

# ***Usage Scenario Testing***

---

## **Usage Scenario Testing**

Slide

### **Testing Functional Requirements**

Steps in testing the functions:

- Test each individual function
- **Test usage scenarios**
- Test to the operational profile



## **Steps to Defining Use Cases**

Slides

### **Steps to Defining Use Cases**

Use cases identify how the system will be used.

1. Define the system's scope
  2. Identify the actors
  3. For each actor, identify their interactions with the system
    - ♦ Main tasks or functions performed
    - ♦ Information acquired, produced, or changed
    - ♦ Information given to the system
    - ♦ Information wanted from the system
    - ♦ Notifications wanted from the system
  4. Develop a use case for each interaction
  5. Merge or create sub-use cases as appropriate
- [based on Pressman-01]



*continued on next page*

## **Usage Scenario Testing**, continued

---

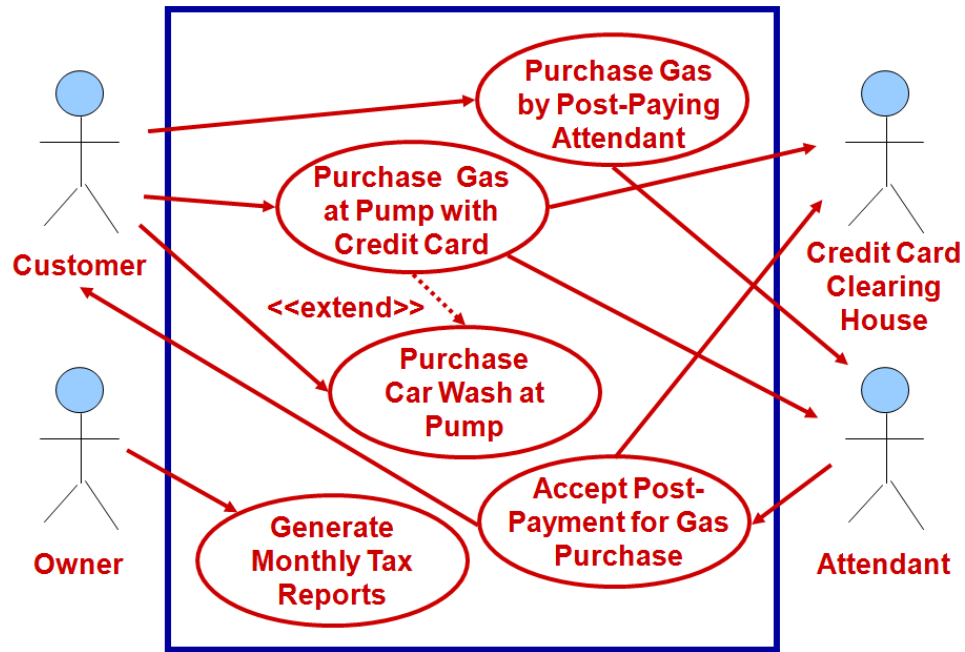
### **Steps to Defining Use Cases (cont.)**

