



*Department of
Software Engineering*

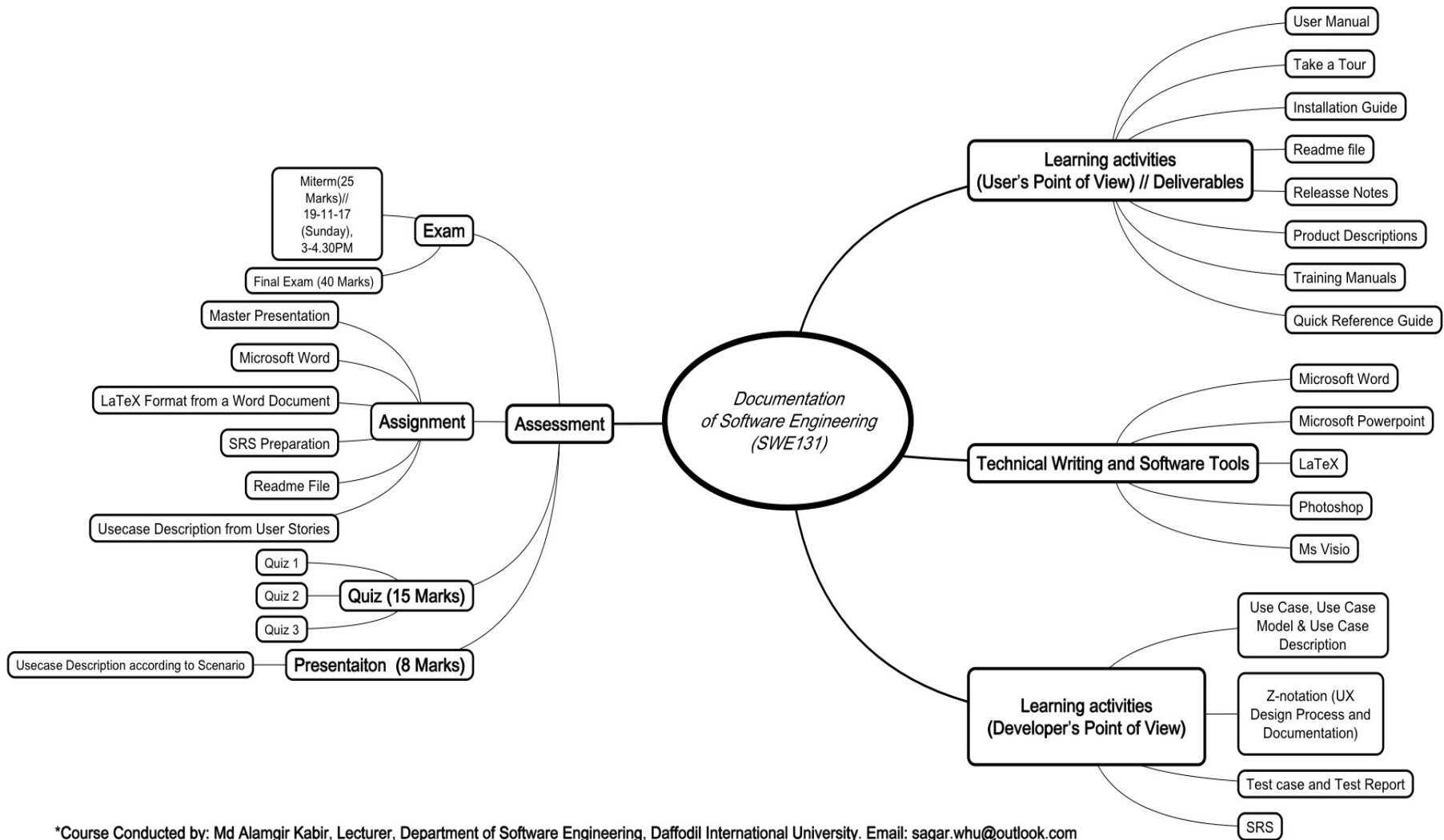
Documentation of Software Engineering

Use Case, Use Case Model, and Use Case Description

Presenter

Md Alamgir Kabir

sagar.whu@outlook.com



*Course Conducted by: Md Alamgir Kabir, Lecturer, Department of Software Engineering, Daffodil International University. Email: sagar.wbu@outlook.com

