Use Cases From A Testing Perspective

Magdy Hanna, Ph.D.

International Institute for Software Testing
mhanna@iist.org
What is Wrong with Natural Languages For Requirement Definition

- Natural language text is not a reliable way of specifying requirements.
  - Ambiguous
  - Vague
  - Incomplete
  - Difficult to communicate and understand
  - Not easily testable
  - Does not stimulate thinking
  - Takes time to write and keeping it current is a tedious task
Solution!

Model-Based development and testing
What is a Model?

• A description of some aspect of the software system
• It uses precise notation that has syntax and semantics
• Easy to communicate with analysts, testers, developers and customers
• Concise, unambiguous, and to the point
Why Models

• To capture different aspects of the system under consideration
• To facilitate discussions and communication between analysts, users, analysts and developers
• Models can be used to build prototypes faster and to generate implementations automatically
Why Models

Most Importantly:

• Models provide complete, unambiguous, and detailed information for both testability and achievability of requirements

• Almost every model of the system will provide for test design that covers much more scenarios than what can be tested using natural language text
Models for Specifying Requirements

• Use any of the following models to augment natural language text:
  – Data Models
  – User Interface Model
  – Process Models
  – Object Models
  – State Models
  – User Interface Models
  – Decision Tables
  – Decision Trees
  – Use Cases
Where Do Models Fit in the Test Process?

- Functional Requirement
- Test Scenario
- Test Case
- Test Procedure (Script)
Where Do Models Fit in the Test Process?

- Functional Requirement
- Model
- Test Scenario
- Test Case
- Test Procedure (Script)
Where Do Models Fit in TDD

- Functional Requirement
- Model
- Test Scenario
- Test Case
- Test Procedure (Script)
- Design and Coding
Use Case: Definition

A Use Case is any set of interactions between the system and any set of actors (users or other systems) to complete a function.
Where Do Use Cases Fit?

- Functional Requirement
- Use Case
- Test Scenario
- Test Case
- Test Procedure (Script)
Reservation Agent

Make Reservation

Check Flight Status

Cancel Reservation

Check In

Upgrade to First Class

Waive a Penalty

Customer

Accounting System

Reservation Supervisor
Example Use Cases

1. Customer makes a flight reservation
2. Reservation agent makes a flight reservation on behalf of customer
3. Reservation agent cancels a flight reservation on behalf of customer
4. Reservation supervisor waives a penalty
5.
6.
7.
8.
Primary and Alternate Scenarios of a Use Case

A use case typically has a number of scenarios, each representing a usage of the system.

• Primary scenario:
  – The most common way for the use case to happen; as if everything goes well.
  – It represents the normal functionality described by the use case

• Alternate scenarios:
  – Any sequence of actions and events that are different from the one described in the primary scenario. This includes error conditions.
Components of a Scenario

• Pre-conditions:
  – Anything that must happen before the scenario can start and complete successfully
  – What state the system must be in before the scenario starts and completes successfully

• Steps
  – All interactions between the system and actors that are necessary to complete the scenario

• Post-conditions:
  – Anything that must be true after the scenario has been completed
  – What state the system acquires after the scenario has been completed successfully
Use Case
“Reservation agent cancels a flight reservation”

Primary Scenario: “Happy Path”

• Pre-conditions:
  1. Reservation exists
  2. Reservation was not cancelled before
  3. Reservation has been paid
  4. Credit card used for payment still valid
  5. Flight time is at least 24 hours before cancellation time (business rule)
  6. A boarding pass has not been issued
  7. A seat has been assigned to passenger
Steps

BEGIN

1. Reservation agent enters reservation number
2. System displays reservation information
3. Reservation agent clicks “cancel”
4. System marks the reservation as being cancelled
5. System releases the seat
6. System sends a credit transaction to accounting system
7. Accounting system returns a credit transaction confirmation number
8. System displays credit transaction confirmation number

END
Post-conditions

1. Reservation will have been marked as cancelled
2. Seat will have been made available
3. A credit transaction has been generated by the system and confirmation number logged
Use Case
“Customer makes a flight reservation”

Primary Scenario: “Happy Path”

