### **Shahbaj Singh**

### **Journal: Software Testing Techniques**

For the milestones, I employed both black-box and white-box testing techniques, as they aligned with specification-based and structure-based testing approaches. According to Knovel, black-box techniques “derive test cases directly from the specification or from some other kind of model of what the system should do” (Hambling et al., 2015). These methods included equivalence partitioning, used to test valid and invalid inputs; decision tables, applied to evaluate conditions and actions; and state transition testing, which assessed event-driven changes in state or outputs. Additionally, boundary value analysis ensured thorough testing at the limits of input ranges, while use cases were derived directly from test cases.

White-box techniques, also known as structure-based testing, were heavily utilized for coverage analysis and testing conditional logic like *if-then* statements. These techniques break tests into sections, making it easier to analyze individual components. Examples include statement coverage, which checks if all lines of code are executed; path coverage, ensuring all possible paths are traversed; and branch coverage, which evaluates decision points in the code. As Hambling et al. (2015) note, structure-based techniques are particularly useful for exploring system or component structures at various levels.

The testing techniques I did not use for the milestones fall under the category of experience-based methods. These approaches rely on the knowledge and intuition of users and testers to identify critical areas in a system and test them in ways that mimic both expected use and potential abuse. Hambling et al. (2015) describe these techniques as invaluable for identifying areas prone to errors, especially under time constraints or with insufficient specifications.

Specific experience-based techniques include error guessing, where prior knowledge helps pinpoint likely problem areas, and exploratory testing, which focuses on testing components without detailed specifications. I opted not to use these techniques due to my limited experience in testing and the structured nature of the milestone requirements.

Each testing approach has unique applications depending on the project’s needs and context. Black-box techniques are ideal for scenarios where functionality is clearly defined, making them suitable for outsourced testing or systems with robust specifications. White-box techniques, on the other hand, are best used when the tester has a deep understanding of the system’s inner workings and when the project demands thorough validation of code execution paths.

Experience-based techniques are highly practical in situations with incomplete specifications or under tight deadlines. They can uncover issues that formal techniques might miss by leveraging tester intuition and creative testing strategies. Hambling et al. (2015) highlight that these techniques are particularly useful for identifying "special tests" that address gaps in more structured methods.

Overall, the choice of testing techniques depends on the project requirements, team expertise, and available resources, ensuring that the approach aligns with the system’s complexity and testing objectives.