

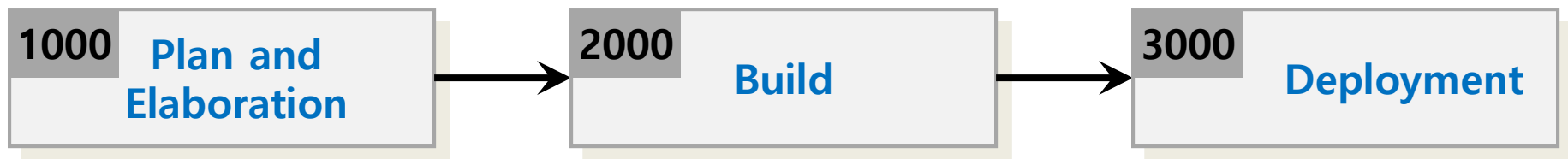
# Software Modeling & Analysis

OOPT (Object Oriented Process with Trace)

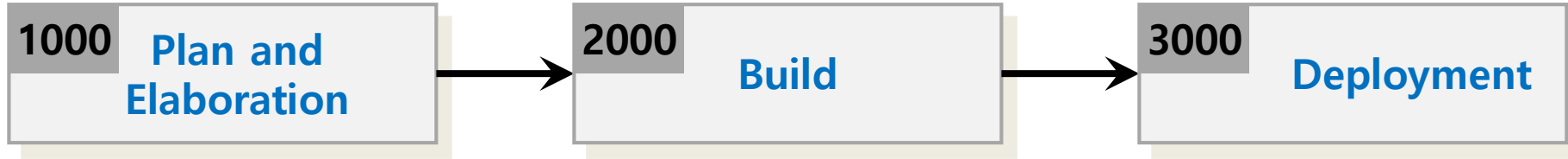
Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

# What is OOPT?

- OOPT (Object Oriented Process with Trace)
  - A software process based on RUP
  - Revision of OSP (by Tailored to SE classes in universities)
- Characteristics of OOPT
- 3 Stages
  1. Iterative : Multiple development cycles
  2. Incremental : System grows incrementally as each cycle is completed
  3. Architecture : Stage > Cycle > Phase > Activity