• Pre-conditions:
  1. Database has availability for a round trip between two cities
  2. A valid credit card is available
  3. Customer is logged in and validated
Steps

1. Customer selects type of trip (round trip or one way)
2. Customer enters departure city, arrival city, departure date and return date
3. Customer clicks “search”
4. System displays available itineraries and prices
5. Customer selects an itinerary
6. System displays the passenger information page
7. Customer fills out passenger information
8. System displays payment page
9. Customer fills in credit card information
10. Customer clicks “purchase”
11. System sends a charge transaction to accounting system
12. Accounting system returns a payment confirmation number
13. System displays confirmed itinerary
Post-Condition

- Reservation has been made
- Seat has been assigned and removed from inventory
- A confirmed credit card charge transaction has been posted
- Customer is still logged into the system
How to Find More Scenarios

• Check every step in every scenario for any of the following:
  – Any other action that can be taken at this point – this will result in another alternate scenario.
  – Any event that can happen that can cause the system to behave differently – this would result in another alternate or error scenario.
  – Anything that can go wrong (by the system, the user, or any other actor) – this will result in an error scenario.
Some Alternate Scenarios for Use Case “Customer makes a flight reservation”

“Cancel Reservation” Alternate Scenario:
Any time before the customer clicks “Purchase” in step #10, customer may click “cancel” or close window
Post condition: No reservation is made and user is still logged into the system

Invalid Credit Card # Alternate Scenario:
In step #12, the accounting system may notify that the credit card has been declined
The system will allow the customer to re-enter/correct information
The scenario starts at step #7

“System Failure 1” Alternate Scenario:
10.1 Anytime before Customer clicks “Purchase” in step #10, the system may crash
   a) System saves partially completed reservation in a temporary file
   b) When system is up, an email will be sent to customer with instructions to access the system and complete the reservation
Post condition: No reservation is made
Deriving Tests Based on Use Cases

• Create tests to test each scenario
• Start with the primary scenario if one has been identified
• Select scenarios to test based on:
  – Risk
  – Frequency of use
  – Complexity
Deriving Tests Based on Use Cases (cont.)

• Compose test cases as follows:

  Input + Pre-condition  ➔  Output + Post-condition

• Pre-Condition: Use the pre-condition of a scenario as the pre-condition of your test case
• Input: Identify any input in the steps of the scenario or in the pre-condition as input for your test case
• Expected Results: Use the post-condition of a scenario along with any output described in the steps of the scenario to compose your expected results of your test case
• Test Procedures: Use the steps within the scenario as your test procedures
A Test Procedure Example

**Use Case:**
“Customer makes a flight reservation”

**Scenario:**
Primary scenario (Happy path)

**Test Procedure:**
- Make sure all items under the pre-condition are met
- Follow the steps within the scenario
- If ALL items in the post-condition are met, mark the test case as “Pass” otherwise, mark it as “Fail”.

Repeat the above procedures for different pre-conditions and input
Destructive (Negative) Testing

• There are two sources of negative test cases:
  – Test cases that exercise alternate scenarios which use invalid data
  – Test cases that violate items in the pre-condition

• Make sure to cover most if not all these cases.
Example Negative Tests for
Use Case "Reservation agent cancels a flight reservation"

Based on the 7 pre-conditions of the primary scenario:

1. Cancel a reservation that does not exists
2. Cancel a reservation that was not cancelled before
3. Cancel a reservation that has not been paid
4. Cancel a reservation where credit card used for payment is not valid any more
5. Cancel a reservation where flight time is in less than 24 hours from cancellation time
6. Cancel a reservation for which a boarding pass was already issued
7. Cancel a reservation for which a seat has not been assigned to passenger
Conclusion

- Use cases will help identify more scenarios that can ever be identified using natural language text or any other modeling technique
- Use cases must be developed with testing in mind
- Test team participation in the development and validation of use cases is crucial to the quality of use cases
- Remember to apply Use Cases ONLY for what they were made for, “INTERACTIONS”
- Use Cases should not be used to describe logic or algorithms
- Project teams that use Use Cases to replace requirements are missing the point