

# Software Modeling & Analysis

## OSP Summary

Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

# What is OSP?

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



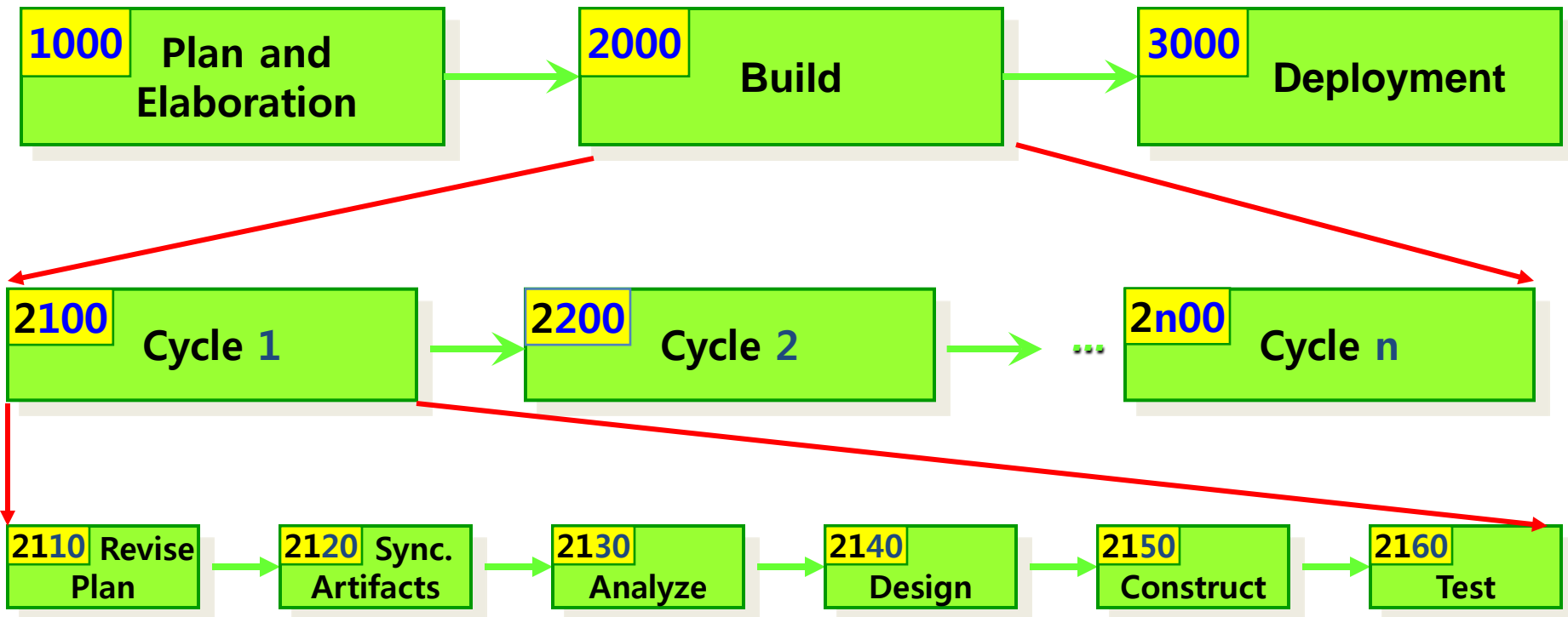
# 1. 3 Stages



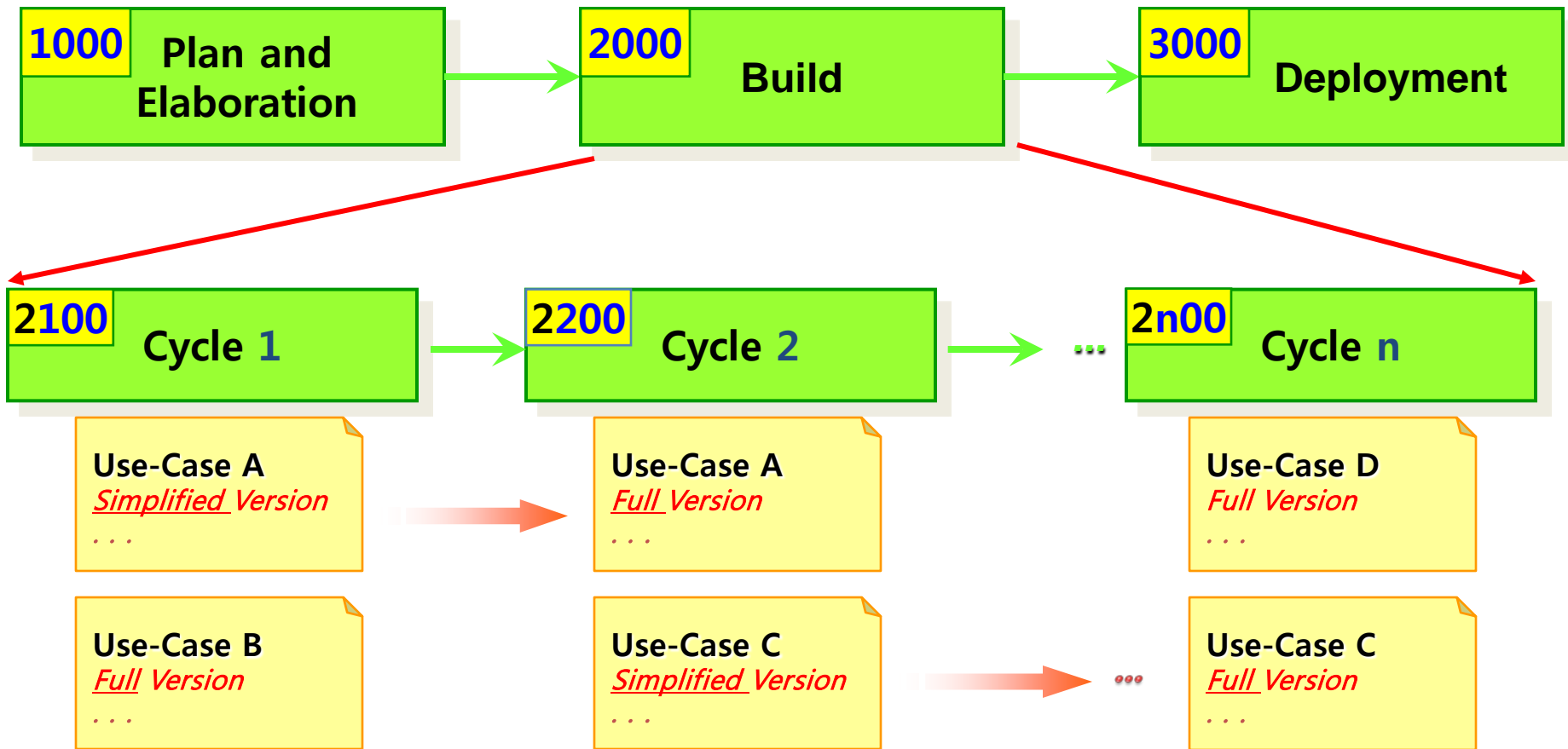
- Stage 1000 : Plan and Elaboration
  - Planning, defining requirements, building prototyping, etc