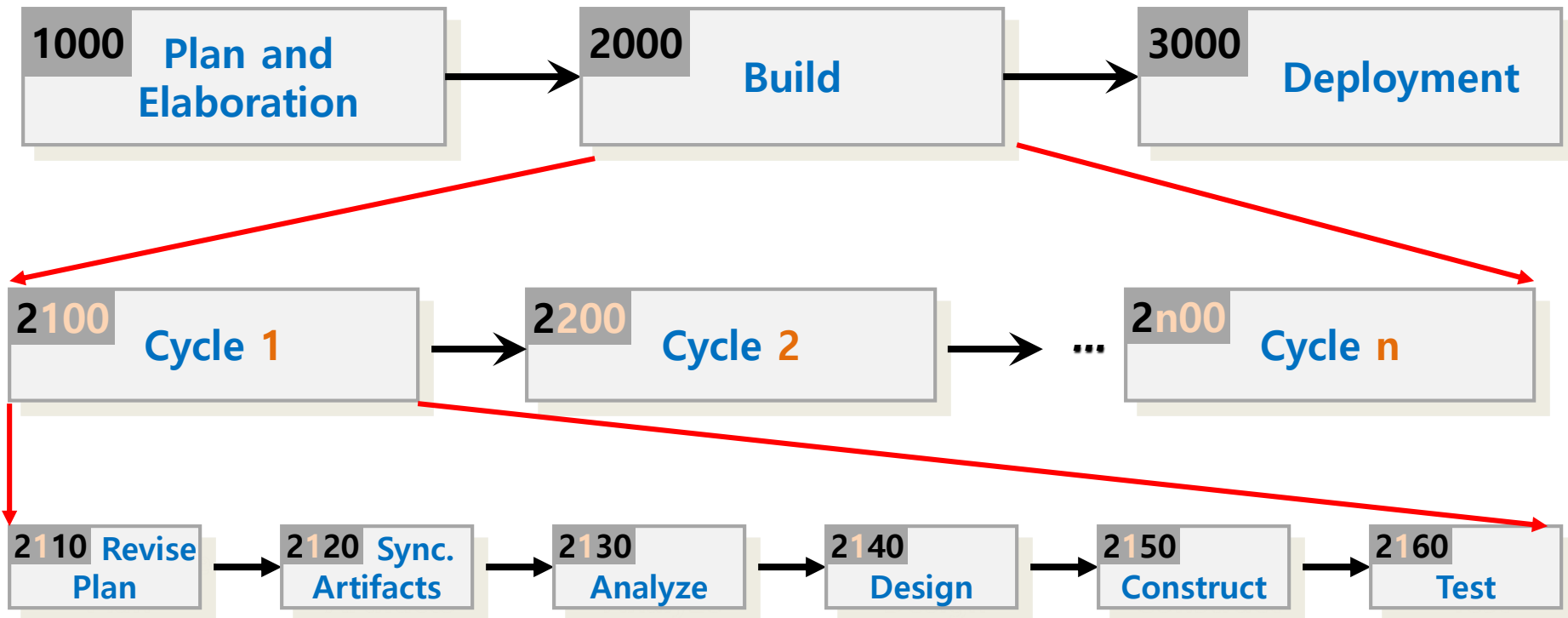
# 1. 3 Stages



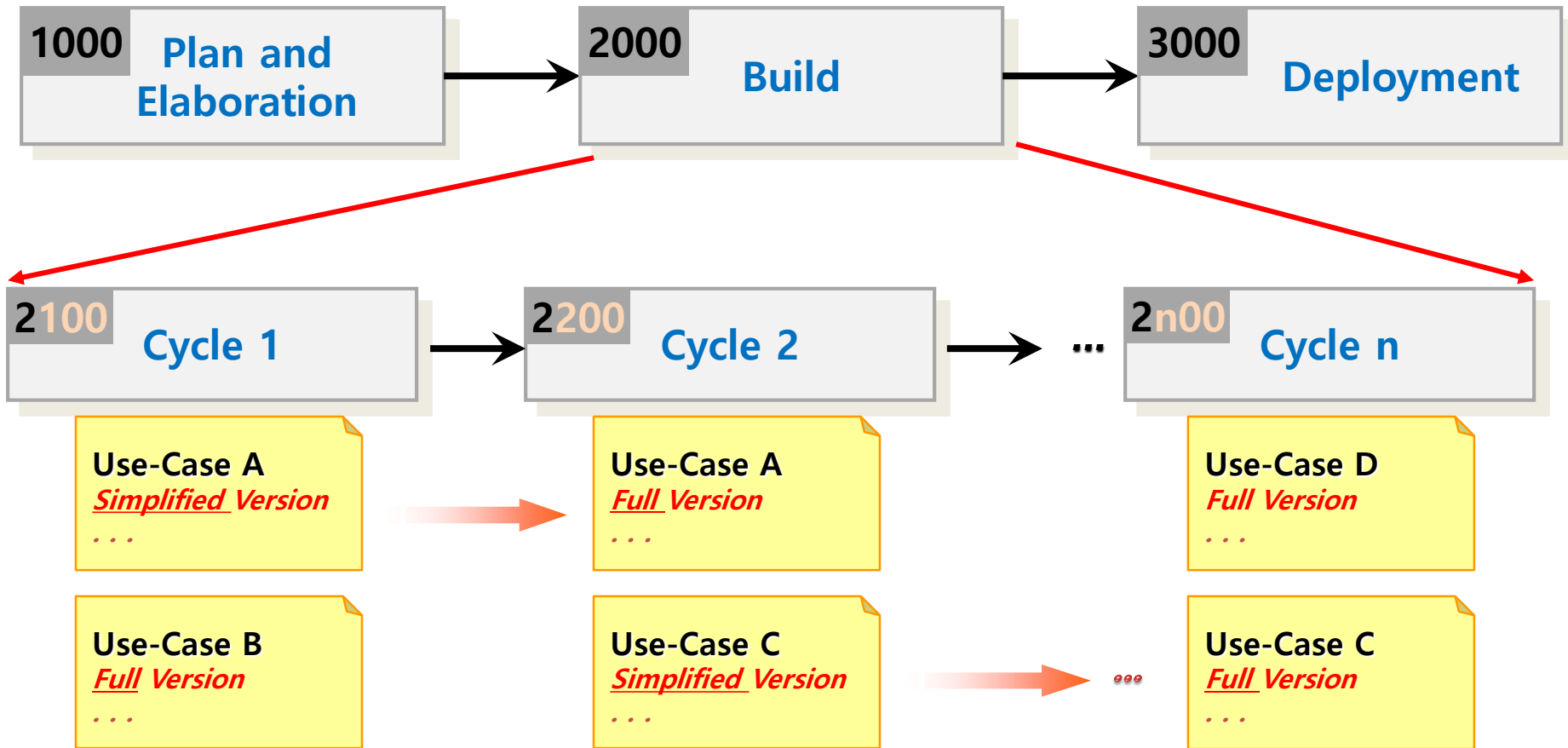
- Stage 1000 : Plan and Elaboration
  - Planning, defining requirements, building prototyping, etc
  - Corresponding to Inception/Elaboration phases in the RUP
- Stage 2000 : Build
  - Construction of the system
  - Corresponding to Construct phase in the RUP
- Stage 3000 : Deployment
  - Implementation of the system into use
  - Corresponding to Transition phase in the RUP