Outline

- Use case
- Use Case Model
- Use Case Description



*Department of
Software Engineering*

Use Case
Use Case Diagrams
Actors and Examples

USE CASE

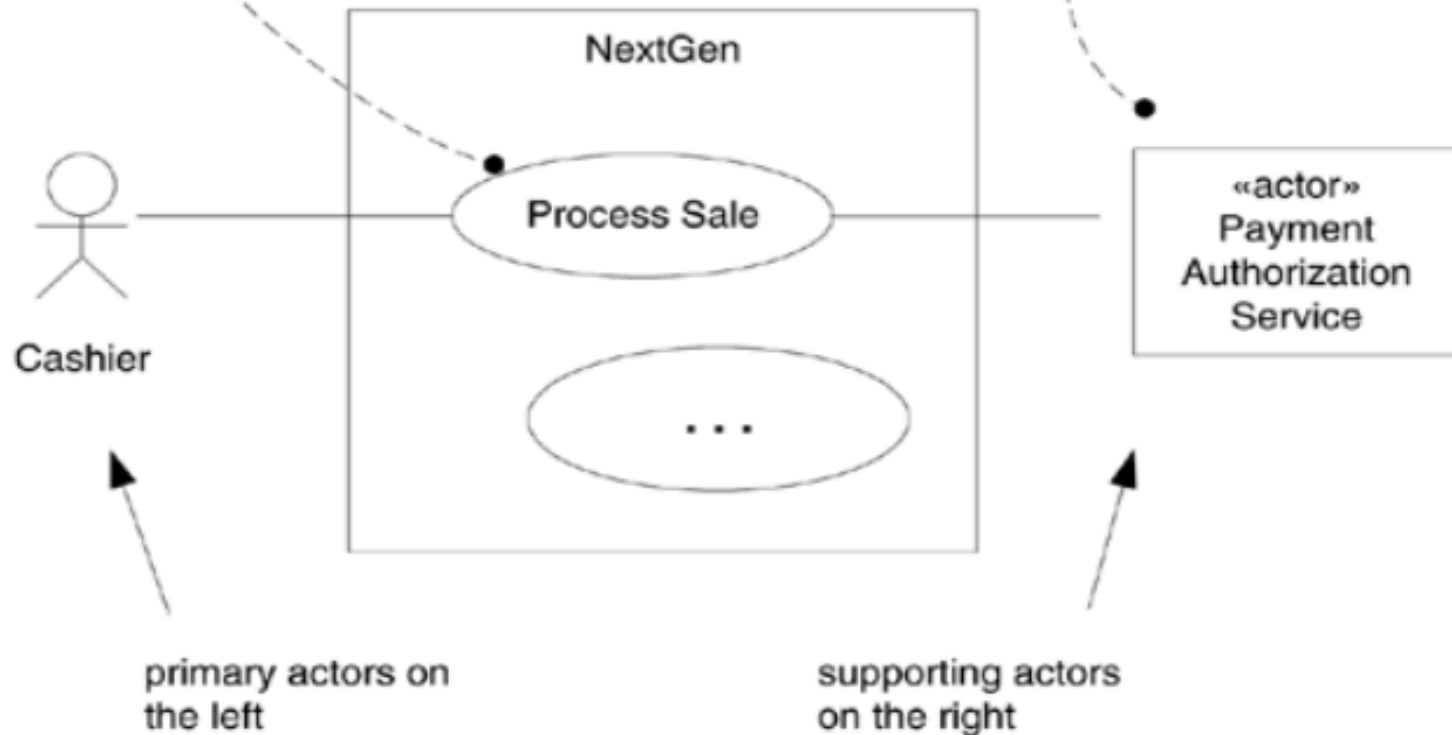
Use Cases

- Informally, use cases are **text stories of some actor using a system to meet goals.**
- The essence of use cases is discovering and recording **functional requirements** by writing stories of using a system to fulfill user goals;
- 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 **then leaves** with the items.