|   |  |
|---|--|
| <b>Use Case</b>                             | <p>A use case is a scenario created to describe a thread of usage for the system to be implemented. An entire set of use cases describes how the system will be used and provides a basis for functional testing.</p>  |
| <b>System Scope</b>                         | <p>The first step in defining use cases is to define scope of the system for which we are defining use cases. This requires the identification of the boundaries of the system under consideration.</p> <p>For example, we can define the entire gas station as the system under consideration, in which case the product we are creating is just one of the components of that system. If the system under consideration is the entire gas station, the actors that interact with it are outside that system. For example, actors would include the Customers purchasing gas and the Bank Consortium that validates the credit cards. The Gas Station Attendant, Manager and Bookkeeping Software would not be actors because they are part of the gas station system.</p> <p>We could also go down a level and define the GasPay product that we are developing as the system under consideration. Now, not only are the Customer and Bank Consortium example actors, but the Gas Station Attendant, Manager, Bookkeeping Software are also examples of the actors for this system under consideration.</p> <p>We can drill down even further and identify one of the components of our GasPay system as the system under consideration. For example, we could identify the Credit Card Reader component as our system under consideration. This would cause the list of actors to again be different because not all of the actors that interact with the GasPay system would interact with the Credit Card Reader component. Other components within the GasPay system that interact with the Credit Card Reader would also now be actors.</p> |
| <b>Actors</b>                               | <p>The second step in defining use cases is to identify the different actors that are external to the software product and interact with the product. Actors are different than users. A user may perform several roles when interacting with the system. For example, a single user of a word processing package could have several different roles including author, reviewer and editor of a document. Each of these roles would be a separate actor. Actors can also be other hardware devices or systems that interact with the software product. For example, the Bank Consortium actor for the GasPay product is another software application that GasPay must interact with to obtain authorization for credit or debit card purchases.</p> <p>The scope of the system determines who the actors are and are not. For example, if the Gas Pump is considered part of the system it is not an actor. However if the Gas Pump is outside the system boundary, it would be considered an actor.</p>   |
| <b>Interactions</b>                         | <p>We must then identify a list of interactions between the actors and the system. These interactions may be initiated by either the actor or by the system on behalf of the actor.</p>  |
| <b>Develop Use Cases</b>                    | <p>Use cases are then developed to describe each start-to-finish interaction. Use cases are the basic scripts that describe the different steps in each interaction between the actors and the product.</p>  |
| <b>Sub-Use Cases &amp; Merged Use Cases</b> | <p>As use cases are created; redundancies in the use cases may be identified. Sub-use cases can be created to include the redundant steps and these sub-use cases can then be included in multiple primary use cases. Smaller sequential use cases can also be merged together to form a single use case as appropriate.</p>   |

### Steps 1-3: Use Case Diagram

Slide

#### Steps 1-3: Use Case Diagram



**Use Case Diagram** A use case diagram model is used to identify use cases and show their interactions to actors and their interrelationship to each other.

**Step 1: Scope** A use case diagram starts with a box that represents the scope or boundaries of the software product.

**Step 2: Actors** Outside the box are stick figures that represent the various actors that will interact with the product. Each stick figure is labeled with the name of that actor. For example, for the GasPay product the actors might be the customer, gas station owner, gas station attendant and the Bank Consortium. As with the context diagram, the scope of the product determines who the actors are and are not. For example, if the gas pump is considered part of the product it is not an actor. However if the gas pumps were outside the product boundary, it would be considered an actor.

**Use Cases** The ovals inside the software product boundary represent the interactions (use cases). Use cases are typically names using a verb and an object. For example, *Purchase Ticket* or *Print a Boarding Pass* or *Add a User*. They may also have additional qualifiers. For example, using adjectives like *Generate Monthly Tax Reports* or other qualifiers like *Purchase Gas at Pump*.

*continued on next page*

### **Steps 1-3: Use Case Diagram (cont.)**

**Step 3:Interactions** The arrows in a use case diagram show the interactions between the actor and the use cases. If the arrow goes from the actor to the use case, it indicates that the actor is the primary actor that initiates that use case. An arrow from a use case to an actor indicates that the actor is a secondary actor involved in the use case but not the actor who initiated it. For example, in the GasPay use case diagram above, the primary actor for the Purchase Gas at Pump use case is the Customer who initiates the transaction. Secondary actors that are also involved include the Bank Consortium and the Gas Station Attendant.

**<<extend>>** Use case diagrams can also indicate the relationships between use cases. An <<extend>> use case adds behavior to the base use case at a specific “extension point”. For example, the Purchase Gas at Pump use case can be extended at a specific point with the Purchase Car Wash at Pump use case.

**<<include>>** Another relationship between use cases is the <<include>> relationship. An <<include>> use case separates behavior that is similar across multiple use cases into a separate use case in order to remove redundancy and complexity. An analogy might be the concept of taking software code that is duplicated in several modules and moving it to a separate subroutine called by each of those modules.