# 2. Iterative Development

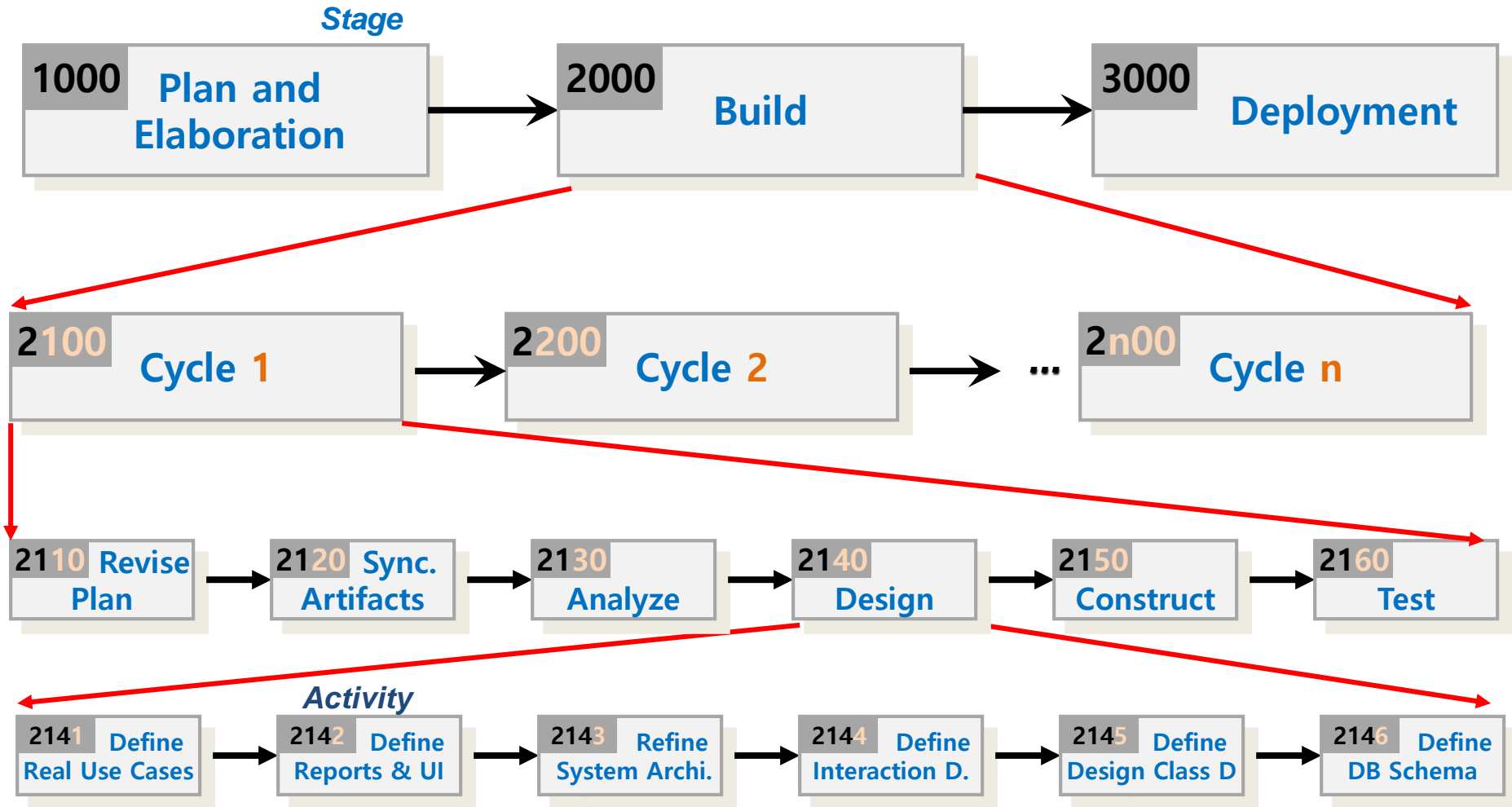
- Multiple iterations in the Build stage
- Each iteration took about 2 to 8 weeks



