

Process of writing use cases

Find actors



Student

Find use cases

Register for Course

Brief description: This use case allows a Student to register for course offerings in the current semester. A Schedule of core and option courses is produced.

Outline a
use case

Register for Course Outline

- Flow of events
- Step by step

Detail a
use case

Register for Course Use-Case Specification

- Detailed Flow of Events
- Special Requirements
- Pre/Postconditions

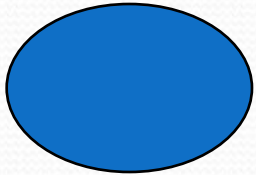
Use Case 'Simple' Description of a requirement.

<i>Use Case No:</i>	<i>Use Case Name:</i>	<i>Rating MoSCoW</i>
<i>Purpose:</i>		
<i>Main actor:</i>	<i>Secondary Actors:</i>	
<i>Description:</i> <ul style="list-style-type: none">•		

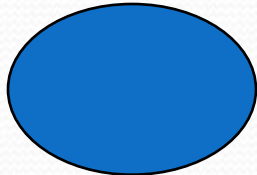
Is *Log in* a use case?

- By UML definition, *log in* is not a use case, because it does not produce results of value to an actor.

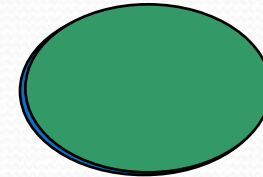
Which one is a potential use case?



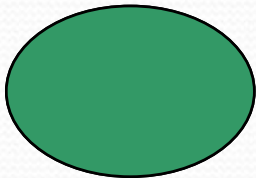
Support IE7.0



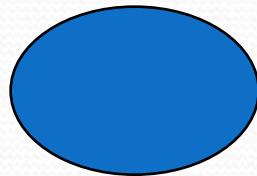
Have system 100% available
During regular business hours



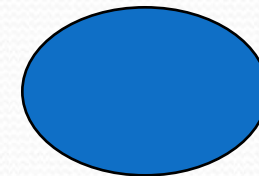
Withdraw cash from an ATM



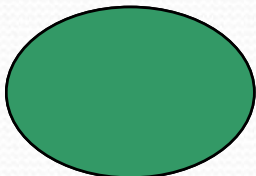
Transfer funds



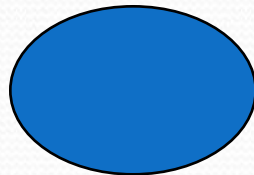
Display error messages
within 30 seconds