  - Corresponding to Inception/Elaboration phases in the RUP
- Stage 2000 : Build
  - Construction of the system
  - Corresponding to Construct phase in the RUP
- Stage 3000 : Deployment
  - Implementation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

# 2. Iterative Development

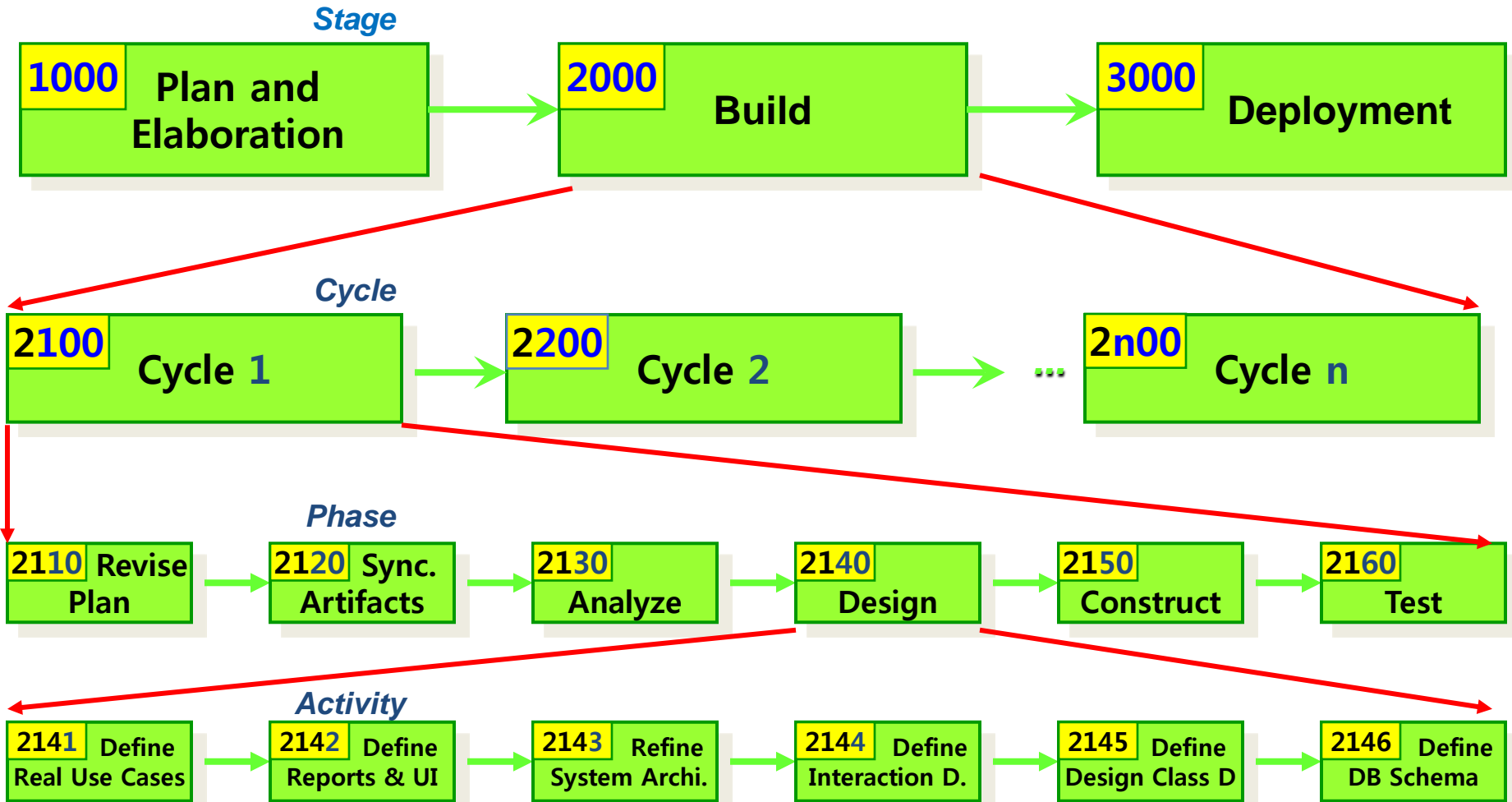
- Multiple iterations in the Build stage
- Each iteration took about 2 to 8 weeks