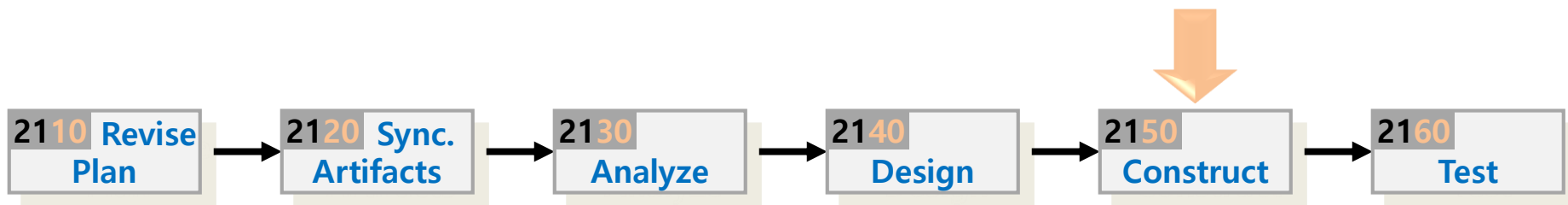
# 3. Incremental Development



# 4. Architecture of OSP

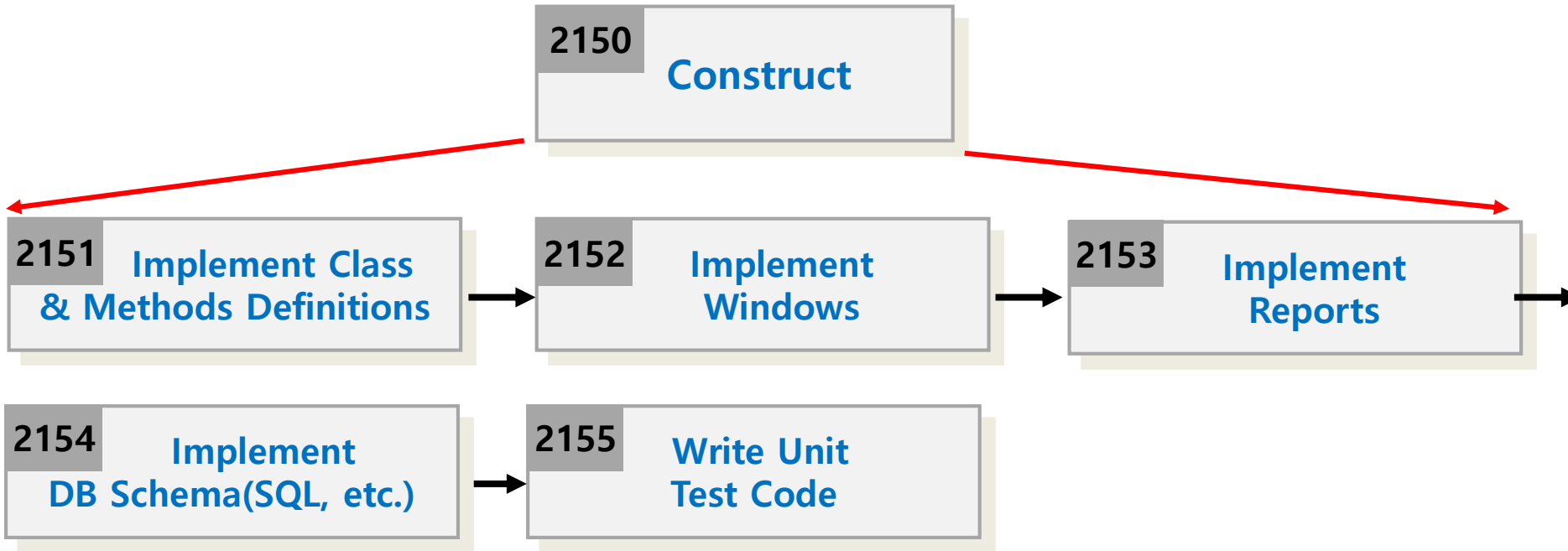


# Phase 2050. Construct



# Phase 2050. Construct

- Phase 2050 Activities





# Activity 2051.

## Implement Class & Methods Definitions

### 2151 Implement Class & Methods Definitions

- Description
  - Implement class & methods in accordance with design class diagram
  - Describes the description of class and methods
    - It is used to design unit test cases
  - Input : Design class diagram, real use cases, Interaction diagram
  - Output : Class & Methods description
- Steps:
  1. Identify all classes and methods in design class diagram(2045)
  2. Writing description of the target by using below format
  3. Implement

# Activity 2051.

## Implement Class & Methods Definitions

<b>Type</b>	Class or Method
<b>Name</b>	The name of class o method
<b>Purpose</b>	Writing the purpose of class or method
<b>Overview (Class)</b>	Information of class or method
