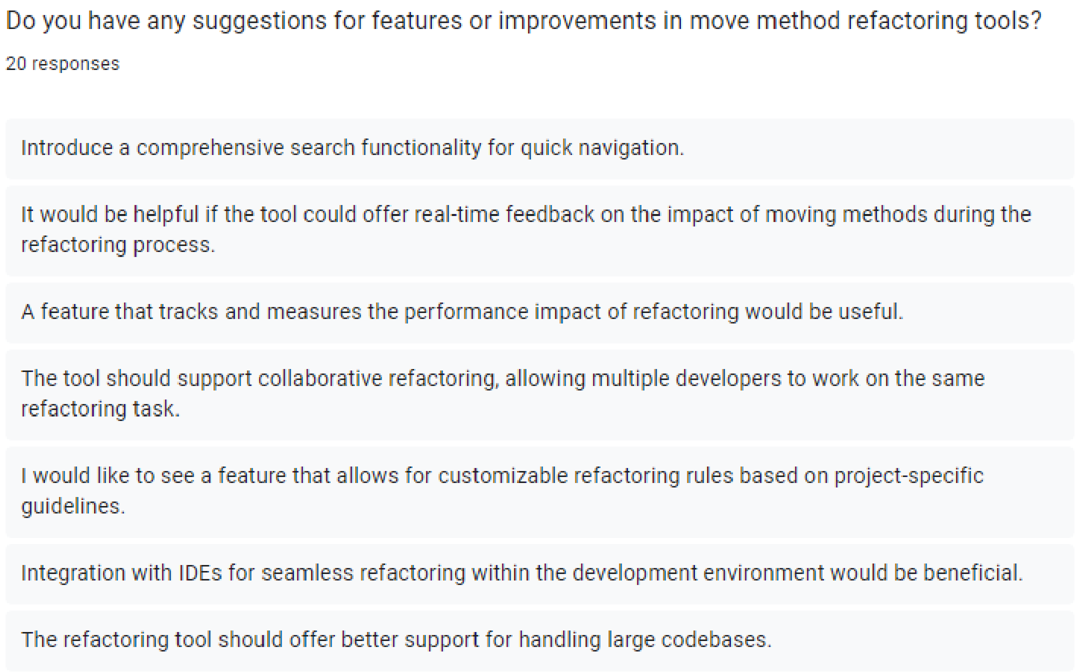
50 responses





Q4.Which of these tools do you prefer?

50 responses



Do you have any suggestions for features or improvements in move method refactoring tools?

20 responses

Introduce a comprehensive search functionality for quick navigation.

It would be helpful if the tool could offer real-time feedback on the impact of moving methods during the refactoring process.

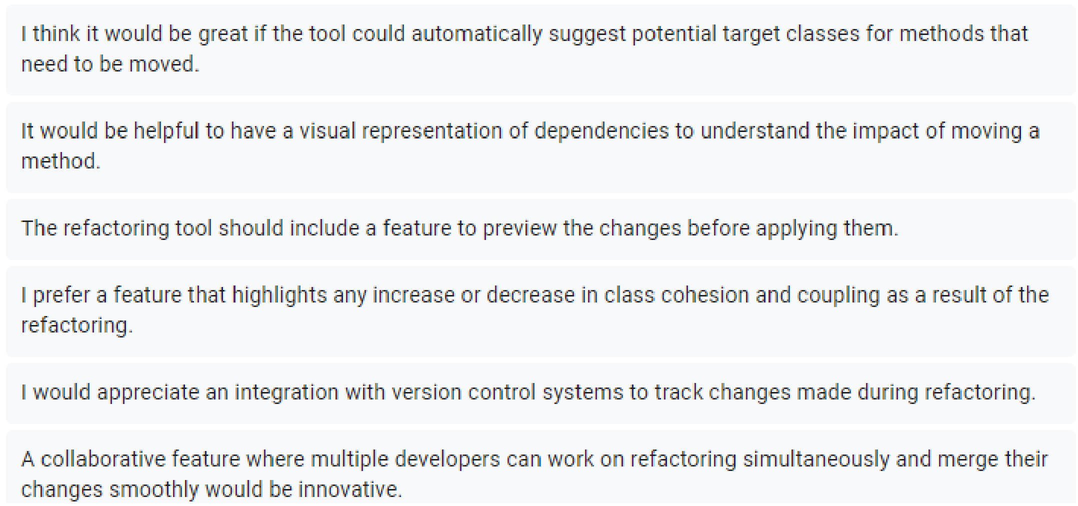
A feature that tracks and measures the performance impact of refactoring would be useful.