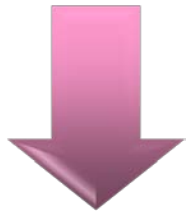
# 3. Incremental Development



# 4. Architecture of OSP



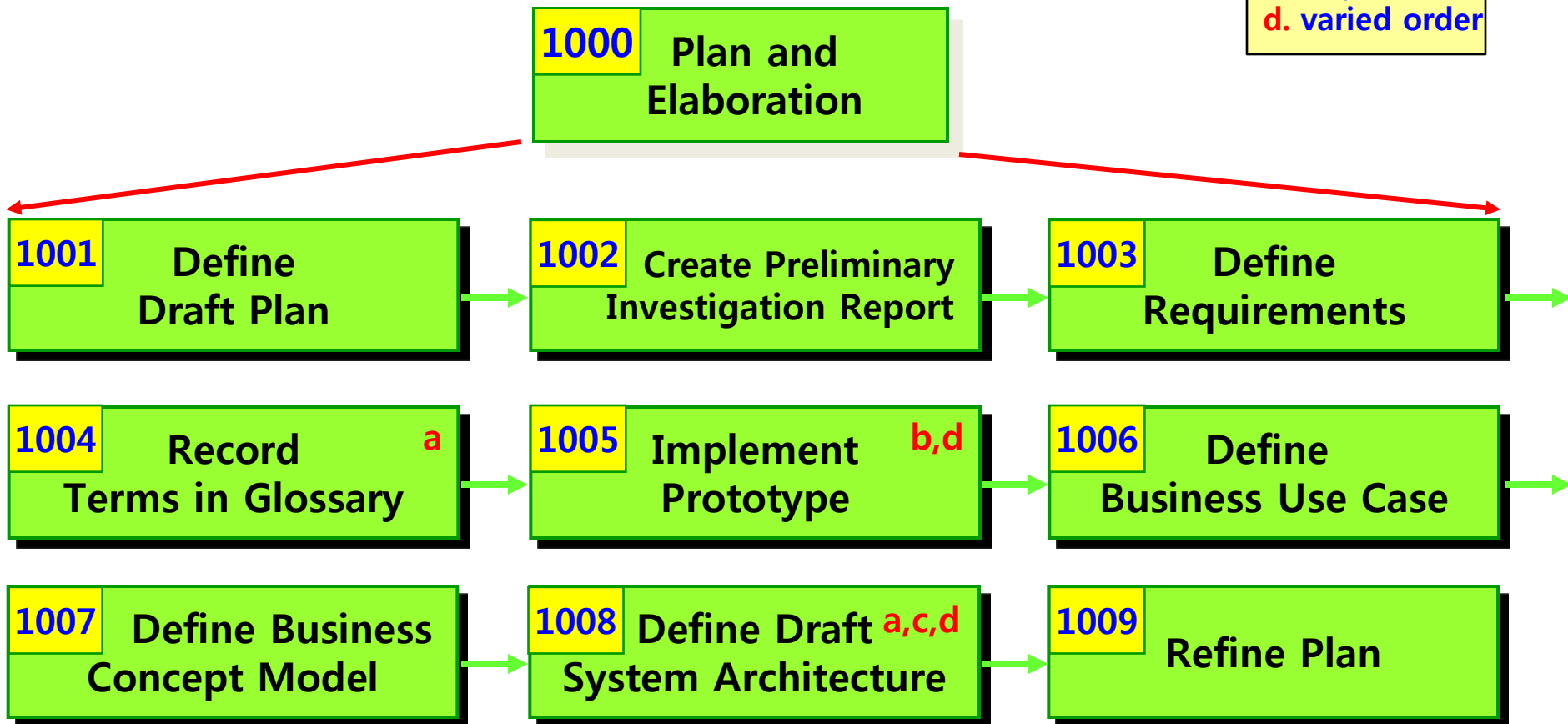
# Stage 1000. Plan and Elaboration



# Stage 1000. Plan and Elaboration

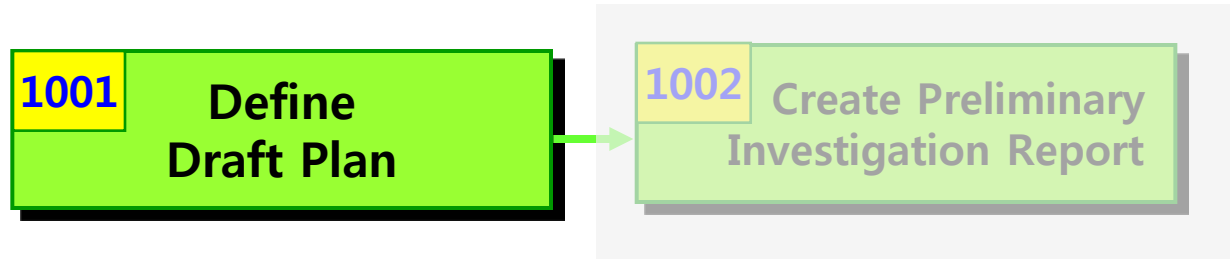
- Stage 1000 Activities

a. ongoing  
b. optional  
c. may defer  
d. varied order





# Activity 1001. Define Draft Plan



- Description
  - Write a draft plan for schedule, resources, budget, objective, etc
  - Input : related documents of previous similar projects
  - Output : a draft project plan
- Steps
  1. Write motivation and objective of project
  2. Write scope of project
  3. Identify and write functional requirements
  4. Identify and write non-functional requirements
  5. Estimate resources (human efforts(M/M), human resources, duration, budget)

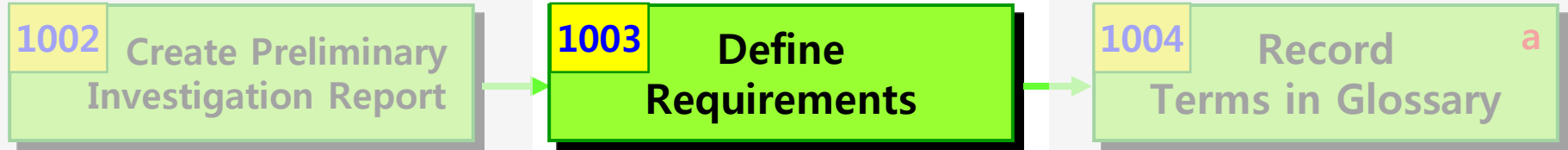
# Activity 1002.

## Create Preliminary Investigation Report



- Description
  - Write an investigation report on alternatives, business needs, risk, etc
  - Input : draft project plan
  - Output : an investigation report
- Steps
  1. Write alternative solutions
  2. Write project's justification (business needs)
  3. Identify and manage risks, and write risk reduction plans
  4. Analyze business market
  5. Write managerial issues

# Activity 1003. Define Requirements



- Description
  - Write a requirement specification for a product
  - Input : draft project plan, investigation report
  - Output : a requirement specification
- Functional requirements
- Non-functional requirements
- Recommended reference : IEEE Std. 830-1998

