

Test technique: Equivalence Partitioning (EP)



สาขาระบบสารสนเทศทางธุรกิจ มหาวิทยาลัยราชภัฏจันทรเกษม



www.chandra.ac.th



AGENDA

- What is Equivalence Partitioning ?
- How it Works
- Example 1
- Benefits
- Summary
- Example 2-4

ผู้ช่วยศาสตราจารย์ ดร.สุรชาติ บัวชุม

Equivalence Partitioning (EP)

Equivalence Partitioning (EP) is a software testing technique that divides input data into equivalent partitions or classes. These classes are designed to represent sets of inputs that are expected to be treated similarly by the system under test.

What is Equivalence Partitioning ?

What is Equivalence Partitioning ?

- **Definition:** Equivalence Partitioning is a method of dividing the input data of a software unit into partitions of equivalent data from which test cases can be derived.
- **Purpose:** The main goal is to reduce the number of test cases to a manageable level while still maintaining reasonable test coverage.

Equivalence Partitioning (EP): How it Works

How it Works

- **Identify Input Data:** Determine the inputs to the system or component.
- **Create Partitions:** Divide these inputs into groups (partitions) where the system is expected to behave similarly for all inputs in a group.
- **Select Representatives:** Choose a representative value from each partition to use as a test case.

Equivalence Partitioning (EP): Example 1

Example 1

Consider a function that accepts an integer input between 1 and 100:

- **Valid Partition: 1-100**
- **Invalid Partitions: less than 1, greater than 100, non-integer values**

For testing, you might select:

- **Valid: 50 (middle of the valid range)**
- **Invalid: 0 (less than the minimum), 101 (greater than the maximum), "abc" (non-integer)**

Equivalence Partitioning (EP): Benefits

Benefits

- **Efficiency:** Reduces the number of test cases needed.
- **Effectiveness:** Ensures coverage of different input scenarios.
- **Simplicity:** Easier to identify and create test cases.

Equivalence Partitioning (EP): Summary

Summary

Equivalence Partitioning helps testers focus on significant test cases that are more likely to uncover defects, ensuring efficient and effective testing without the need to test every possible input.

Example 2: การทดสอบฟังก์ชันรับคะแนนนักเรียน

ความต้องการ:
ฟังก์ชันจะรับค่า
คะแนนของนักเรียน
เป็นตัวเลขระหว่าง 0
ถึง 100 และจะแบ่ง
ผลลัพธ์ออกเป็น
เกรด A, B, C, D, และ
F ตามเกณฑ์ที่
กำหนด

Equivalence Partitions:

Valid Partitions:

- 0-49: Grade F
- 50-59: Grade D
- 60-69: Grade C
- 70-79: Grade B
- 80-100: Grade A

Invalid Partitions:

- Less than 0
- Greater than 100
- Non-integer values

ตัวอย่างการทดสอบ:

- Valid: 25 (F), 55 (D), 65 (C), 75 (B), 85 (A)
- Invalid: -5 (น้อยกว่า 0), 105 (มากกว่า 100), "abc" (ไม่ใช่ตัวเลข)

Example 3: การทดสอบระบบการจองห้องพักโรงแรม

ความต้องการ:
ระบบการจองห้องพักจะรับค่าจำนวนคืนที่ต้องการจอง และวันที่เริ่มต้น และจะยอมรับจำนวนคืนตั้งแต่ 1 ถึง 30 คืนเท่านั้น

Equivalence Partitions:
Valid Partitions:

- 1-30 คืน

Invalid Partitions:

- Less than 1 คืน
- Greater than 30 คืน
- Non-integer values

ตัวอย่างการทดสอบ:

- Valid: 5 คืน, 15 คืน, 30 คืน
- Invalid: 0 คืน (น้อยกว่า 1), 31 คืน (มากกว่า 30), "three" (ไม่ใช่ตัวเลข)

Example 4: การทดสอบระบบสมัครสมาชิกออนไลน์

**ความต้องการ:
ระบบสมัครสมาชิก
ออนไลน์จะรับค่า
รหัสผ่านที่มีความ
ยาวระหว่าง 8 ถึง
16 ตัวอักษร**

Equivalence Partitions:

Valid Partitions:

- 8-16 characters

Invalid Partitions:

- Less than 8 characters
- Greater than 16 characters
- Non-alphanumeric characters (ถ้ามีข้อกำหนดเรื่องตัวอักษรที่ใช้)

ตัวอย่างการทดสอบ:

- Valid: "password", "Passw0rd1", "ValidPass16"
- Invalid: "short" (น้อยกว่า 8 ตัวอักษร), "ThisIsAVeryLongPassword" (มากกว่า 16 ตัวอักษร), "Invalid#Pass" (มีตัวอักษรพิเศษถ้ามีข้อกำหนดเรื่องตัวอักษรที่ใช้)

Q&A

Take home: Make the User Stories