<b>Cross Reference</b>	Functions: Writing related system functions Use Cases : Writing related use cases
<b>Input (Method)</b>	Writing all of inputs of methods
<b>Output (Method)</b>	Writing the output of methods
<b>Abstract operation (Method)</b>	Writing operation information of methods briefly
<b>Exceptional Courses of Events</b>	Writing exception state and handling information

# Activity 2052. Implements Windows

2152
Implement  
Windows

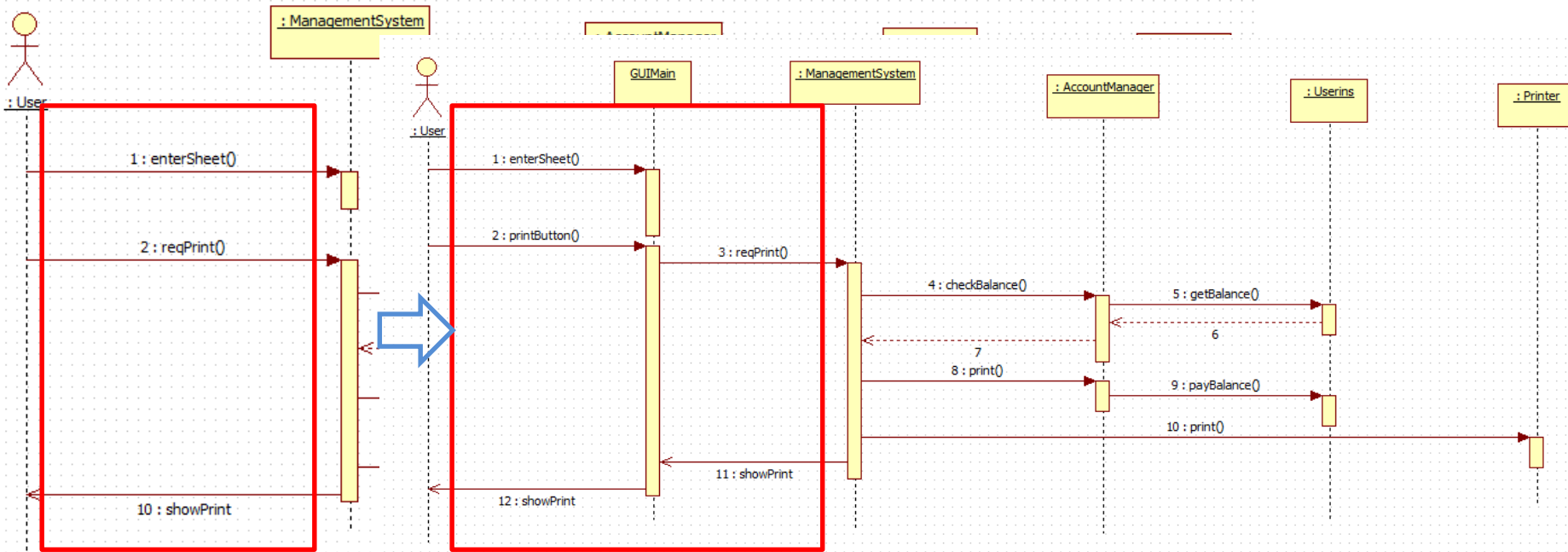
- Description
  - Define relations between GUI and operation
    - Implements GUI class between actor and system
  - Input : Interaction diagram, design class diagram, real use case description, UI Story board, operation contracts
  - Output : GUI implements results and description, Refined Class Diagram
- Steps:
  1. Define GUI system operation between Actor and System in interaction diagram
  2. Draw the refined interaction diagram by adding GUI operation
  3. Draw up the description (using below format) of GUI operation in diagram