# Activity 1004. Record Terms in Glossary



- Description
  - Similar to “Data Dictionary”
  - Dictionary of terms and any associated information(constraints and rules)
  - Input : requirements specification
  - Output : a term dictionary
- Steps
  1. Describe meaning of terms specified in requirements specification
  2. Write alias of each term

# Activity 1005. Implement Prototype



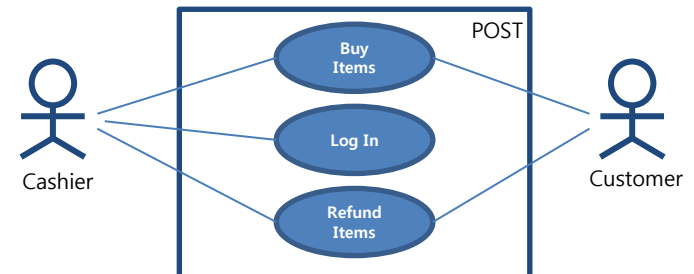
- Description
  - Develop a prototype system to permit use feedback, determine feasibility, or investigate timing or other issues
  - Input : requirements specification
  - Output : a prototype
- Steps
  1. Develop a prototype

# Activity 1006. Define Business Use Case



- Description

- To obtain a deeper understanding of the processes and requirements identified so far
- Identify business tasks as business use cases, and illustrate their relationships in use case diagrams
- Input : requirements specification
- Output : a business use case model (High-level use case)
  - Business Use Case Diagram
  - Business Use Case Description



# Activity 1007.

## Define Business Concept Model



- Description
  - Identify “business concept” in the target domain which can be candidates for “classes”
  - Input : requirements specification  
term dictionary  
business use case model
  - Output : a business concept model
- Steps
  1. Identify business terms or business concepts from requirements specification or through interviews with domain experts
  2. Define identified terms as business concepts
    - Implementation details can't be business concepts

