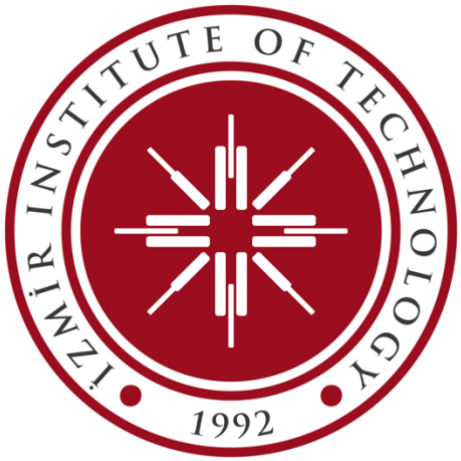

Izmir Institute of Technology

Faculty of Engineering

Department of Computer Engineering

CENG 552 Software Testing Experiment Test Report

Hasan Basri Erdoğan -280201002

Hüseyin Özdemir - 300201106

Mustafa Şahin - 280201026

Table of Contents

Introduction

This report compares three software testing techniques - Decision Table Testing, Equivalence Partitioning, and Boundary Value Analysis - applied to two different projects. Below is an overview of these projects:

Project 1: Shipping Cost Calculation

Description

This project involves a ShippingCostCalculator class designed to compute the shipping cost based on:

1. Purchase Amount: Total cost of items purchased.

2. Number of Items: Quantity of items in the shipment.

3. Delivery Option: Delivery speed/type (e.g., NEXT_DAY, SECOND_DAY, WEEK).

Implementation

The shipping cost is calculated using the following rules:

1. Purchase Amount <= $100:

- ≤ 3 Items:

- NEXT_DAY: Flat rate of $25.

- SECOND_DAY: Flat rate of $10.

- WEEK: $1.50 per item.

- > 3 Items:

- NEXT_DAY: $6 per item.

- SECOND_DAY: $2.50 per item.

- WEEK: Free shipping.

2. Purchase Amount > $100:

- ≤ 3 Items:

- NEXT_DAY: Flat rate of $35.

- SECOND_DAY: Flat rate of $15.

- WEEK: Flat rate of $10.

- > 3 Items:

- NEXT_DAY: $7.50 per item.

- SECOND_DAY: $3.50 per item.

- WEEK: $2.50 per item.

Error Handling
- If purchase amount or number of items is invalid (e.g., ≤ 0), or the delivery option is null, an IllegalArgumentException is thrown.

Project 2: Access Control System

Description
This project implements an AccessSystem class to control access based on:

1. Person Status: User role (e.g., EMPLOYEE, SPECIAL_EMPLOYEE, AUDITOR).

2. Current Date and Time: The specific day and time when access is requested.

Implementation
The access is determined using the following rules:

1. EMPLOYEE:

- Access is allowed only on weekdays during working hours (9:00 AM - 5:00 PM).

2. SPECIAL_EMPLOYEE:

- Always granted access, regardless of time or day (employee with a special code).

3. AUDITOR:

- Access is allowed during working hours (9:00 AM - 5:00 PM), including weekends.

Error Handling
- If the date/time or person status is null, an IllegalArgumentException is thrown.

Boundary Value Analysis (BVA) for Project 1: Shipping Cost Calculation

Inputs Tested

1. Purchase Amount:
- Tested values: Just below $0, $0, $0.01, $50(nom), $99.99, $100, $100.01, and $150(nom).

2. Number of Items:
- Tested values: -1, 0, 1, 2(nom) , 3, 4, and 10(nom).

3. Delivery Options:
- Valid options: NEXT_DAY, SECOND_DAY, WEEK.

Equivalence Class Testing (ECT) for Project 1: Shipping Cost Calculation (Strong)