Use Case Diagrams

For a use case context diagram, limit the use cases to user-goal level use cases.

Show computer system actors with an alternate notation to human actors.



Actor

- An **actor** is something with behavior, such as a **person** (identified by role), computer system, or organization, who uses the system
 - for example, a cashier.



Three Kinds of Actors

Primary Actor

- has user goals fulfilled through **using services** of the SuD (**system under discussion**)
- For example, the cashier

Supporting Actor

- **provides a service** (for example, information) to the SuD.
- The **automated payment authorization** service is an example.

Offstage Actor

- has an interest in the behavior of the use case, but is not primary or supporting;
- for example, a government tax agency.

Scenario

- A **scenario** is a specific sequence of actions and interactions between actors and the system; it is also called a **use case instance**.



Use Cases

- A use case is a collection of related success and failure scenarios that **describe an actor using a system to support a goal.**



More example:

- Main Success Scenario:
 - A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item ...
- Alternate Scenarios:
 - If the customer paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash.
 - If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).
 - If the system detects failure to communicate with the external accounting system, ...

Exercise

- Elevator system
 - Provide some valid use cases





*Department of
Software Engineering*

THE FUTURE OF BANKING



*Department of
Software Engineering*

Use Case Model
Use Case Guidelines
Use Case Diagrams

USE CASE MODEL

Use Case Model

- Use-case model is a collection of all written use cases.
 - It defines the **system's functionality**.
- The Use-Case Model may include a UML use case diagram to show the names of use cases and actors, and their relationships.

Use Cases – Guidelines 1




- Guideline 1
 - Write black-box use cases(why?)
 - Do not describe the internal workings of the system, its components, or design.
 - Rather, the system is described as having responsibilities

Black-box style	Not
The system records the sale.	The system writes the sale to a database. ...or (even worse): The system generates a SQL INSERT statement for the sale...

Use Cases – Guidelines 2 & 3

- Guideline 2
 - **Take an Actor and Actor-Goal Perspective**
 - Write requirements focusing on the users or actors of a system, **asking about their goals.**
- Guideline 3
 - During early requirements work,
"keep the user interface out, focus on user intent."

Use Cases – Guidelines 4

-  **Step 1: Choose the System Boundary**
-  **Steps 2 and 3: Find Primary Actors and Goals**
-  **Step 4: Define Use Cases.**

Use Cases – Guidelines 4

- Guideline 4

How to find use cases? A Step by step procedure.

- ✦ Step 1: **Choose the System Boundary**

- the POS system itself is the system under design; everything outside of it is outside the system boundary, including the cashier, payment authorization service, and so on.

- ✦ Steps 2 and 3: **Find Primary Actors and Goals**

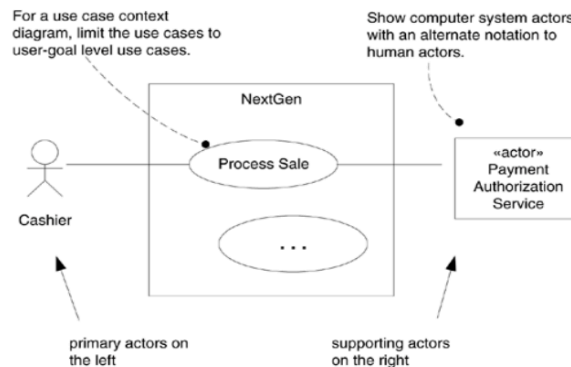
- in a requirements workshop, people brainstorm and generate a mixture of both. Sometimes, goals reveal the actors, or vice versa.

Use Cases – Guidelines 4

- In addition, the following questions help identify others that may be missed:
 - How to Organize the Actors and Goals?

Approach:

- Write an actor-goal list first, review and refine it,
- Then draw them in a **use case diagram**, naming the goals as **use cases**.