# Activity 2052. Implements Windows

Name	A name of operation
Responsibilities	Writing the paper number to text box
Type	GUI
Cross References	R 2.1
Notes	Shows the value of text box in screen
Pre-Conditions	User should be logged in before
Post-Conditions	Paper text box shows the input value

# Activity 2052. Implements Windows

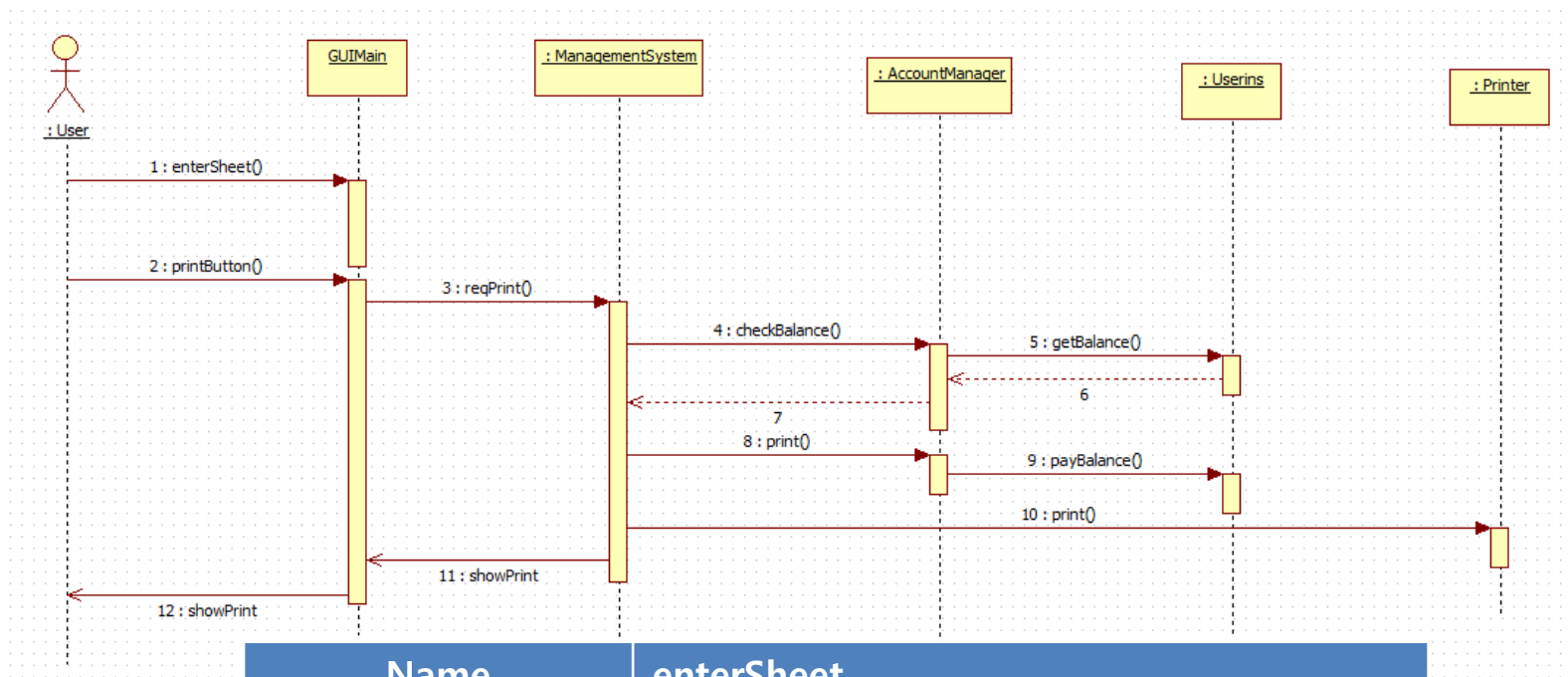
- An example of operations in Request Print use case



enterSheet	(text input with only GUI)	Text Input
reqPrint		Connection with printButton
showPrint	Print message	Connection with showPrint