Input Domains

1. Purchase Amount:
- Valid: 50, 150

- Invalid: -1

2. Number of Items:
- Valid: 2, 5

- Invalid: -1

3. Delivery Option:
- Always valid: NEXT_DAY, SECOND_DAY, WEEK.

Decision Table Testing (DTT) for Project 1: Shipping Cost Calculation

Decision Table

Rule	Purchase Amount	Number of Items	Delivery Option	Expected Outcome
1	≤ 100	≤ 3	NEXT_DAY	Cost = $25
2	≤ 100	≤ 3	SECOND_DAY	Cost = $10
3	≤ 100	≤ 3	WEEK	Cost = Items × 1.5
4	≤ 100	> 3	NEXT_DAY	Cost = Items × 6.0
5	≤ 100	> 3	SECOND_DAY	Cost = Items × 2.5
6	≤ 100	> 3	WEEK	Cost = $0
7	> 100	≤ 3	NEXT_DAY	Cost = $35
8	> 100	≤ 3	SECOND_DAY	Cost = $15
9	> 100	≤ 3	WEEK	Cost = $10
10	> 100	> 3	NEXT_DAY	Cost = Items × 7.5
11	> 100	> 3	SECOND_DAY	Cost = Items × 3.5
12	> 100	> 3	WEEK	Cost = Items × 2.5
13	Invalid (e.g., < 0)	Any	Any	Throws IllegalArgumentException
14	Any	≤ 0	Any	Throws IllegalArgumentException
15	Any	Any	Null	Throws IllegalArgumentException

Comparison of Testing Techniques

Number of Tests

Boundary Value Analysis (BVA): 78 tests

The extensive number of tests underscores the robustness of this approach by thoroughly covering critical transitions and edge cases, such as boundary points for purchase amounts and item counts. While this method excels at capturing fine-grained errors at specific boundaries, it may lead to potential redundancy, as testing similar boundary points does not always provide significant additional value.

Equivalence Partitioning (EP): 27 tests

This approach focuses on representative equivalence classes for each input domain, efficiently minimizing redundant tests by selecting values that represent each class. However, due to the nature of *strong equivalence class testing*, it introduces some redundancy by testing multiple invalid values within the same domain. Strong equivalence class testing emphasizes comprehensive coverage of all possible classes, which can lead to overlap in scenarios, such as testing both -1 and 0 for purchase amount or item count. While this provides balanced coverage by addressing key cases, it could be further optimized by testing just one representative invalid value per domain to reduce unnecessary redundancy.

Decision Table Testing (DTT): 15 tests

This approach tests all logical combinations of inputs to ensure that no critical rule is missed in the decision-making process. It is both compact and comprehensive, as it concentrates on well-defined decision rules, making it highly effective for scenarios where logical correctness is crucial. However, its primary weakness lies in its lack of focus on edge cases, which means it may not be as effective at identifying boundary-specific errors. While it ensures thorough testing of logical rules, additional measures might be needed to handle edge cases more robustly and improve the overall reliability of the system.

Execution Time Per Test

Boundary Value Analysis (BVA):
- Total Time: 0.015 seconds

- Number of Tests: 78

- Execution Time per Test: 0.000192 seconds

Decision Table Testing (DTT):
- Total Time: 0.008 seconds

- Number of Tests: 15

- Execution Time per Test: 0.000533 seconds

Equivalence Class Testing (ECT):
- Total Time: 0.007 seconds

- Number of Tests: 27

- Execution Time per Test: 0.000259 seconds

Insights
- BVA is the most efficient with the fastest execution time per test (0.000192 seconds), despite having the highest number of tests.

- ECT performs moderately with an execution time of 0.000259 seconds per test.

- DTT is the slowest per test (0.000533 seconds), though it has the smallest total number of tests.