# Activity 1008.

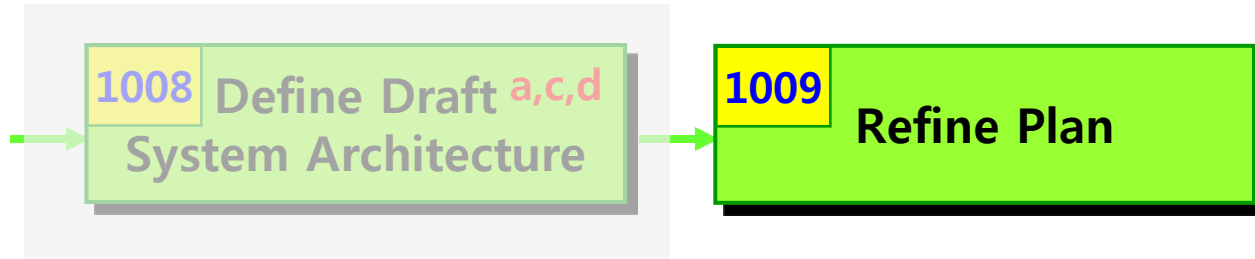
## Define Draft System Architecture



- Description
  - Construct a rough preliminary system architecture model
  - Input : requirements specification  
business use case model
  - Output : a draft system architecture
- Steps
  1. Define logical/physical layers of the target system
  2. Separate the whole system into several subsystems
  3. Assign business use cases into each subsystem
  4. Identify and draw up hardware resources

