USE CASE BASED TESTING

A business oriented test methodology

Lakshmi Prasad Parimi¹ and Veerabhadraiah Sandru²

¹ Designation: Assistant Consultant
Department: Hitech Assurance COE
Company: TATA Consultancy Services
Address: Kohinoor Park, Plot No.1, Jubilee Gardens, Hyderabad -500081, AP, India
Email Address: Lakshmiprasad.parimi@tcs.com

² Designations: Consultant
Department: Hitech Assurance COE
Company: TATA Consultancy Services
Address: Kohinoor Park, Plot No.1, Jubilee Gardens, Hyderabad -500081, AP, India
Email Address: sandru.veerabhadraiah@tcs.com

Abstract

For any product or IT based organizations, the delivery quality is always a driving factor to achieve their business goals. IT products and applications increasingly comprises of application penetration, complex business processes, continuous growth of requirements, external and internal dependencies. To reduce IT spends and to manage their growing requirements and business Goals, customers are refining their quality processes more aligned to business objectives. Use case based testing is a structured business oriented approach which addresses the above mentioned challenges.

Use case is a series of end user actions on a system to achieve a business goal. Use case based testing helps in achieving the delivery quality with business oriented approach in line with the business Goals.

This White paper explains how the use case based testing forms the basis for business oriented test approach. Also a quick glimpse on the use case benefits, creation and its applicability.

1. Introduction

Delivering the right product/application, in right time, with in right budget is essential to live with the competitive world for any IT based organizations. Organization always have to work on these SLAs targeting qualitative and rapidly roll out service to meet new business demands, and to attain constructive view of the delivery quality in line with the business requirements.

But, based on the observations and experiences, we often hear few concerns from the customers on the level of test process maturity that the IT test teams currently adopting to validate the system. Few of them are as below:

- Testing without business knowledge. Technical oriented rather business oriented
  o Testing is based on unrealistic expectations and invalid assumptions
  o Unaware of end user objective, Out of sync with business testing
  o Poor Test data selection, Less coverage on exceptional scenarios
- Significant increase in UAT / Pre-production effort due to Post-IT defects

We need a seamless approach to address the testing challenges and to achieve the end user requirements. In this context, different methodologies and test models are evolved and practiced in the industry such as Experience based testing, defect based testing, model based testing, exploratory testing, use case based testing etc., these models consider tester’s domain knowledge, skill set, realistic data, and understanding on system etc., into confidence.

To achieve the customer expectations and business goals, we need to adopt a business oriented test approach. Use case based testing would be better choice as it evolves from the end user’s business transactions and realistic scenarios.
2. Use cases

Before we discuss on the use case based testing, let’s focus on understanding use case and features.

Use Case is created from the series of steps an actor performs on a system to achieve an end customer goal. Use case composes of actors (Primary and Secondary actors), systems, workflows and events. Use case representation can be done using Unified Modeling Language (UML) / Case diagrams.

<table>
<thead>
<tr>
<th></th>
<th>Identify</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identify actors and objects</td>
<td>Document all basic, alternate and exceptional flows</td>
</tr>
</tbody>
</table>

Actors can be a humans or a system who initiates an action or set of actions within the system. Primary actor will play a role of initiating the action/task in the system and Secondary actor will play the role of responding the primary actor’s action. Events are the triggers that would sent by system while the task is being processed. Set of tasks along with events between the actors can form a workflow.

2.1 Use case – an object oriented paradigm

Ivar Jacobson, a famous author for object oriented software engineering has described about use cases in his book “Object Oriented Software Engineering a Use Case Driven Approach.” As per him, use case acts as an object-oriented paradigm. The concept of use cases is similar to object oriented design, where complex program divided into simple objects and made as re-usable and extensible. Similar to the object oriented concept, use case also has set of classes identified along with their inter dependencies / relationships.

2.2 Use case creation process

Use case creation process is a set of the activities that would involve in understanding, identification, definition of the end user actions and the responses from the system.

Following are the sequence of steps that would be followed while creating a use case:

Identify:

This phase is intended to identify the actors or objects and the tasks that would be involved in the system. Detailed activities are as below.

- Conduct a business/ end user workshop to capture all system components
- Ask client and domain experts how system will be used by end users
- Who (actors) will perform the obvious tasks? How they interact with other systems?
- Clear and differentiated names

Outline:

During this phase all the dependencies, relationships between the actors, system, events and use case paths will be identified. Detailed activities are as below.

- Tasks (various workflows or user paths) does actor want the system to perform
- Information must the actor provide to the system
- Events that the actor must tell system about
- If the actor need to be informed when something happens in system

Detail:

This phase helps in creation of the use case with outcome of above phases. Detailed activities are as below.

- Identify and document all basic, alternate and exceptional flows
- Define goal success and goal failure criteria
Create an use case with set of all events, workflows
Identify the data actor uses in the real system

2.3 Use case applicability
Use case provides a powerful communication approach between end customer, business and IT stakeholders.

At the requirement level, use case helps in understanding and representing project’s business goals, pertaining to all the project stakeholders, including direct users, or actors, creators of the system.

At design level, Use cases are the best representation of user tasks and a step-by-step description of a basic course of actions, exception conditions and variant paths.

Functional requirements are typically written from the point of view of the software, but use cases are written from the end user transactions and hence they can serve as foundation for developing system test cases.

Use cases can be used in all the phases of SDLC provided goal success and goal failure criteria for each workflow is clearly defined.

3. Use case based testing (UBT)

3.1 Approach
Use case based testing focuses every aspect of the application / product considering all the end user actions and system responses. Used case based testing process typically identifies user paths consists of basic, alternate, negative workflows which forms the basis for validation of system behavior to end user actions.