**Simplified** The use case diagram in the example above has been simplified for illustration purposes and does not show all of the possible actors or use cases. Examples of other use cases might include:

- Change Gas Prices
- Change Tax Rates
- Identify Stolen Credit Card

List other use cases that might be included:

- 
- 
- 
- 
- 
- 
- 
-

### **Step 4: Develop a Use Case for Each Interaction**

#### **Slide**

#### **Step 4: Develop a Use Case for Each Interaction**

Sub-steps to developing a use case:

- Identify primary actor & secondary actors
- Define preconditions
- Define postconditions
- Identify main success scenario
- Document steps in main success scenario
- Identify alternate scenarios
- Identify exception scenarios
- Document steps in these additional scenarios

#### **Use Case**

The details of each identified use case are defined in individual use cases. The use cases define:

- **Primary Actor:** The actor that initiates the use case
- **Secondary Actors:** Other actors that interact with the use case
- **Preconditions:** Specific, measurable conditions that must be met before the use case can be initiated (i.e., entry criteria).
- **Postconditions:** Specific, measurable conditions that must be met before the use case is considered complete (i.e., exit criteria).
- **Main Success Scenario:** Also called the happy path, is the normal or typical sequence that results in a successful interaction with the actor
- **Alternative Variation Scenarios:** Other alternate sequences are variations that still result in a successful completion of the task (i.e., satisfaction of the post conditions)
- **Exception Variation Scenarios:** Other exception sequences are variations that result in an unsuccessful completion of the task (i.e., post conditions are not satisfied)

## ***Usage Scenario Testing,*** continued

---

### **Use Case - Example**

#### **Slides**

#### **Use Case - Example**

**Use Case:** Accept Payment for Gas

**Primary Actor:** Attendant

**Secondary Actors:** Customer, Bank Consortium

**Preconditions:**

- Gas has been successfully pumped
- Customer has arrived at Attendant to pay for gas

**Post Conditions:**

- Payment has been received
- Customer was able to pay for gas with payment type of preference
- Customer received receipt

**Main Success Scenario:**

| Actor Actions  | System Responses  |
|--|---|
| 1. Attendant greets Customer & asks which pump                             |   |
| 2. Customer identifies pumps   |   |
| 3. Attendant polls system for pump information                             | 4. System reports gallons pumped & total price                    |
| 5. Attendant confirms price with customer & acknowledges price with system | 6. System accepts price & displays price on cash register display |
| 7. Attendant ask if other items & Customer responds no                     |   |
| 8. Attendant asks for payment type & Customer pays in cash                 |   |
| 9. Attendant enters cash tendered into cash register                       | 10. System calculates & displays change & prints receipt          |
| 11. Attendant provides change & receipt to customer                        |   |
| 12. Attendant ends transaction   | 13. System saves transaction information & resets pump            |

---

*continued on next page*

## Usage Scenario Testing, continued

### Use Case - Example (cont.)

#### Slides Use Cases – Example (cont.)

##### Alternative Scenarios:

| Actor Actions   | System Responses  |
|---|---|
| 5a1. Customer identified <b>wrong pump</b> & changes pump identification<br>5a2. Attendant cancels previous pump<br>5a4. Return to step 3 | 5a3. System resets to no pump identified  |
| 7a1. Customer <b>purchase other items</b><br>7a2. Attendant enters prices for each item in cash register<br>7a4. Return to step 8         | 7a3. System accepts price & displays item prices & running total price on cash register |

##### Exception Scenarios:

| Actor Actions   | System Responses  |
|---|---|
| 4a2. Attendant manually checks pump display & enters amount into cash register<br>4a3. Return to step 7   | 4a1. System <b>can not communicate with pump</b>  |
| 8c1. Attendant asks for payment type & Customer swipes <b>invalid credit card type (not accepted type of card)</b><br>8c4. Return to step 8   | 8c2. System reads & parses magnetic strip<br>8c3. System displays error   |
| 8d1. Attendant asks for payment type & Customer swipes <b>invalid credit card (expired, reported stolen or over limit)</b> correctly (after one or more tries)<br>8d4. Bank Consortium disapproves transaction<br>8d6. Return to step 8 | 8d2. System reads & parses magnetic strip<br>8d3. System establishes communications with Bank Consortium & transmits merchant information, credit card information & transaction amount<br>8d5. System displays disapproval |

