



Specification Based Testing

Presented by
Jack Cobb, Laboratory Director

May 20, 2015

Specification Based Testing

- Specification based testing is a type of testing that focuses on the behavior of a system or component, not on how the behavior is achieved.
 - The architecture, design or implementation of the system or component is considered irrelevant.
- The primary focus of this testing is to determine if the system or component satisfies the requirement or specification it is being evaluated against.
 - This is achieved by comparing the system or component behavior to a specific and clearly defined requirement.
- A specification can be any verifiable requirement, design, behavior or characteristic of a system or component.

“Black-Box” Testing

- Specification based testing is sometimes called “black-box” testing because the test object is viewed as a black box in which the testers have no knowledge of the system structure inside the box.
- Focuses on the input and the expected output of a system or component.

Specification Based Testing Techniques

Equivalence Partitioning

Boundary Value Analysis

Decision Tables

State Transition Testing

Equivalence Partitioning– Evaluating valid and invalid inputs to a software component.

Boundary Value Analysis– Evaluating valid and invalid inputs to a software component at the boundaries between partitions.

Decision Tables– Tables that display the combinations of inputs with the associated outputs that are used to design test cases.

State Transition Testing– Used when a system has defined states and transitions between those states.

Specification Based Testing Techniques

The previously listed techniques focus on the inputs that the system expects and the expected outputs of the system. To develop test cases for testing at the system-level or acceptance-level of testing, use cases are utilized.



Use Cases

Use Cases– Use case technique is evaluating a system using a specific sequence of transactions between roles and the system with known results.

Equivalence Partitioning

- Evaluating valid and invalid inputs to a software component.
- The idea behind the test case development is to partition a set of test parameters into groups that can be considered the same and then cover each partition once to examine the system behavior for valid and invalid data.
- Can be applied at any level of testing and is often a good technique to start with but should not be used as the only technique when evaluating a software component.
- Should be supplemented with boundary value analysis.

Boundary Value Analysis

- Evaluating valid and invalid inputs to a software component at the boundaries between partitions.
- The idea behind the test case development is to test the minimum and one less than the minimum and also test the maximum and one more than the maximum.
- Allows for evaluation of both valid and invalid boundaries.
- Used to ensure the software component is designed to handle values in a controlled manner.

Decision Tables

- Tables that display the combinations of inputs with the associated outputs that are used to design test cases.
- Used to capture requirements that contain logical conditions and document internal system design.
- Consist of conditions, actions and rules.
 - Conditions are used to display the input condition.
 - Actions are used to display the actions that will be taken based on combinations of conditions.
 - Rules define a unique set of conditions that results from the actions taking place.

State Transition Testing

- Used when a system has defined states and transitions between those states.
- An event in one state might produce one action; the same event from another state produces a different action.
- Four basic parts:
 - The states that the software may occupy
 - The transitions from one state to another
 - The events that cause a transition
 - The actions that results from a transition
- One of the advantages of state transition technique is the model is flexible and can be as detailed as needed.

Use Case Testing

- Evaluating a system using a specific sequence of transactions between roles and the system with known results.
- Used to develop test cases for testing at the system-level or acceptance-level of testing.
- The idea behind the test case development is to define all the roles and their respective use of the system.
- Each use case describes the process flow through the system for each role.
- This technique captures the system's functional requirements from the user's perspective.
- Each use case consists of a sequence of simple steps, beginning with a user's goal and ending when the goal is fulfilled.

Specification Based Testing

It should be noted that specification based testing focuses on testing the functionality of the system against a set of stated requirements (such as a requirements matrix developed for a certain standard), but it is typically inadequate for evaluating the system against high-risk scenarios or complex situations.