Test Generation Time

1. Boundary Value Testing (BVT): 3 minutes 45 seconds

2. Equivalence Class Testing (ECT): *3 minute 21 seconds*

3. Decision Table Testing (DTT): 7 minutes 15 seconds

ECT, at 3 minutes and 21 seconds, is the fastest due to its use of representative cases in strong equivalence class testing. BVT, slightly longer at 3 minutes and 45 seconds, focuses on detailed boundary conditions. DTT, taking 7 minutes and 15 seconds, is the most time-intensive, ensuring comprehensive rule coverage. ECT suits quick checks, BVT targets edge cases, and DTT excels in thorough validations.

Coverage Analysis

1. Boundary Value Analysis (BVA)
- Coverage Achieved: 95.6%

Boundary Value Analysis (BVA) excels at testing edge cases like purchase amount and item count but falls short in covering all logical rules. For example, it does not explicitly test null values for the WEEK delivery option. While its strength lies in robust boundary testing, it requires additional measures to address gaps like untested logical scenarios.

2. Equivalence Class Testing (ECT)
- Coverage Achieved: 95.6%

Equivalence Class Testing (ECT) is efficient as it tests representative cases for both valid and invalid equivalence classes, avoiding redundancies. However, like BVA, it fails to test null delivery options for the WEEK case, resulting in a slight gap in statement coverage. Its strength lies in focusing on efficient and representative testing, but it requires additional checks to ensure complete coverage of all scenarios.

3. Decision Table Testing (DTT)
- Coverage Achieved: 100%

Decision Table Testing (DTT) ensures complete coverage of all logical combinations by explicitly defining all rules and their outcomes, including null values for all delivery options. This thoroughness guarantees that no rule is overlooked, providing a comprehensive testing approach. However, this comes at the expense of testing fewer edge cases, as the focus remains on covering all possible logical scenarios rather than boundary-specific conditions.

Insights
- BVA and ECT are excellent for testing edge cases and equivalence classes, but they may miss certain logical checks if not explicitly defined (like null values).

- DTT ensures all logical rules are verified, making it more reliable for 100% rule coverage but potentially less exhaustive for edge scenarios.

Mutation Testing Results Analysis

1. Boundary Value Analysis (BVA)
- Line Coverage: 43/44 (98%)

- Mutations Generated: 18

- Mutations Killed: 17

- Mutation Coverage: 94%

- Mutations with No Coverage: 0

- Test Strength: 94%

- Tests Run: 77 (4.28 tests per mutation)

Boundary Value Analysis (BVA) demonstrates high mutation coverage and strong test strength, highlighting its robustness in detecting subtle edge-case errors. However, it shows slightly lower line

coverage, which may be attributed to untested scenarios such as null values, leaving minor gaps in the overall testing scope.

2. Decision Table Testing (DTT)
- Line Coverage: 44/44 (100%)

- Mutations Generated: 18

- Mutations Killed: 16

- Mutation Coverage: 89%

- Mutations with No Coverage: 0

- Test Strength: 89%

- Tests Run: 48 (2.67 tests per mutation)

Decision Table Testing (DTT) achieves 100%-line coverage, ensuring that all rules are thoroughly tested and accounted for. However, it has slightly lower mutation coverage and test strength compared to BVA, as it focuses less on edge cases and more on comprehensive rule-based testing. Also, efficient with fewer tests per mutation.

3. Equivalence Class Testing (ECT)
- Line Coverage: 43/44 (98%)

- Mutations Generated: 18

- Mutations Killed: 13

- Mutation Coverage: 72%

- Mutations with No Coverage: 0

- Test Strength: 72%

- Tests Run: 83 (4.61 tests per mutation)

Equivalence Class Testing (ECT) shows the lowest mutation coverage and test strength among the techniques, indicating gaps in detecting subtle errors due to fewer cases explicitly targeting those mutations. Additionally, its higher test-per-mutation ratio suggests some redundancy in the tests, which could be optimized to enhance efficiency and effectiveness.---

Insights

- BVA performs best in mutation coverage and test strength, highlighting its robustness in detecting edge-case errors.