*continued on next page*

## **Usage Scenario Testing,** continued

---

### **Use Case - Example (cont.)**

#### **Other Alternative Scenarios:**

| <b>Actor Actions</b>   | <b>System Responses</b>   |
|--|---|
| 8a1. Attendant asks for payment type & Customer swipes <b>valid credit card</b> correctly (after one or more tries)  | 8a2. System reads & parses magnetic strip<br>8a3. System establishes communications with Bank Consortium & transmits merchant information, credit card information & transaction amount<br>8a5. System displays approval                          |
| 8a4. Bank Consortium approves transaction<br>8a6. Return to step 12  |   |
| 8b1. Attendant asks for payment type & Customer swipes <b>valid debit card</b> correctly (after one or more tries)<br>8b3. Customer enters <b>correct PIN number</b> | 8b2. System reads & parses magnetic strip & prompts for PIN number<br>8b4. System establishes communications with Bank Consortium & transmits merchant information, credit card information & transaction amount<br>8b6. System displays approval |
| 8b5. Bank Consortium approves transaction<br>8b7. Return to step 12  |   |

#### **Other Exception Scenarios:**

| <b>Actor Actions</b>  | <b>System Responses</b>  |
|---|--|
| 4a2. Attendant manually checks pump display & enters amount into cash register<br>4a3. Return to step 7                             | 4a1. System <b>can not communicate with pump</b>   |
| 10a2. Customer doesn't get receipt<br>10a3. Return to step 12   | 10a1. System <b>can not print receipt</b> & displays error   |
| 8e1. Attendant asks for payment swipes valid/invalid credit/debit card correctly (after one or more tries)<br>8e4. return to step 8 | 8e2. System <b>cannot read magnetic strip</b> & displays error<br>8e3. System displays error message |

---

*continued on next page*



## Usage Scenario Testing, continued

### Use Case - Example (cont.)

#### Other Exception Scenarios: (cont.)

| Actor Actions  | System Responses   |
|--|--|
| 8f1. Attendant asks for payment type & Customer swipes valid debit card correctly (after one or more tries)  | 8ef2. System reads & parses magnetic strip & prompts for PIN number  |
| 8f3. Customer enters <b>incorrect PIN number (3 times)</b>   | 8f4. System establishes communications with Bank Consortium & transmits merchant information, credit card information & transaction amount |
| 8f5. Bank Consortium disapproves transaction & reports invalid PIN   | 8f6. System displays PIN error message   |
| 8f7. Steps 8e4 through 8e6 repeated twice more so debit card & PIN rejected  | 8f8. System displays error and terminates transaction  |
| 8g1. Attendant asks for payment type & Customer <b>invalid debit card (expired, reported stolen or over limit)</b> correctly (after one or more tries) | 8g2. System reads & parses magnetic strip & prompts for PIN number   |
| 8g3. Customer enters correct PIN number  | 8g4. System establishes communications with Bank Consortium & transmits merchant information, credit card information & transaction amount |
| 8g5. Bank Consortium disapproves transaction   | 8g6. System displays disapproval   |
| 8g7. Return to step 8  |  |
| 8h1. Attendant asks for payment type & Customer swipes <b>valid/invalid credit card</b> correctly (after one or more tries)                            | 8h2. System reads & parses magnetic strip  |
|  | 8h3. System <b>can not establishes communications with Bank Consortium</b>   |
| 8h5. Return to step 8 (requires cash payment)  | 8h4. System displays communication error message   |
| 8i1. Attendant asks for payment type & Customer swipes <b>valid/invalid debit card</b> correctly (after one or more tries)                             | 8i2. System reads & parses magnetic strip & prompts for PIN number   |
| 8i3. Customer enters correct/incorrect PIN number  | 8i4. System <b>can not establishes communications</b> with Bank Consortium   |
|  | 8i5. System displays communication error message   |
| 8i6. Return to step 8 (requires cash payment)  |  |

### **Other Use Case Information**

#### **Slide**

#### **Other Use Case Information**

Other information that should be defined for each use case:

- Use case identifier & use case name
- Created by & date
- Modification history (modifiers & dates)
- Use case description
- Priority
- Frequency of use
- Related business rules
- Assumptions

#### **Priority**

Establishing use case priorities help establish the priorities of the product level requirements derived from the use cases. High priority use cases include:

- Use cases that are critical to the success of the product
- Use cases that are necessary or important to high priority stakeholders
- Use cases that describe activities that are performed frequently
- Use cases that describe activities that are performed by many different stakeholders
- Use cases needed to meet regulatory or legal requirements