Software Testing (UE18CS400SB)
Unit 3

Aronya Baksy

November 2021

1 Black-Box Testing

- Done with only functional knowledge of system, and no knowledge of system internals
- Characteristics:
  - Done based on requirements. Should address stated and implied requirements
  - Encompass the end-user perspective
  - Check for valid conditions/inputs
- Black-box testing methodologies should be chosen so that likelihood of uncovering **new** defects is **maximized**

1.1 Advantages of Black-Box Testing

- Well suited and efficient for large code segments.
- Code access is not required.
- Clearly separates user's perspective from the developer's perspective through visibly defined roles.
- Large numbers of moderately skilled testers can test the application with no knowledge of system internals

1.2 Disadvantages of Black-Box Testing

- Limited coverage, since only a selected number of test scenarios is actually performed.
- Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
- Blind coverage, since the tester cannot target specific code segments or error-prone areas.
- The test cases are difficult to design

2 Specification-Based Testing

- A specification can be anything like a written document, collection of use cases, a set of models or a prototype
- Specification-based testing is applicable at all levels (unit to acceptance test)
- Can be in the form of equivalence partitioning, boundary value analysis, decision tables or state-transitioning

2.1 Positive and Negative Testing

- Positive: check that the product does what it's supposed to do (supply valid input, expect valid output)
- Negative: Check that product does not fail when given unexpected input (supply invalid input, return error message)
- Coverage is well defined in positive testing, and positive testing maps exactly to a specific requirement

2.2 Boundary Value Analysis

- Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.
- 5 types of values are chosen: Minimum, Just above the minimum, A nominal value, Just below the maximum, Maximum
- Reasoning: Most defects occur at these boundaries (from a white-box perspective, decision boundaries)

2.3 Decision-Based Testing

- A decision table is a black-box test technique that presents combinations of inputs and outputs, where inputs are conditions or cases, and outputs are actions or effects.
- Distinct combinations of decision variables in a program lead to different scenarios
- Each scenario occupies a row in the decision table, and the row also has expected results

2.4 Equivalence Partitioning

- Equivalence partitioning is a testing technique that divides the input data into partitions of equivalent data from which test cases can be derived.
- Divide infinite input space into partititons. One element from each partition acts as a representative of that partititon

2.5 State Graph Testing

- Useful for language processors, compilers, workflow/data modelling
- Design test cases corresponding to each valid state-input combination. Design test cases corresponding to the most common invalid combinations of state-input.

3 Requirement-Based Testing

- Ensures that all requirements are covered
- Review: some implicit requirements are translated to explicit, reqs are prioritized

3.1 RTM

- Mapping between requirements and test cases for those requirements
- Makes sure customer needs are met, test cases capture the requirements, identify extra/missing features, track change requests
- Without RTM: unknown test coverage, expensive fixes later on in the cycle, hard to plan/track projects