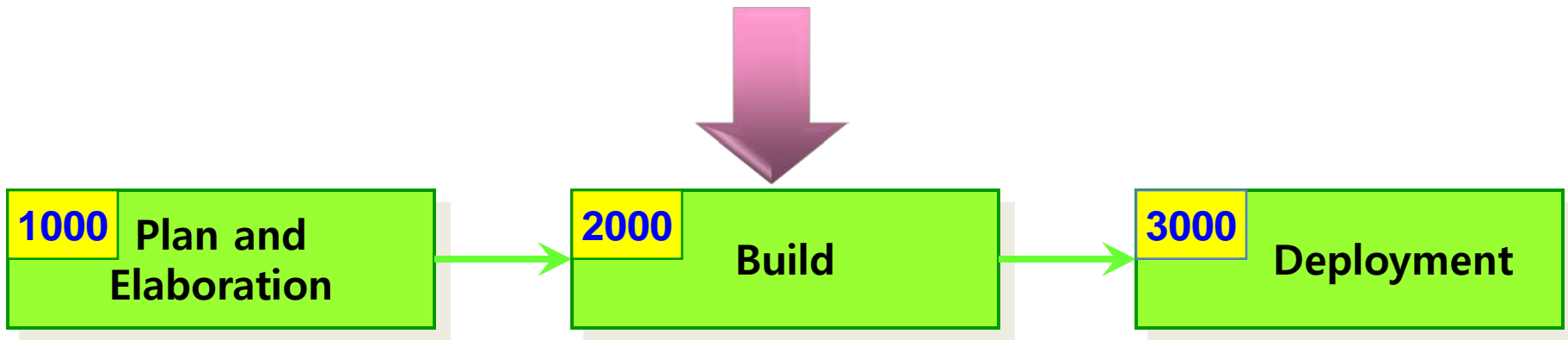


# Activity 1009. Refine Plan

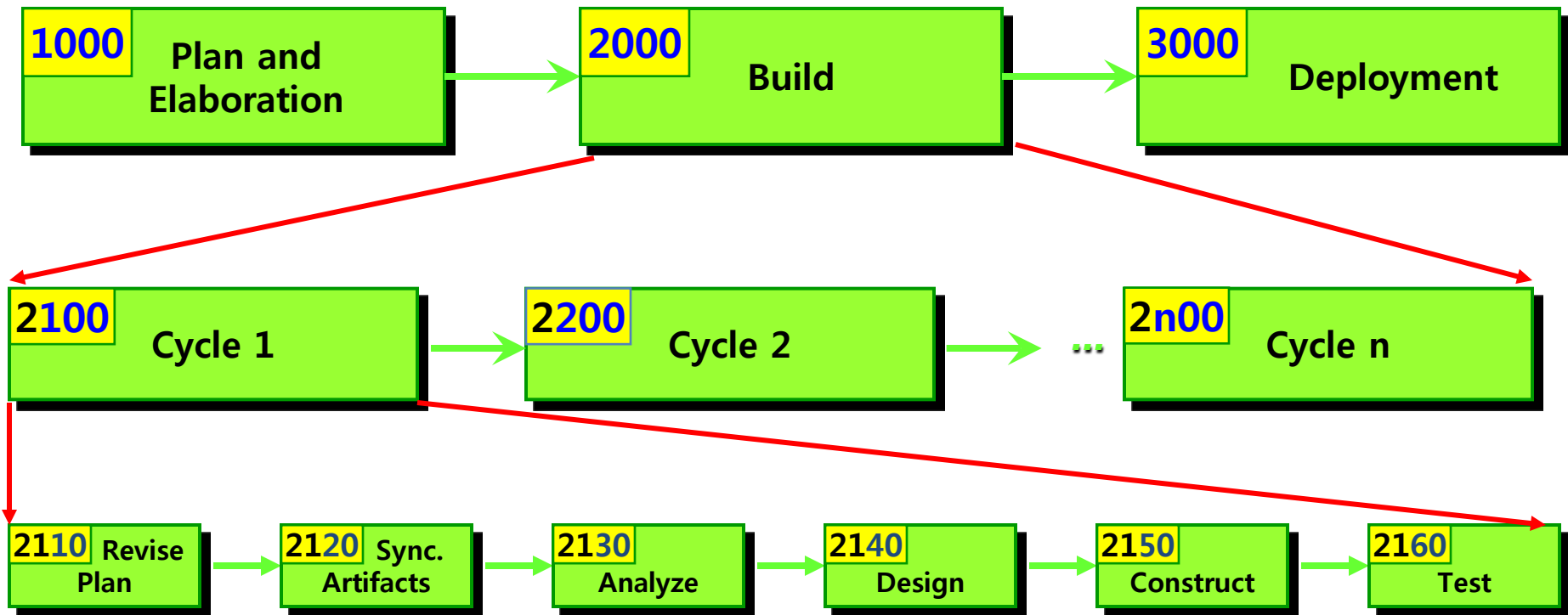


- Description
  - Refine the draft project plan generated in activity 1001
  - Input : all outputs of OSP stage 1000
  - Output: a refined project report
- Steps
  1. Review draft project plan, based on requirements specification, business use case model, business concept model, and draft system architecture
  2. Refine project's scope, duration, cost, and other resources

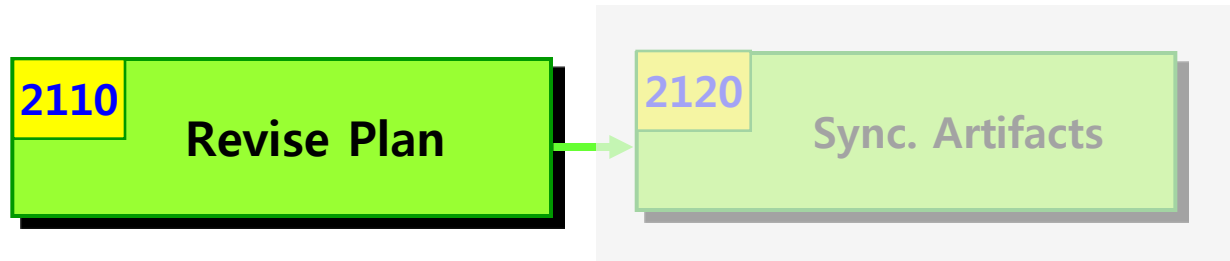
# Stage 2000. Build



# 6 Phases of 'Build' Stage



# Phase 2010. Revise Plan



- Description
  - Correct and enhance the project plan and requirement definition based on the intermediate deliverables
  - Input : intermediate deliverables
  - Output : A refined project plan, a refined requirement specification
- Steps