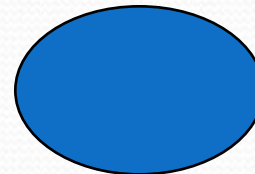
Support English and Japanese



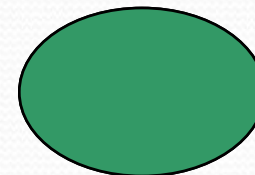
Place order



Create customer
user interface

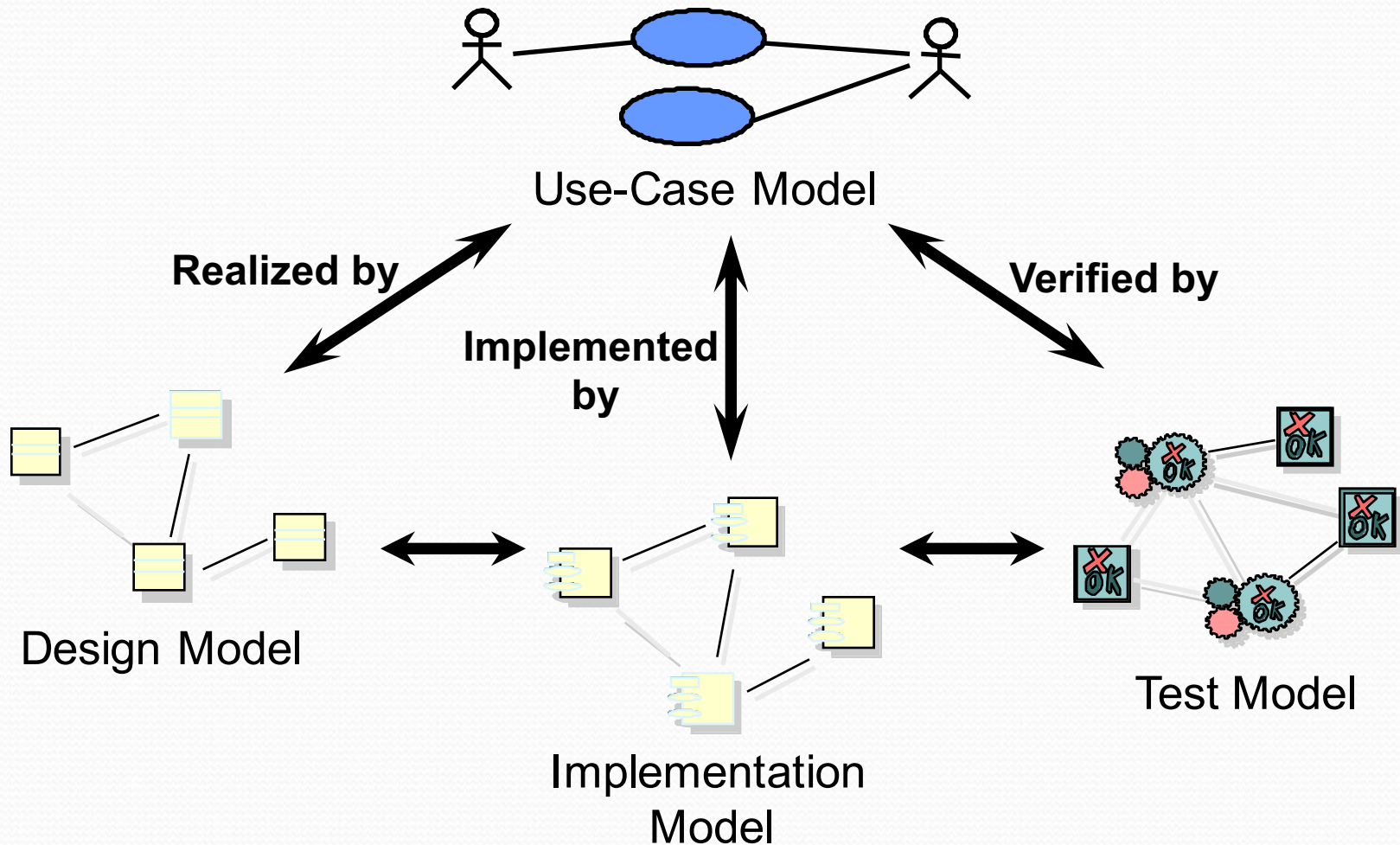


Login



Make reservation

Use cases drive software development



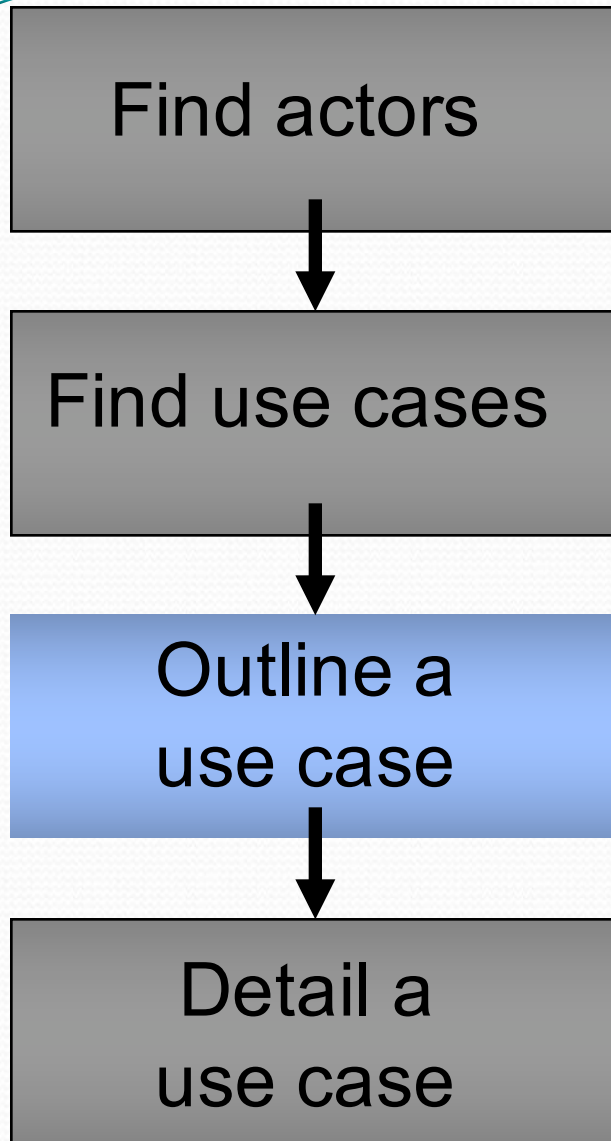
Outlining and Detailing Use Cases

- Objective

After this lecture,

- You should be able to outline and write a detailed description of a use case

Process of writing use cases



- ▶ Outline the flow of events
- ▶ Capture use-case scenarios