The tool should support collaborative refactoring, allowing multiple developers to work on the same refactoring task.

I would like to see a feature that allows for customizable refactoring rules based on project-specific guidelines.

Integration with IDEs for seamless refactoring within the development environment would be beneficial.

The refactoring tool should offer better support for handling large codebases.



I think it would be great if the tool could automatically suggest potential target classes for methods that need to be moved.

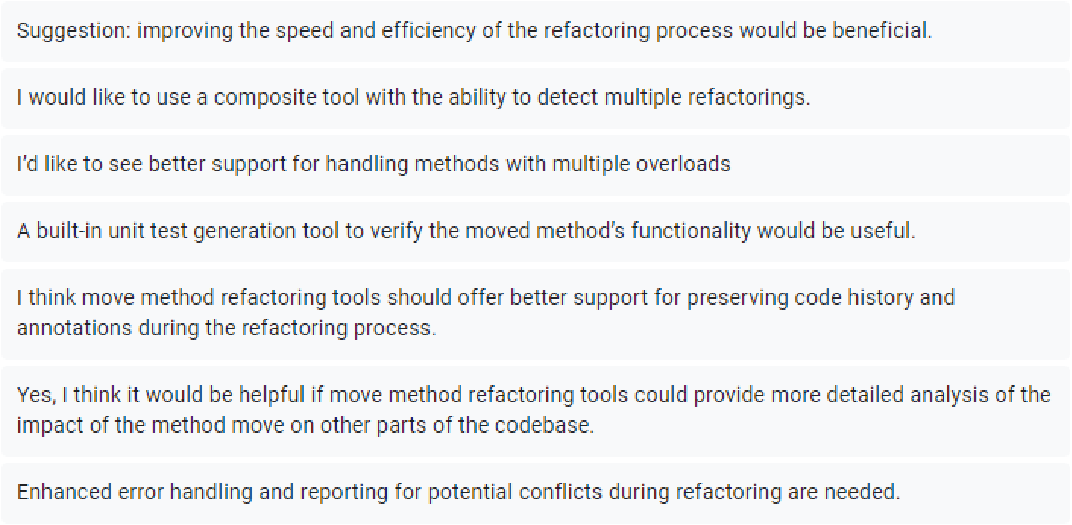
It would be helpful to have a visual representation of dependencies to understand the impact of moving a method.

The refactoring tool should include a feature to preview the changes before applying them.

I prefer a feature that highlights any increase or decrease in class cohesion and coupling as a result of the refactoring.

I would appreciate an integration with version control systems to track changes made during refactoring.

A collaborative feature where multiple developers can work on refactoring simultaneously and merge their changes smoothly would be innovative.



Suggestion: improving the speed and efficiency of the refactoring process would be beneficial.

I would like to use a composite tool with the ability to detect multiple refactorings.

I'd like to see better support for handling methods with multiple overloads

A built-in unit test generation tool to verify the moved method's functionality would be useful.

I think move method refactoring tools should offer better support for preserving code history and annotations during the refactoring process.

Yes, I think it would be helpful if move method refactoring tools could provide more detailed analysis of the impact of the method move on other parts of the codebase.

Enhanced error handling and reporting for potential conflicts during refactoring are needed.