- DTT achieves 100% line coverage and is efficient, with fewer tests per mutation. It ensures rule completeness but sacrifices mutation coverage slightly.

- ECT struggles with mutation coverage (72%) despite high test-per-mutation count, indicating potential redundancy and lack of robustness in catching subtle issues.

Project 2: Access System

Description

This project involves an AccessSystem class designed to determine whether a person is granted access to a facility based on:

1. Person Status: The role of the person requesting access (e.g., EMPLOYEE, SPECIAL_EMPLOYEE, AUDITOR).

2. Current Date and Time: The date and time of the access request.

Implementation

Access is determined using the following rules:

1. EMPLOYEE:

- Access is granted only on weekdays (Monday to Friday).

- The access request must be within working hours (09:00–17:00).

2. SPECIAL_EMPLOYEE:

- Access is granted at any time, regardless of the day or time.

3. AUDITOR:

- Access is granted within working hours (09:00–17:00) on any day, including weekends.

Error Handling

-If `currentDateTime` is null, an IllegalArgumentException is thrown.

-If `personStatus` is null, an IllegalArgumentException is thrown.

Boundary Value Testing (BVT) for Project 2: Access System

Inputs Tested

1. Person Status:
- Tested values: `EMPLOYEE`, `SPECIAL_EMPLOYEE`, `AUDITOR`.

2. Day Type:
- Tested values: `WEEKDAY`, `WEEKEND`.

3. Time:
- Tested boundary values around working hours:

- 08:59 (just before working hours).

- 09:00 (start of working hours).

- 09:01 (just after start).

- 16:59 (just before end of working hours).

- 17:00 (end of working hours).

- 17:01 (just after working hours).

Equivalence Class Testing (ECT) for Project 2: Access System

Input Domains

1. Person Status:
- Valid values: `EMPLOYEE`, `SPECIAL_EMPLOYEE`, `AUDITOR`.

- Different roles determine access rules.

2. Date and Time:
- Weekday Work Hours: Represented by Monday at 10:00 (within valid working hours).

- Weekday Non-Work Hours: Represented by Monday at 18:00 (outside working hours).

- Weekend Work Hours: Represented by Saturday at 10:00 (valid for auditors).

- Weekend Non-Work Hours: Represented by Saturday at 18:00 (not valid for any except special employees).

Test Cases

1. Employee on a weekday during work hours:

- Representative class: `Valid Person Status` + `Weekday Work Hours`.

- Result: Access allowed.

2. Special Employee on a weekday outside work hours:

- Representative class: `Valid Person Status` + `Weekday Non-Work Hours`.

- Result: Access allowed.

3. Auditor on a weekend during work hours:

- Representative class: `Valid Person Status` + `Weekend Work Hours`.

- Result: Access allowed.

4. Employee on a weekend outside work hours:

- Representative class: `Valid Person Status` + `Weekend Non-Work Hours`.

- Result: Access denied.

Decision Table Testing (DTT) for Project 2: Access System

Decision Table

Rule	Person Status	Day Type	Time	Expected Outcome
1	Null	Any	Any	Throws IllegalArgumentException
2	Any	Null	Null	Throws IllegalArgumentException
3	EMPLOYEE	Weekday	Work Hours	Access **Granted**
4	EMPLOYEE	Weekday	Non-Work Hours	Access **Denied**
5	EMPLOYEE	Weekend	Any	Access **Denied**
6	SPECIAL_EMPLOYEE	Any	Any	Access **Granted**
7	AUDITOR	Weekday	Work Hours	Access **Granted**
8	AUDITOR	Weekday	Non-Work Hours	Access **Denied**
9	AUDITOR	Weekend	Work Hours	Access **Granted**
10	AUDITOR	Weekend	Non-Work Hours	Access **Denied**

Note:
Although `day type` and `hour` are combined as a single input in the implementation, they are tested separately to check each part independently and more easily.

