

Reflection Report on AI-Assisted Development using MCP and Copilot

Jesus Teofilo

November 17 2025

Abstract

This report reflects on the experience of integrating AI-assisted tools into software testing and development. It explores coverage improvement patterns, lessons learned, and recommendations for future enhancements.

1 Introduction

AI-assisted development tools can significantly improve productivity and testing efficiency. This report analyzes the application of such tools in a recent software project, highlighting the effects on code coverage, test reliability, and developer workflow.

2 Methodology

The methodology involved:

- Using AI-assisted code generation to implement and enhance unit tests.
- Measuring code coverage before and after AI-assisted testing.
- Reporting issues caused by overwriting of existing tests and files.

3 Results and Analysis

3.1 Coverage Improvement Patterns

AI-assisted testing improved code coverage by automatically generating tests for untested branches. Some patterns observed include:

- Increased coverage in codebase.
- During one test run for example branch coverage went from 89% to 97% and line coverage went from 99% to 100%. Good coverage improvement
- Overwriting of existing tests or duplicating test files can cause temporary regression in testing reliability and can hinder overall progress.

3.2 Lessons Learned

Key lessons include:

- Always backup critical tests and files before running AI-assisted generators.
- AI tools can sometimes be unpredictable and can sometimes be hard to get it to understand what you want exactly
- Debugging AI-generated tests or outputs requires careful understanding and foundational knowledge in software.

3.3 Future Enhancement Recommendations

- I think in the future it would be good if the AI tools could generate like backup folder for files and other things because I encountered that problem where files were overwritten or duplicated in my codebase folder. I really struggled with that while using the AI.
- Reduce overall hallucinations and guessing. Many times the AI doesn't actually know the answer or solution to something and gives misleading information.
- Enhance AI tools so they better interpret context

4 Conclusion

AI-assistance I think can be very powerful if you know what you are doing. I spent a lot more time debugging and fixing things than actually doing the assignment requirements for this project. I am not well versed yet in software testing but I was still able to make a tool that improved coverage for a codebase. But overall the AI was very helpful but since I still don't have good foundational knowledge in this area I had to debug and write better prompts more than if I had more knowledge in this domain. This project highlights the overall state of AI I think. Still needs improvement but is still definitely very helpful.