# Activity 2052. Implements Windows

- Request Print use case



Name	enterSheet
Responsibilities	Text box 에 인쇄 매수를 입력 한다.
Type	GUI
Cross References	R 2.1
Notes	Text box 에 입력한 숫자를 화면이 표시한다.
Pre-Conditions	유저 로그인 상태
Post-Conditions	Paper text box 에 입력 값 표시

# Activity 2053. Implement Reports



2153 **Implement Reports**

- Description
  - Reports which have information of analysis, design results is refined and implemented in this process again
  - Input : All of information
  - Output : Analysis, design reports

# Activity 2055. Write Unit Test Code

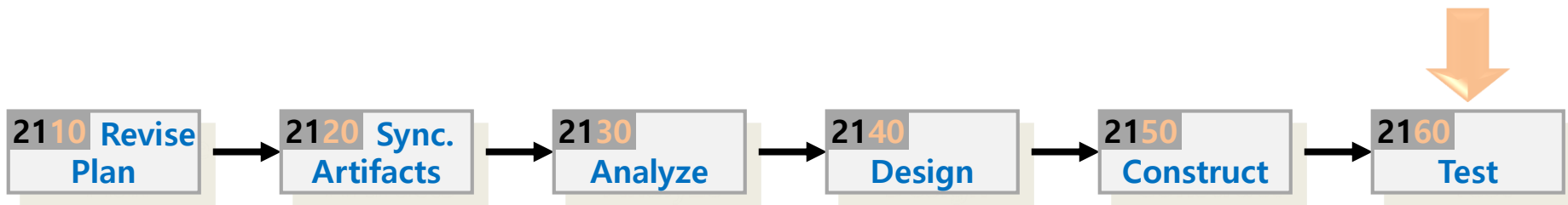
2153

Write  
Test Code

- Description
  - Writing test code for performing unit testing
  - Input : Implements results, class & methods definitions
  - Output : Unit test code
- Steps:
  1. Identify all methods in design
  2. Writing unit test code accordance with the definition of methods