Use Cases – Guidelines 4

Actor	Goal		Actor	Goal
Cashier	process sales process rentals handle returns cash in cash out ...		System Administrator	add users modify users delete users manage security manage system tables ...
Manager	start up shut down ...		Sales Activity System	analyze sales and performance data
...

Use Cases – Guidelines 4

- Use **Event Analysis** to find Actors and Goals.

External Event	From Actor	Goal/Use Case
enter sale line item	Cashier	process a sale
enter payment	Cashier or Customer	process a sale
...		

Use Cases – Guidelines 4

- Step 4: **Define Use Cases.**
 - In general, define **one use case for each user goal.**
 - For example
 - Goal: process a sale;
 - Use Case: Process Sale.
 - **Start the name of use cases with a verb**
 - A common exception to one use case per goal is to collapse CRUD (create, retrieve, update, delete) separate goals into one CRUD use case, idiomatically called Manage <X>. For example, the goals "edit user," "delete user," and so forth are all satisfied by the Manage Users use case.

Use Cases – Guidelines 4

- Which of these is a valid use case?
 - Negotiate a Supplier Contract
 - Handle Returns
 - Log In
 - Move Piece on Game Board

Use Cases – Guidelines 4

- There are several rules of thumb, including:
 - The Boss Test
 - The Size Test

Use Cases – Guidelines 4

- **The Boss Test**

- Your boss asks, "What have you been doing all day?"
You reply: "Logging in!" Is your boss happy ? 😊
- If not, the use case fails the Boss Test, which implies
 - It may be a use case at some low goal level, but not the desirable level for requirements analysis.
- **User authentication** may fail the boss test, but may be important and difficult.

Use Cases – Guidelines 4

- **The Size Test**

- A use case is very seldom a single action or step; rather, a use case typically contains **many steps**.
- **A common mistake in use case modeling is to define just a single step** within a series of related steps as a use case by itself.