- ▶ Collect additional requirements

Outline each use case

Use-Case Size

Too Small?

Too Big?

- An outline captures use case steps in short sentences, organized sequentially

Is it more than one use case?

Use Case Name

Brief Description

Basic Flow

1. First step
2. Second step
3. Third step

Structure the basic flow into steps

Outlining helps find alternative flows

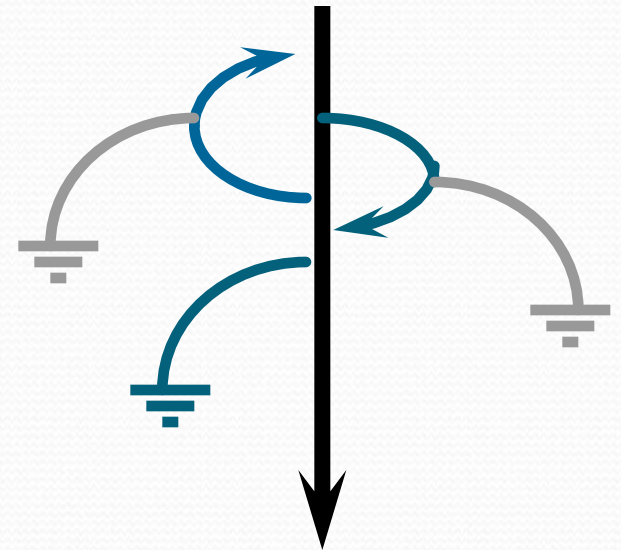
- Alternative flow 1
- Alternative flow 2
3. Alternative flow 3

Identify alternative flows

Number and name the steps

Flows of events (basic and alternative)