Comparison of Testing Techniques for Project 2: Access System

Number of Tests

Boundary Value Testing (BVT): 36 tests

Focuses on edge cases, ensuring that transitions between valid and invalid inputs are thoroughly tested. With 36 tests, it provides the highest coverage of critical boundaries, such as time values just before, during, and after working hours, and different roles across weekdays and weekends. This granularity makes it highly effective in detecting subtle errors but also introduces redundancy, as similar boundary points often yield comparable results.

Equivalence Class Testing (ECT): 4 tests

Adopts a concise approach by representing broader input ranges with weak equivalence classes, resulting in only 4 tests. This method is efficient in capturing key functionality with minimal effort but does not focus on transitions or edge-specific errors, potentially missing boundary conditions. The reduced test count reflects its focus on general cases rather than exhaustive validation.

Decision Table Testing (DTT): 10 tests

Ensures logical completeness by testing all rule combinations, covering every possible interaction between roles, days, and times. With 10 tests, it provides a balance between coverage and efficiency, explicitly validating each rule while avoiding redundancy. However, it lacks the depth to capture errors specific to boundaries, as its focus is on rule validation rather than edge conditions.

Insights:
The test count differences highlight the focus of each method. BVT's detailed boundary validation leads to the highest number of tests, while ECT minimizes redundancy with only 4 tests by using weak equivalence class testing. DTT achieves logical completeness with 10 tests, balancing efficiency and coverage. The choice of method depends on the priorities of the system—BVT for edge case robustness, ECT for efficient validation of general cases, and DTT for comprehensive rule-based testing.

Execution Time Per Test

1. Boundary Value Testing (BVT):
- Execution Time per Test: 0.00144 seconds

- Boundary Value Testing (BVT) is ideal for time-sensitive systems requiring frequent validation of edge cases, as it achieves the fastest execution per test due to the granularity of boundary conditions. While the higher number of tests ensures detailed validation and robustness, it slightly increases the overall execution time, making it a strong choice for systems prioritizing edge case precision.

2. Equivalence Class Testing (ECT):
- Execution Time per Test: 0.0125 seconds

- Equivalence Class Testing (ECT) has the highest execution time per test due to its minimal number of tests focusing on broader equivalence classes. This makes it more efficient for small-scale systems or scenarios where resource constraints are critical. However, its focus on general cases makes it less effective at catching boundary-specific issues, limiting its applicability for systems requiring detailed edge case testing.

3. Decision Table Testing (DTT):
- Execution Time per Test: 0.0046 seconds

- Decision Table Testing (DTT) strikes a balance between execution time and coverage, offering fewer tests than BVT while ensuring logical completeness for comprehensive validation. Its moderate execution time per test makes it the most balanced approach, particularly suited for systems with diverse conditions, such as access control, where both efficiency and thorough rule coverage are essential.

Test Generation Time

1. Boundary Value Testing (BVT): 2 minutes 26 seconds

2. Equivalence Class Testing (ECT): *1 minute 10 seconds*

3. Decision Table Testing (DTT): 4 minutes 15 seconds

The test generation time reflects the focus of each technique. ECT, at 1 minute and 10 seconds, is the fastest due to its use of weak equivalence class testing, which minimizes test cases by focusing on broad input ranges. BVT takes 2 minutes and 26 seconds, balancing detail and efficiency by testing critical boundary transitions. DTT requires the longest time, 4 minutes and 15 seconds, as it involves creating a comprehensive decision table to cover all logical rules and combinations. The efficiency of ECT suits quick validations, while the thoroughness of BVT and DTT is ideal for detailed, robust testing.

Coverage Analysis

1. Boundary Value Testing (BVT)
- Coverage Achieved: 84.9%

Boundary Value Testing (BVT) focuses on edge cases such as time boundaries, weekday/weekend distinctions, and role-specific conditions. However, it lacks explicit coverage for null inputs or unexpected combinations, leaving some gaps in validation. Its main strength is its ability to provide robust testing for boundary-specific errors, especially in transitions between valid and invalid access times, making it highly effective for systems where edge case precision is critical.

