

## Use Cases

### Lecture 02

The first step in getting what you want is to decide what you want.

1

### Case Cases: What ??

- Use cases are text stories (not diagrams!) used to discover and record requirements
- If a diagram clarifies the text, use it
- Scenario – Specific sequence of actions and interactions between actors. (also called a use-case instance)
- Functional Requirements



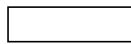

2

### Use Case: Why...?

- Use cases are not part of OO Analysis but these are key requirement input document
- Simple and easy way to ensure various non-technical stakeholders' involvement.
- Ensure goal based requirement specification

3

### Terms and Symbols

- Actor – something with a behavior, such as a person, an input device, etc. 
- Use Case is a collection of related success and failure scenarios that describe an actor using a system to support a goal 
- Boundary 
- Connections 

4

### Three Formats

- **Brief** – Terse, one-paragraph summary, usually the main success scenario. Create during early requirements phase.
- **Casual** – Informal paragraph format. Can cover various scenarios (alternate flow as well) in multiple paragraphs.
- **Fully-dressed** – All steps and variations written in detail. Has supporting sections, success guarantees, main scenario, alternate scenarios, etc.

5

### Scope

- Defines how broad the use case is. This can be for the whole system, as in the POS example, or narrow, as in a use case for creating a journal entry in an accounting system.

6

### Roles

- The person (or sometimes object) that calls upon system services to fulfill a goal is Actor
- The stakeholders are people who have a reason to want this system. The Interests are their reasons for wanting it and what they expect from it.

7

### Level

- **User-goal**: Common kind of scenario. Scenarios that let a user get something done. Typically primary Actors' goals are fulfilled.
- **Sub function**: smaller steps required to support a user goal. Used to factor out duplicate sub steps shared by several use cases e.g. Pay by Credit.

8

### Actors

- Anything that have behavior e.g. people, machine, organization, software
- Primary Actors
- Supporting Actors
- Offstage Actors
- Actor name in Capital e.g. Cashier, Manager

9

### Primary Actors

- Primary actors has user goals fulfilled through using services of the system
- Why??
- To find user goals
- Goals derives requirements  $\longleftrightarrow$  use case

10

### Supporting Actors

- Provides services to system e.g. information
- Payment Authorization service is supporting actors
- Why..?
- To clarify external interface and protocols

11

### Offstage Actors

- Actors have interest in the behavior of the use case e.g. tax agencies
- Why...?
- To ensure all necessary interests are identified and satisfied
- Easy to miss unless explicitly named

12

### Main Success Scenario

- This satisfies the interests of the stakeholders. You get your groceries, the store gets your money, inventory is reduced, etc.
- Steps:
  - An interaction between actors
  - Validation (by the system)
  - State change to the system

13

### Extensions or Alternate Flows

- These include all other possible outcomes, both success and failure.
- Used mostly in fully dressed use case.
- Complex and larger than happy path
- Usually branch out from main scenario n then merge back to it.
- Complex extensions can be a separate use case.

14

### The include Relationship

- Don't duplicate text. Separate it into its own subfunction use case and indicate its inclusion
- Paying by credit: include *Handle Credit Payment*

15

### Preconditions and Success Guarantee

- These should be non-obvious. *System is on*
- *Preconditions* state what must ALWAYS be true before you can start the scenario. This often defines the success of another use case.
- *Success guarantees* state what must be true on successful completion of the use case.

16

### Use Case: Scenario

Process Sale: A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and leaves with the items.

17

### Performing Another Use Case

- Use cases can branch to other use cases. For example, if a POS system rejects a bar code, the cashier can request alternate lookup.
- Denote this by underlining: Cashier performs Find Product Help to get item ID and price

18

### Technology and Data Variations

- Technical variations on how something must be done:
  - Scan bar code
  - Key item ID
- Avoid early design decisions; keep things general.

19

### Write in a UI-Free Style

- Most programs are dependent upon a particular user interface. However, avoid constraining your program too early:
- “The user keys an ID and password into a dialog box and presses the OK button.”
- “The user identifies himself to the system.”
- The latter allows for biometric ID, keyin, etc.

20

### Essential Style

- Focus on the essence, or basic idea, not the details of implementation
- Contrast with concrete style

21

### Write Black-Box Use Cases

- Don't describe internal workings
- Describe *responsibilities*
- "The system records the sale" vs. "The System writes the sale record to a database"

22

### Finding Use Cases

1. Choose the system boundary
2. Identify the primary actors
3. Identify the goals for each primary actor
4. Define use cases that satisfy these goals

23

### Questions to Find Actors and Goals

- Who starts and stops the system?
- Who does user and security management?
- Who does system administration?
- Is "time" an actor because the system does something in response to a time event?
- How are software updates handled?
- Who gets notified of problems?

24

## Actor-Goal List

Actor	Goal
Cashier	Process Sales Process rentals Handle returns Cash in Cash out
Manager	Start up Shut down
System administrator	Add/Modify/Delete users Manage security Manage System tables
...	...

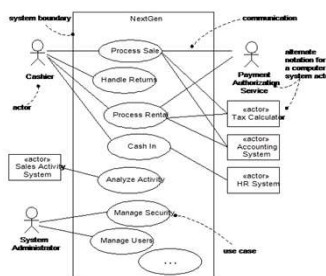
25

## Tests

- The Boss test: "What have you been doing all day?"  
Is this strongly related to achieving results?
- The Elementary Business Process test: Task performed by one person at one place at one time in response to a business event that adds value and leaves data in a consistent state.
- The size test: Fully dressed is 3-10 pages.
- Reasonable violations: Separate sub function, or fails

26

## A simple diagram can add clarity



27

## Requirements in Context

- Use cases organize a set of requirements in the context of a typical use of the system
- High-level feature lists are acceptable
- Some applications need feature-driven viewpoint; don't create use cases for these

28

## Handle Credit Payment

**Level** Subfunction

### **Main success scenario**

1. Customer enters his credit account information
2. System send payment authorization request to external system
3. System receives payment approval and signals cashier

### **Extensions**

- 2a. System cannot communicate with external system

...

29