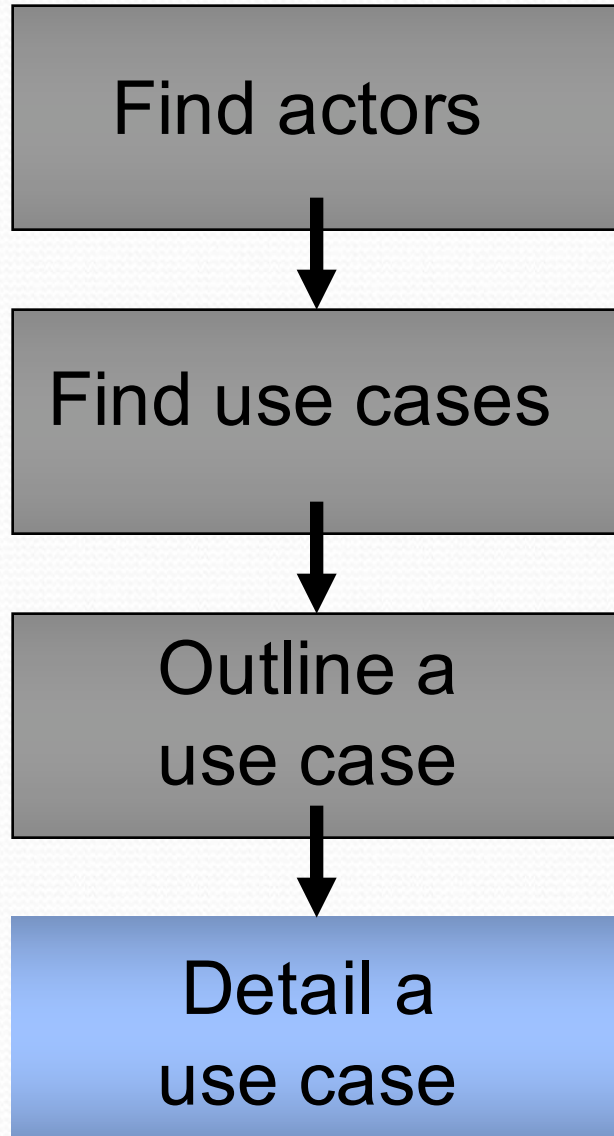
- A flow is a sequential set of steps
- One basic flow
 - Successful scenario from start to finish
- Many alternative flows
 - Regular variants
 - Odd cases
 - Exceptional (error) flows
- An ordered set of flows from the start of a use case to one of its end points



Outline the flows of events

- Basic flow
 - What event **starts** the use case?
 - How does the use case **end**?
 - How does the use case **repeat** some behavior?
- Alternative flows
 - Are there **optional** situations in the use case?
 - What **odd** cases might happen?
 - What **variants** might happen?
 - What may **go wrong**?

Process of writing use cases



- ▶ Detail the flow of events
- ▶ Structure the flow of events
- ▶ Specify additional use case properties

Detail the basic flow of events

Structure the flow into steps

Number and title each step

Describe the steps

Register for Courses

1.1 Basic Flow

1. **Registration open.**

This use case starts when someone chooses to register for courses. The system checks that registration is possible at this time.

2. **Select “Create a Schedule ”.**

The system displays the functions available to the student. The student selects “Create a Schedule ”.

3. **Obtain Course Information.**

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the student .The student can search the list by department, professor, or topic to obtain the desired course information .

4. **Select Courses.**

The student selects four primary course offerings and two alternate course offerings from the list of available offerings course offerings.

...

Phrasing of steps

- Use the active voice
 - Say: “The Professor provides the grades for each student”
 - Instead of: “When the Professor has provided the grades”
- Say what triggers the step
 - Say: “The use case starts when the Professor chooses to submit grades”
- Say who is doing what (use the Actor name)
 - Say: “The Student chooses ...”
 - Instead of: “The user chooses ...”
 - Say: “The System validates ...”
 - Instead of: “The choice is validated ...”

For Alternative flows

Cross-referencing using a label

Register for Course

2. Flow of Events

2.1 Basic Flow

1. Registration open

The system checks that a student is able to register for a course at this time of the academic year.

2. Select "Create a Schedule."

The system displays the functions available to the student. The student selects "Create a Schedule."

3. Obtain Course Information]

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the student....

The student selects four primary course offerings and two alternate course offerings from the list of available offerings course offerings.

4. Submit Schedule.

The student indicates that the schedule is complete. For each selected course offering on the schedule, the system

5. Accept Completed Schedule.

The system displays the completed schedule containing the selected courses. The confirmation.....

2.2 Alternative Flows

2.2.1 Registration not open

In step 1, Registration open in the Basic Flow, if the system determines students can't register for courses at this time, then a message is displayed. The use case ends.

2.2.2 Quit

The Course Registration System allows the student to quit at any time during the use case. The student may choose to save a partial schedule before quitting. All courses that are not marked as "enrolled in" are

In the Registration open step of the Basic Flow,

Detail of Alternative Flows

Describe what happens

Alternative Flows

2.2.1 Registration not open

In the Registration Open step of the Basic Flow, if the system determines students can't register for courses at this time, then a message is displayed, and the use case ends.