2. Equivalence Class Testing (ECT)
- Coverage Achieved: 80.2%

Equivalence Class Testing (ECT) efficiently tests representative cases across equivalence classes, providing compact and effective coverage for core functional scenarios. However, it does not address boundary-specific conditions or null inputs, leaving potential gaps in edge case validation. Its strength lies in its ability to perform efficient and focused testing for general cases while minimizing redundancy.

3. Decision Table Testing (DTT)
- Coverage Achieved: 91.9%

Decision Table Testing (DTT) ensures complete rule coverage by including edge conditions and handling null inputs. Its strength lies in guaranteeing that no rule or logical condition is missed, making it a highly comprehensive approach for thorough logical validation.

Insights

- BVT focuses on providing detailed insights into edge conditions, making it highly effective for identifying time-specific boundary issues in access systems. However, it lacks logical completeness as it does not validate all possible rules, which may leave gaps in coverage for complex logical scenarios.- Ideal for identifying time-specific boundary issues in access systems.

- ECT is efficient in representing key cases by minimizing redundant tests and focusing on core functional scenarios. While it works well for testing simple systems or general cases, it falls short in providing robust validation for systems with complex conditions or exhaustive requirements.

- DTT Offers the most balanced and complete coverage, making it ideal for validating access rules in systems with multiple conditions.

Mutation Testing Results Analysis

1. Boundary Value Testing (BVT)

- Line Coverage: 25/27 (93%)

- Mutations Generated: 13

- Mutations Killed: 13

- Mutation Coverage: 100%

- Mutations with No Coverage: 0

- Test Strength: 100%

- Tests Run: 19 (1.46 tests per mutation)

This approach achieves perfect mutation coverage and test strength, demonstrating exceptional robustness in identifying errors. However, slightly lower line coverage indicates that some specific execution paths may not be fully explored, leaving minor gaps in overall testing completeness.

2. Equivalence Class Testing (ECT)

- Line Coverage: 25/27 (93%)

- Mutations Generated: 13

- Mutations Killed: 11

- Mutation Coverage: 85%

- Mutations with No Coverage: 1

- Test Strength: 92%

- Tests Run: 15 (1.15 tests per mutation)

This approach maintains reasonable mutation coverage but shows a slight weakness in detecting all mutations. It is efficient in terms of tests-per-mutation but sacrifices some test strength compared to BVT. Additionally, it misses subtle logical errors that are more effectively captured by other techniques, highlighting a trade-off between efficiency and thoroughness.

3. Decision Table Testing (DTT)

- Line Coverage: 27/27 (100%)

- Mutations Generated: 13

- Mutations Killed: 12

- Mutation Coverage: 92%

- Mutations with No Coverage: 1

- Test Strength: 100%

- Tests Run: 23 (1.77 tests per mutation)

This approach achieves perfect line coverage, ensuring all possible rules and execution paths are thoroughly tested. Its high mutation coverage and test strength make it highly effective for logical completeness. However, a slightly higher tests-per-mutation ratio compared to ECT indicates a more comprehensive validation process, prioritizing thoroughness over minimalism.

Insights

- BVT stands out with 100% mutation coverage and test strength, highlighting its robustness in identifying subtle edge-case errors.

- ECT is efficient in terms of tests-per-mutation but falls behind in mutation coverage (85%), indicating a potential gap in detecting deeper logical flaws.

- DTT achieves perfect line coverage and high mutation coverage, ensuring thorough rule validation and logical completeness.

Conclusion

In this analysis, we evaluated three testing techniques - Boundary Value Testing (BVT), Equivalence Class Testing (ECT), and Decision Table Testing (DTT) - across two distinct projects: the Shipping Cost Calculator and the Access Control System. Each testing approach demonstrated unique strengths and weaknesses, providing insights into their applicability and effectiveness.