Use Cases – Guidelines 4

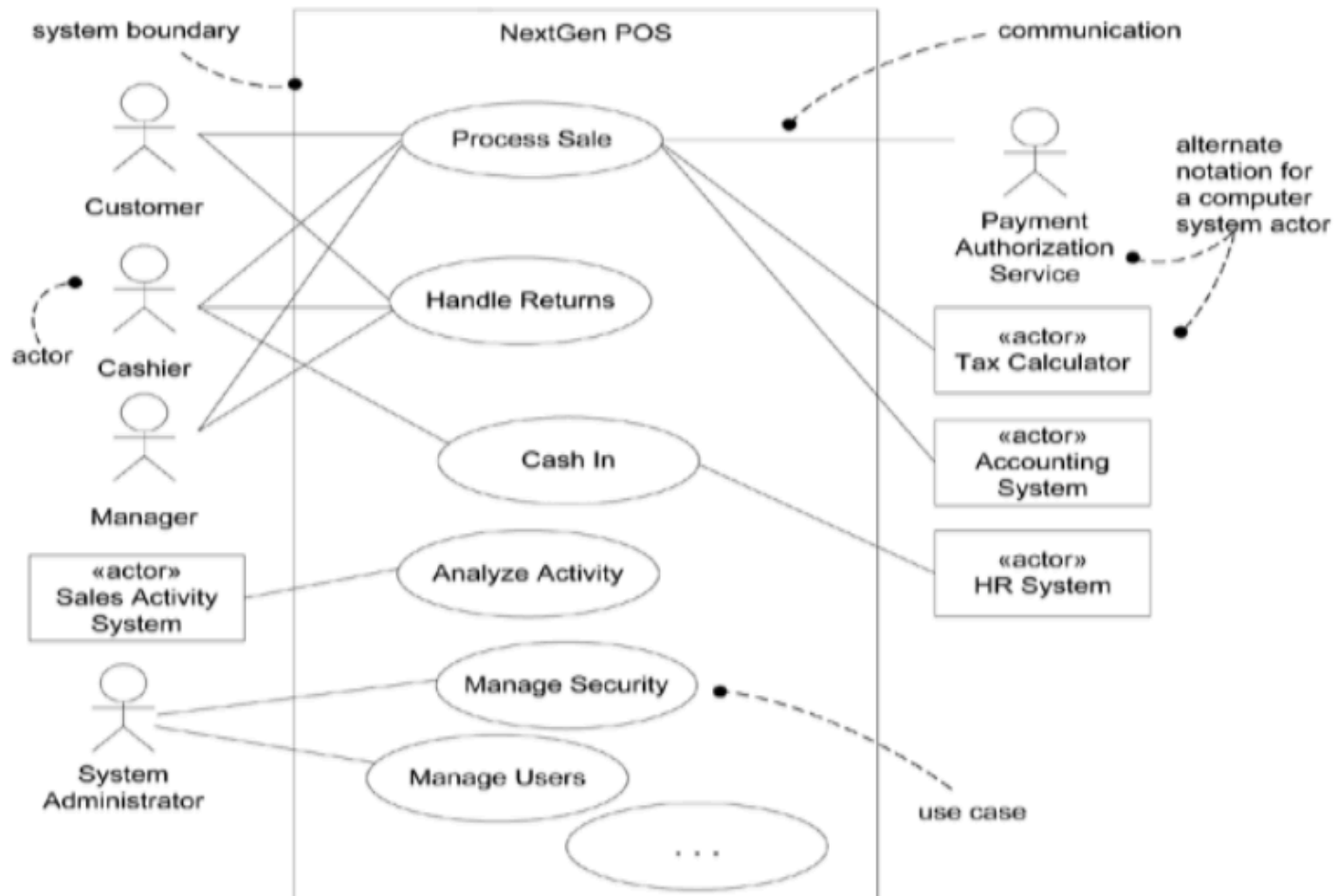
Example: Applying the Tests

- Negotiate a Supplier Contract
 - Much broader and longer than an EBP. Could be modeled as a business use case, rather than a system use case.
- Handle Returns
 - OK with the boss. Size is good.
- Log In
 - Boss not happy if this is all you do all day!
- Move Piece on Game Board
 - Single step - fails the size test.

Use Case Diagrams

- The UML provides **use case diagram** notation to illustrate the names of use cases and actors, and the relationships between them.

Use Case Diagrams





*Department of
Software Engineering*

Life-in-2050



*Department of
Software Engineering*

Use Case
Use Case Diagrams
Actors and Examples

USE CASE DESCRIPTION

Use Case Description

- Scope
- Level
- Primary Actor
- Stakeholders and Interests
- Preconditions
- Main Success Scenario
- Alternative Scenario

Check document:- Use Case UC1: Process Sale

Use Case Name:	Create new order	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk	
Related Use Cases:	Includes: <i>Check item availability</i>	
Stakeholders:	Sales department: to provide primary definition Shipping department: to verify that information content is adequate for fulfillment Marketing department: to collect customer statistics for studies of buying patterns	
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Events:	Actor	System
	1. Sales clerk answers telephone and connects to a customer. 2. Clerk verifies customer information. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item (<i>Check item availability</i> use case). 6. Clerk adds item to the order. 7. Repeat steps 4, 5, and 6 until all items are added to the order. 8. Customer indicates end of order; clerk enters end of order. 9. Customer submits payment; clerk enters amount.	3.1 Create a new order. 5.1 Display item information. 6.1 Add an order item. 8.1 Complete order. 8.2 Compute totals. 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.	

Writing Use Case Description

1. Select a use case
2. Write abbreviated *full description* (Use case name, Scenario (if any), Business Event, Actors, Flow of steps, Exception conditions)
3. For figuring Flow of steps,
 - Keep in mind general system model: Input-Processing-Output
 - Steps should be at nearly the same level of abstraction (each makes nearly same progress toward use case completion)
4. For figuring exception conditions, focus on if-then logic.



*Department of
Software Engineering*

SUM UP



*Department of
Software Engineering*

REFERENCES

References/1

Book:

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) by Craig Larman



*Department of
Software Engineering*

QUESTIONS ?



*Department of
Software Engineering*

thank you!