# Phase 2020. Synchronize Artifacts

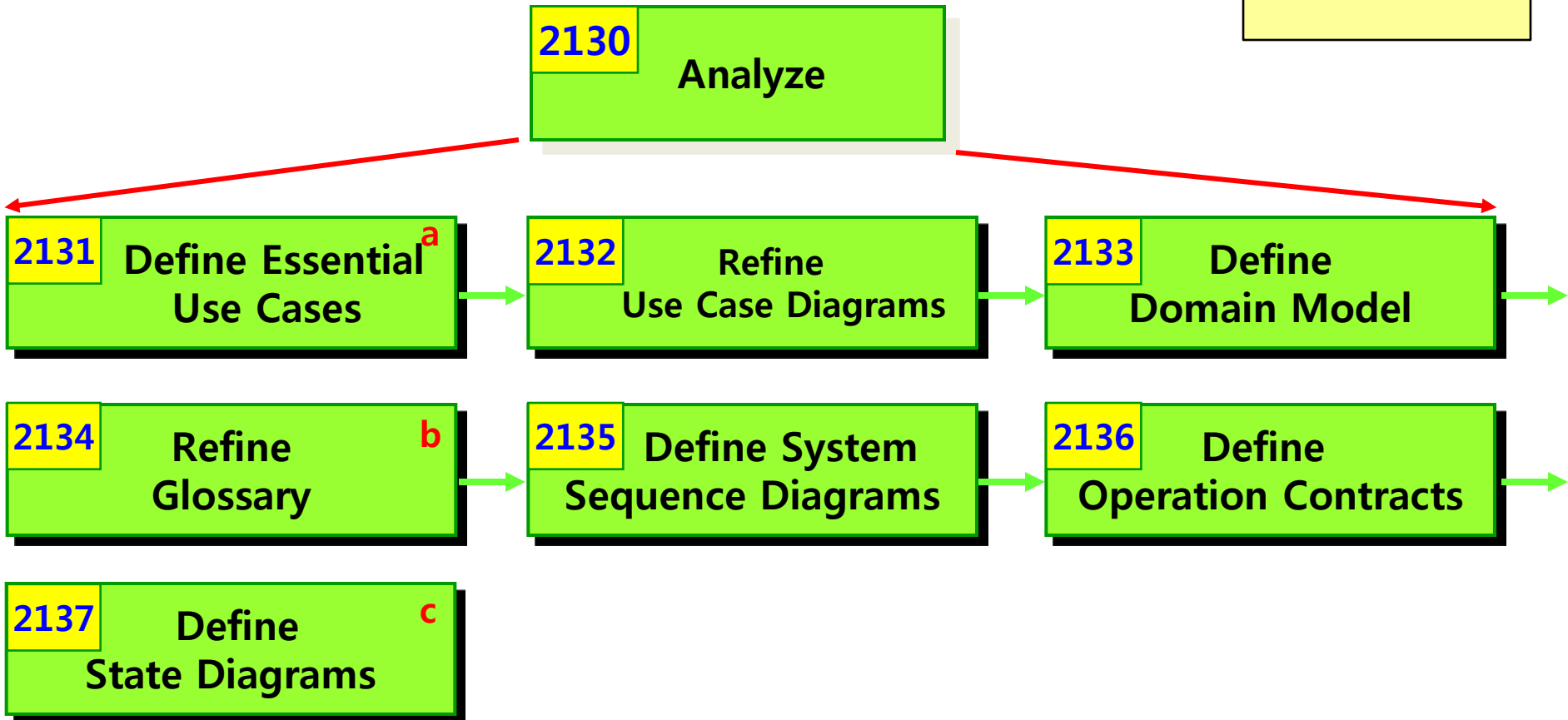


- Description
  - Configure and manage various types of artifacts (Project Repository)
  - Control versions and variations
  - Input :
  - Output :
- Steps

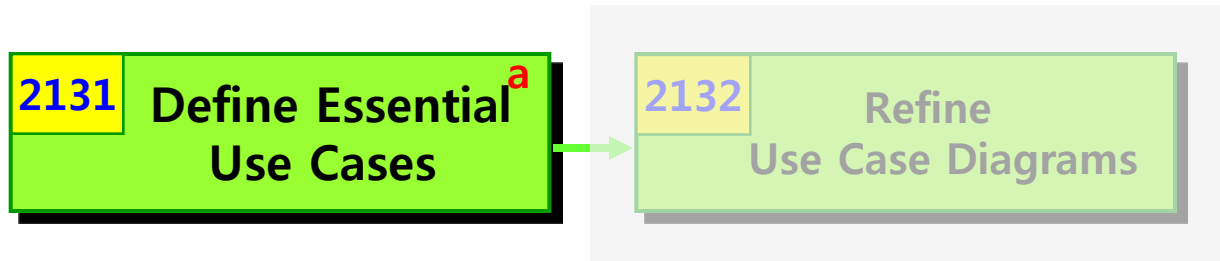
# Phase 2030. Analyze

- Phase 2030 Activities

a. if not yet done  
b. ongoing  
c. optional



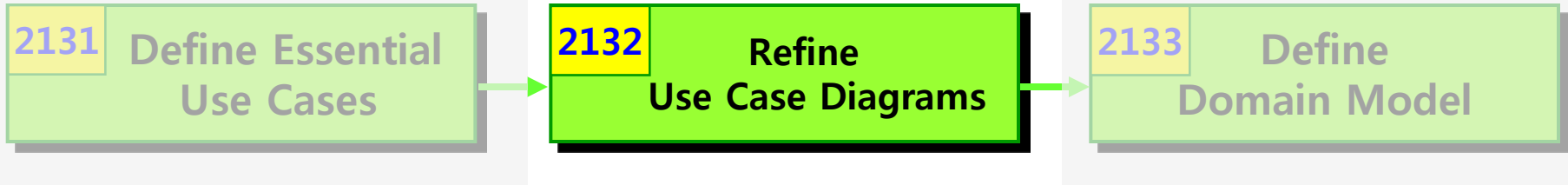
# Activity 2031. Define Essential Use Cases



- Description
  - Add event flows to business use case(high-level) descriptions
  - Input : business use case descriptions (activity 1006)
  - Output : An essential use case descriptions

<b>Use Case</b>	Buy Items
<b>Actor</b>	Customer, Cashier
<b>Purpose</b>	Capture a sale and its payment
<b>Overview</b>	A Customer arrives at a checkout with items to purchase. The Cashier records the items and collects a payment, which may be authorized. On completion, the Customer leaves with the items.
<b>Type</b>	Primary and Essential
<b>Cross Reference</b>	Functions: R1.1, R1.2, R1.3, R1.7, R1.9, R2.1, R2.2, R2.3, R2.4 Use Cases: Log In use case
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) This use case begins when a customer arrives at the POST to checkout with items to purchase. 2. (A) The Cashier records each item.(E1) 3. (S) Determines the item price and adds the item information to the running sales transaction. 4. (A) On completion of item entry, the cashier indicates to the POST that item entry is complete. 5. (S) Calculates and presents the sale total. 6. (A) The Cashier tells the customer the total.
<b>Alternative Courses of Events</b>	...
<b>Exceptional Courses of Events</b>	E1: If invalid item identifier entered, indicate error.