Boundary Value Testing (BVT) was highly effective in detecting edge-case errors by focusing on critical transitions and boundary values. For both projects, BVT achieved high mutation coverage and test strength, making it robust and reliable. However, BVT introduced redundancy in some scenarios, such as testing multiple invalid boundary points, which added little value to the results. Despite its higher test count, BVT maintained a fast execution time per test, demonstrating its efficiency for boundary-sensitive systems.

Equivalence Class Testing (ECT), which utilized strong equivalence classes in the Shipping Cost Calculator (Project 1) and weak equivalence classes in the Access Control System (Project 2), highlighted different testing strategies. Strong equivalence in Project 1 ensured more comprehensive coverage of input domains, capturing nuanced scenarios effectively. However, in Project 2, weak equivalence aimed to reduce redundancy by selecting minimal representative cases, leading to a leaner test suite. While this approach efficiently covered key cases, it struggled to detect subtle errors and missed boundary-specific issues, as evidenced in the mutation testing results.

Decision Table Testing (DTT) emerged as the most logically complete approach across both projects, achieving close to full line coverage in both cases. By systematically testing all possible combinations of inputs and their expected outcomes, DTT ensured no logical rule or condition was missed. This made it particularly suitable for systems with intricate decision-making rules, such as the Access Control System. However, DTT's focus on logical completeness limited its ability to address edge cases, as it lacked the granularity offered by BVT.

In terms of execution time, BVT was the fastest per test despite having the highest test count. ECT, with its smaller number of tests, was efficient in scenarios with limited complexity but had the slowest execution time per test in Project 2. DTT balanced coverage and execution time, providing comprehensive validation at a moderate cost in execution time per test.

Mutation testing results revealed that BVT demonstrated the highest robustness in detecting subtle flaws, achieving perfect mutation coverage and test strength in Project 2 and near-perfect results in Project 1. DTT also performed well, ensuring high mutation coverage and logical completeness, though

its focus on logical rules left some gaps in edge-case detection. ECT lagged behind in mutation coverage and test strength, particularly in Project 2, where weak equivalence testing limited its ability to uncover deeper logical flaws.

General Insights

1. Testing Strategies:

- Project 1 (Shipping Cost Calculator) benefited from strong equivalence classes, ensuring thorough validation of representative cases across input domains.

- Project 2 (Access Control System) relied on weak equivalence classes, reducing redundancy but introducing gaps in edge-case coverage.

BVT is ideal for systems with critical boundary conditions, such as shipping cost calculations or time-sensitive access control, due to its detailed edge-case testing and robust mutation coverage.

Weak ECT provides a minimalistic and efficient approach for systems with fewer input domains or simpler conditions. However, its reliance on weak equivalence (as in Project 2) can lead to gaps in boundary and logical rule validation.

Strong ECT, while providing comprehensive validation by thoroughly testing all equivalence classes, can introduce unnecessary cases. By covering multiple invalid or redundant scenarios within the same domain, it adds extra tests that do not significantly enhance coverage. This can lead to inefficiencies, especially in systems where fewer but representative tests could achieve similar results without sacrificing accuracy.

DTT excels in ensuring logical completeness and covering all possible decision rules, making it the most reliable technique for systems with complex logic and multiple conditions, such as access control rules.

Final Recommendations

A hybrid testing approach that combines the strengths of each technique would provide the most comprehensive validation:

- Use BVT for detecting critical boundary errors.

- Leverage strong equivalence classes (ECT) for comprehensive coverage of representative cases in systems with wide input domains.

- Apply DTT for systems with complex decision-making rules to ensure logical completeness and rule validation.

This approach would maximize coverage and robustness while optimizing test efficiency for both edge-sensitive and logic-heavy systems.

Used Tools

-EclEmma Plugin for coverage analysis

-Pitclipse for mutation analysis

-Eclipse as main development IDE