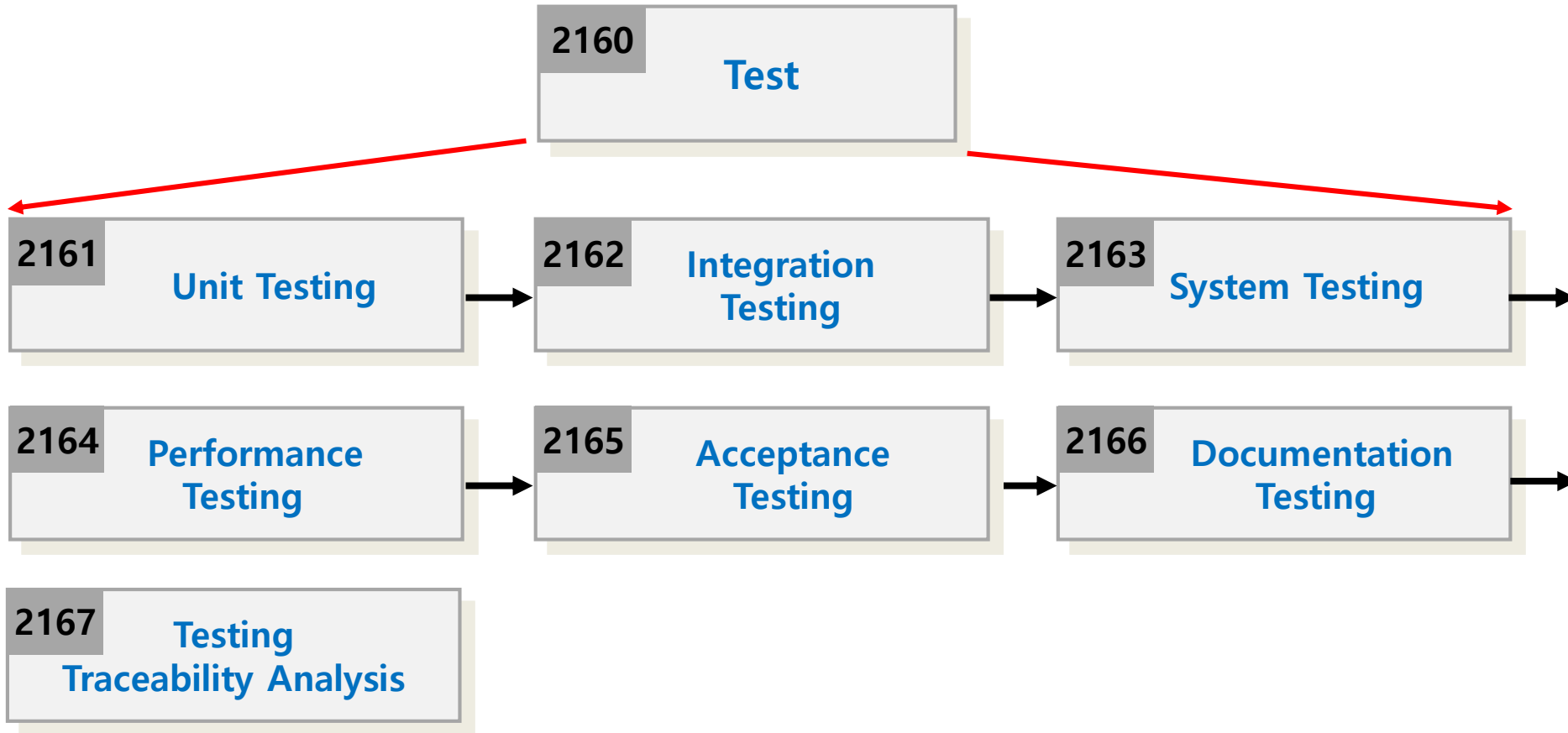


# Phase 2060. Test



# Phase 2060. Test

- Phase 2060 Activities



# Activity 2061. Unit Testing

2161

Unit Testing

- Description
  - unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.
  - Perform unit testing and identify the results of testing
  - Input : Unit test code, implement results
  - Output : Unit testing results, reports
- Steps:
  1. Performing unit test with test codes and implement results of each methods (JUnit)

# Activity 2062. Integration Testing



2162 Integration Testing

- Description
  - Integration testing is the phase in software testing in which individual software modules are combined and tested as a group
  - Input : Class & Method definitions
  - Output : Integration testing results, reports

# Activity 2063. System Testing

2163

## System Testing

- Description
  - System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
  - Input : Implements results, system test plan and cases
  - Output : System testing results, reports
- Steps:
  1. Identify system test cases before defined
  2. Set the test data of test cases for testing
  3. Performing system testing with system test plan and cases (JFeature)

# Activity 2063. System Testing

- Set the test data of test cases for testing
  - Example of test case 1-1 & 1-2

1 - 1	로그인 시험	존재하는 계정의 Id, pw를 입력하고 로그인 시도를 하여 로그인 기능 test	1. Login	R 1.1
1 - 2	로그인 시험	존재하지 않는 계정의 Id, pw를 입력하고 로그인 시도를 하여 로그인 기능 test	1. Login	R 1.1

- 1-1 : id 입력란에 test, pw 입력란에 test 입력 후 login 버튼을 누른다.
- 1-2 : id 입력란에 test2, pw 입력란에 asdf 입력 후 login 버튼을 누른다.

# Activity 2064. Performance Testing

2164
Performance  
Testing

- Description
  - Performance testing is in general, a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload.
  - Input : Implements result (program)
  - Output : Performance testing results, reports

# Activity 2065. Acceptance Testing

2165
Acceptance  
Testing

- Description
  - Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met
  - It is used to determine the final acceptance
  - Input : Requirements specification, system
  - Output : Acceptance testing results, reports



# Activity 2066. Documentation Testing

2166
Documentation  
Testing

- Description
  - Documentation testing is a type of non-functional testing
  - It intends to check the quality of documentation
  - Input : All documents in process
  - Output : Documentation testing results, reports

# Activity 2066. Testing Traceability Analysis

2167
Testing  
Traceability Analysis

- Description
  - Identify the relations between design and test cases
  - Input : Testing results, traceability documents
  - Output : Testing – design traceability reports
  
- Step
  1. Identify system test cases and unit test cases
  2. Identify the relations between requirements and system test cases
    - System test cases should be covered all requirement specification
  3. Identify the relations between methods (component) and unit test cases