Few aspects of use case based testing:
- Pre and post conditions, Business Rules (UI / Data Elements)
- Basic Flow (No branches, No loops, No conditions, happy path)
- Alternate Flows (Self Contained, Consistent format)
- Negative flows, Error messages , Exceptional handling, Special Requirements
- End to end system verification (inclusive of all above)

Below diagram describes the test design process in identifying the test cases from Use cases:

3.2 Example
Let’s take an ATM example to describe the use case based testing process:

In the above example, primary and secondary actors and expected goal success criteria and goal failure criteria has been identified. All the possible valid user scenarios i.e., basic, alternate and invalid user scenarios i.e., exceptional flows are outlined and detailed in form of workflows. Every workflow will become a test case with set of test steps and thus ensures the test coverage for a given use case in the user perspective. Business SME inputs and review for every use case would further help in better coverage in testing.

Below described test design template derived from the use case based testing. This would ensure traceability from user scenarios / use cases to the test cases.

At the coverage prospective, minimum test coverage will achieve by having one test case at least for every workflow
in use case. Maximum test coverage can be achieved while deriving at least one test case for every path of workflow.

3.3 Benefits
Benefits of the use cases and use case based testing are as follows:

- Early requirements elicitation and review from the user’s perspective.
- Provide the basis for identifying a system's key internal components, structures, databases, and relationships
- Use cases (UML diagrams) can be leveraged for design and implementation of Objected oriented applications with the identification of major classes (objects/actors) and inters dependencies / relationships.
- Can be adoptable for rapid feature development and testing in agile environments.
- Serve as the foundation for developing system test cases

3.4 Is UBT is replacement to Functional testing?
The evolution of the use case based testing is not the indication for the replacement of traditional functional testing. Use case based testing also focuses on the functionality of the system, but with the User perspective rather feature perspective.

Use case based testing will helps in bringing value-add to the functional testing as it focuses on the system compliance to that of end user’s transactions. Requirement based traditional testing will still be valid along with use case based testing to complete the test coverage at system feature perspective.

3.5 Agility with use case based testing
Due to the rapid changes in the industry, the companies are implementing agile methodologies to adopt the changes in their software/ products. Use case-based testing can be leveraged for agile methodologies by involving testers in use case development and harvesting use case scenarios. Testers will review the use cases in the early phases and will come up with set of user stories. These user stories will become the basis for preparation of the test cases.

3.6 Tools and technologies
Few of the industry tools / methodologies that are evolved in recent times are as below.

<table>
<thead>
<tr>
<th>S no</th>
<th>Tool/technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM Use Case Based Testing (UCBT)</td>
<td>UCBT is a technique addresses phases where the tester is interested in exploring behavior that flows through multiple use cases.</td>
</tr>
<tr>
<td>2</td>
<td>Smart testing tool</td>
<td>Takes UML diagrams, user flow diagrams as an input and generates E2E test cases, Automation scripts and optimal data combinations</td>
</tr>
<tr>
<td>3</td>
<td>PLUTO (Product Lines Use Case Test Optimization)</td>
<td>Simple and intuitive methodology for the early derivation of test scenarios from the Product line requirements specification. Helps for the conformance of the derived product with respect to the product line</td>
</tr>
</tbody>
</table>

4. Conclusions
Use cases helps in easy representation of the business goals, responsibilities in readable, refutable format and early and efficient elicitation of user requirements. It helps in better simulation of user Goals at every phase of the product / software creation. The use cases can be reused extended and can be refined specifications of business requirements.

The use case based testing process focuses on end to end verification of the system as described in the business requirement document and derived from the user scenarios of the system. Thus forms as a Business oriented test approach in validating the product.

Early involvement of testers in agile development using use case based testing approach will show better results by efficient requirement understanding, real user scenarios, end user data etc.,. In additional to that Use case based testing will also helps in improving functional testing coverage mapping to end user’s transactions.

Use case based testing is widely used in user acceptance testing, beta testing, and pre-production testing. Also primarily used for systems that needs to respond to external events such as:

- Customer interface, e-commerce applications such as E-learning, E-shopping, HR, banking etc.,
- Product line testing such as Software products, devices, mobiles etc.,

References


6. Validating and testing software system using use case based approach - Suman Kasnia, 2Deepa Mehta - www.ijcem.org/papers72011/72011_08.pdf


First Author

Lakshmi Prasad Parimi 8 years experience in software testing.
B.Tech in Civil Engineering - 2002
Diploma in Civil Engineering - 1998

TATA consultancy Services, Hyderabad
NS software Technologies, Hyderabad

Currently he is solution developer for Assurance CoE of HiTech Industry solution Unit of TATA consultancy services. In his role his focus areas are Test Design Techniques, Test process Consulting, Test methodologies, Test Metrics and the latest testing trends. He has been part of several best practices and initiatives throughout his testing career including knowledge enhancements across the assurance accounts. Lakshmi Prasad has expertise in Test process, design, Test management, automation, performance and database testing.

Second Author

Veerabhadraiah Sandru has 15 years of experience in IT and 8 years experience in software testing.

Master of Science by research in Hydraulics and water resources engineering in 1997 from IIT Madras.

TATA Consultancy services, Hyderabad.

Currently he is leading the Assurance CoE of HiTech Industry Solution Unit of Tata Consultancy Services. In this role, he is responsible for test offerings creation, proposal solutioning, competency enhancements and asset creation and deployment in the areas of software testing. He and his team also participates in testing projects technical reviews and recommends improvements in the areas of software testing. His area of expertise include test process consulting, test management, test automation and test methodologies in usability testing, globalization, accessibility testing.