Location

Condition

2.2.2 Quit and Save.

At any time the system will allow the Student to quit. The student chooses to quit and save a partial schedule before quitting. The system saves the schedule, and the use case ends.

Actions

Detailed Use Case Template (Structured Natural Language)

Use Case No:	Use Case Name:	Rating: MoSCoW		
Purpose:				
Main actor:		Secondary Actors:		
Pre-conditions:				
Trigger:				
Description:				
Extension:				
Related Use Cases: 'includes'				
Post-conditions:				
Author:	Date:	Approved:	Date:	Version:

Detailed Use Case Template (Structured Natural Language)

Use Case No:	Use Case Name:	Rating: MoSCoW		
Purpose:				
Main actor:		Secondary Actors:		
Pre-conditions:				
Trigger: Passenger arrives at Check-in Desk				
Description:				
Extension:				
Related Use Cases: 'includes'				
Post-conditions:				
Author:	Date:	Approved:	Date:	Version:

Preconditions

- Describe the state that the system must be in before the use case can start
 - Simple statements that define the state of the system, expressed as **conditions that must be true**
 - Do **not** refer to other use cases that need to be performed prior to this use case
 - Should be stated clearly and should be easily verifiable

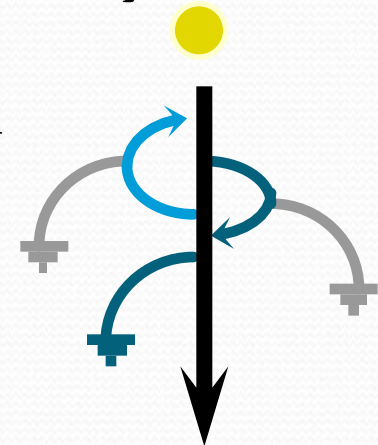
- Optional: Use only if needed for clarification

- Example

Register for Courses use case

Precondition:

- The list of course offerings for the semester has been created and is available to the Course Registration System
- Student has logged into the Course Registration System



Detailed Use Case Template (Structured Natural Language)

Use Case No:	Use Case Name:	Rating: MoSCoW		
Purpose:				
Main actor:		Secondary Actors:		
Pre-conditions:				
Trigger: Passenger arrives at Check-in Desk				
Description:				
Extension:				
Related Use Cases: 'includes'				
Post-conditions:				
Author:	Date:	Approved:	Date:	Version:

Postconditions

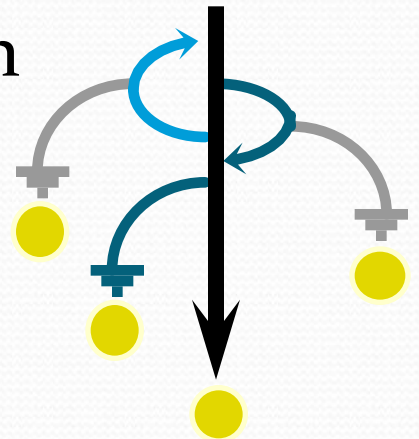
- Describe the state of the system at the end of the use case
 - Use when the system state is a **precondition** to another use case, or when the possible use case outcomes are not obvious to use case readers
 - Should **never refer to other**, subsequent use cases
 - Should be stated clearly and should be easily verifiable

Optional: Use only if needed for clarification

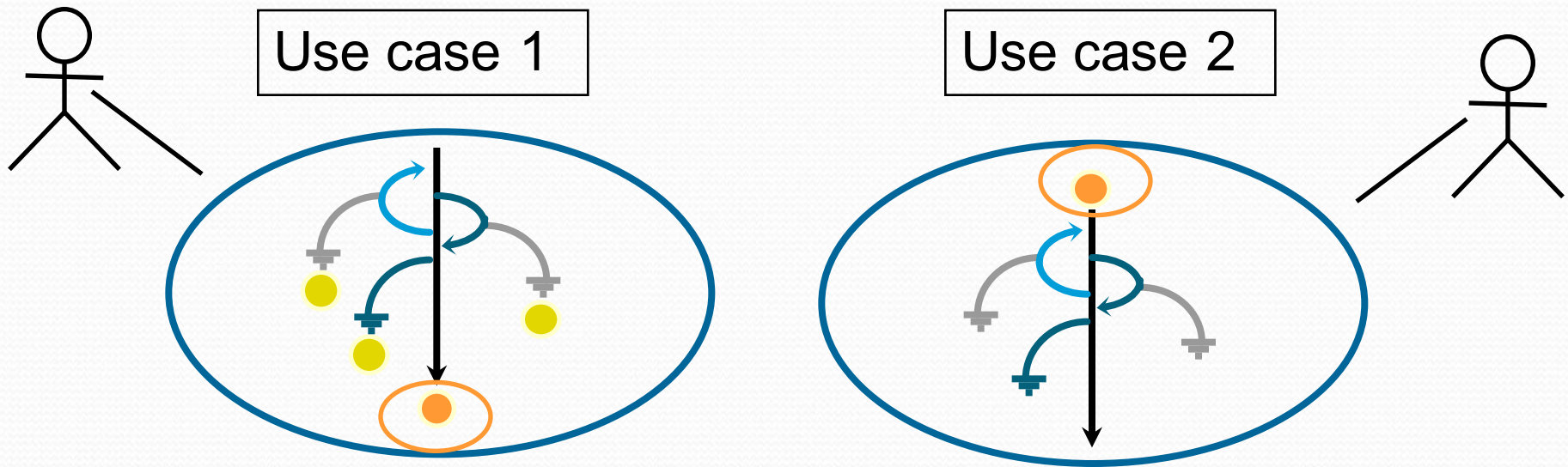
Example:

Register for Courses use case

Postcondition: At the end of this use case either the student has been enrolled in courses, or registering was unsuccessful and no changes have been made to the student schedules or course enrollments



Sequence use cases with pre- and postconditions

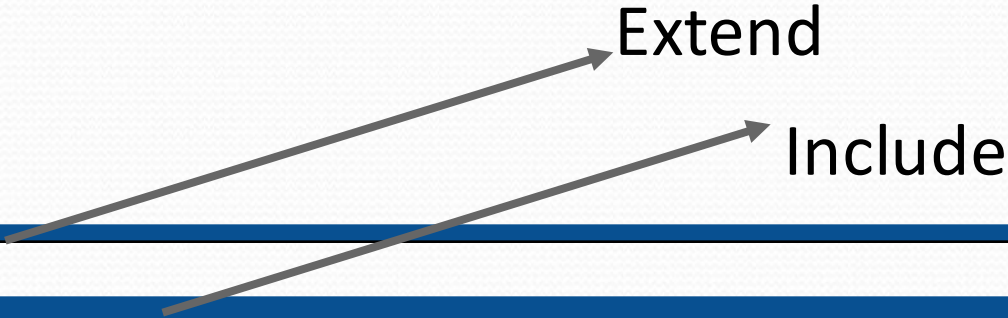


Use cases do **not** interact with each other.
However, a postcondition for one use case
can be the same as the precondition for another.

Use case checkpoints

- ✓ The actor interactions and exchanged information is clear
- ✓ The communication sequence between actor and use case conforms to the user's expectations
- ✓ How and when the use case's flow of events starts and ends is clear
- ✓ The basic flow achieves an observable result for one or more actors

Detailed Use Case Template (Structured Natural Language)

Use Case No:	Use Case Name:	Rating: MoSCoW		
Purpose:				
Main actor:		Secondary Actors:		
Pre-conditions:				
Trigger: Passenger arrives at Check-in Desk				
Description:				
				
Extension:				
Related Use Cases: 'includes'				
Post-conditions:				
Author:	Date:	Approved:	Date:	Version:

Final remarks

- Use the template on the previous slide for your coursework
- We will return to UML later in the module to look at Class Diagrams.