# Activity 2032. Refine Use Case Diagrams



- Description
  - Validate and modify the 'Business Use-Case Diagram'
  - Input : business use case model, essential use case descriptions
  - Output : A refined use case diagram
  - Standard applied : UML's use case diagram
- Step
  1. Review business use case diagrams according to essential use case descriptions
  2. Refine use case diagrams by adding or refining use cases and relationships

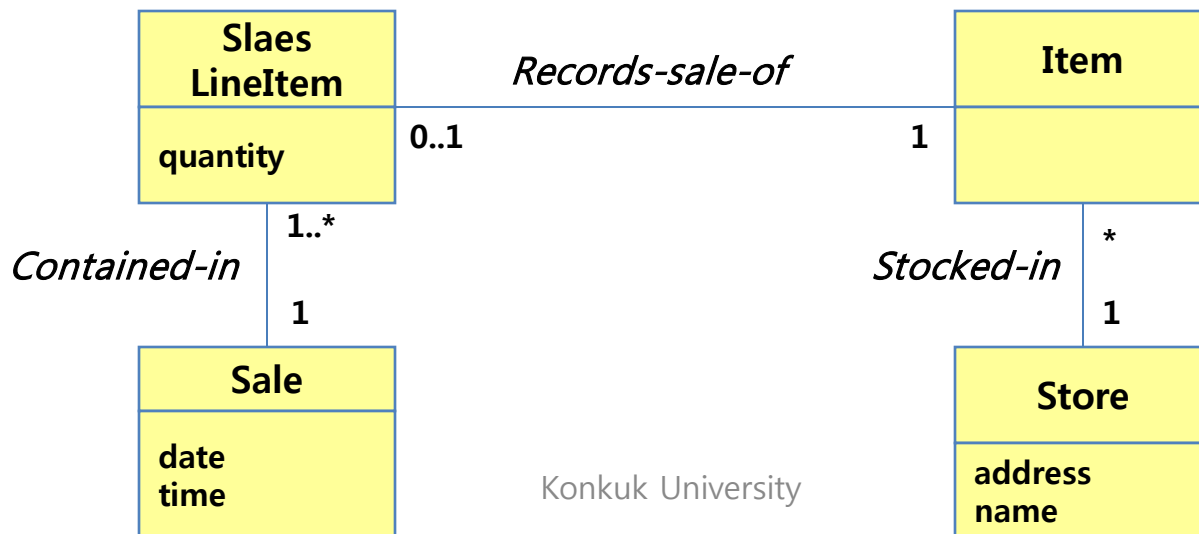


# Activity 2033. Define Domain Model



- Description

- Define domain concept model by reviewing input artifacts
- Input : essential use case descriptions, business concept model
- Output : A conceptual class diagram



# Activity 2034. Refine Glossary

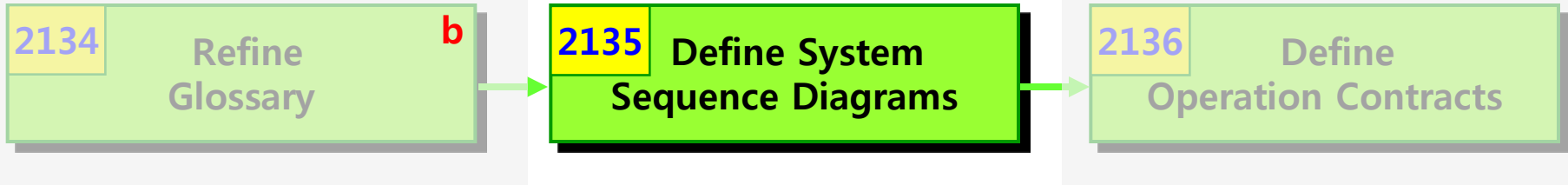


- Description
  - Lists and refines all the terms in order to improve communication and reduce the risk of misunderstanding
  - Input : term dictionary, essential use case descriptions, conceptual class diagram
  - Output : A refined term dictionary
- Step
  1. Refine terms defined in the Plan and Elaborate Phase(use cases, attributes, concept, etc.) during development cycle.
  2. Record terms as following format:

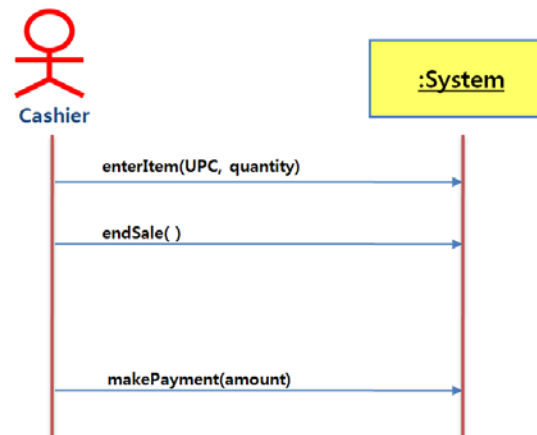
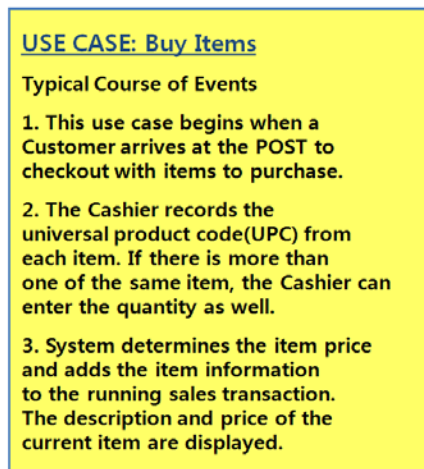
Term	Category	Comments
Payment	Concept (Class)	a cash payment
...	...	...

# Activity 2035.

## Define System Sequence Diagrams



- Description
  - Illustrates events from actors to systems.
  - To investigate the system to build
  - Input : essential use case descriptions, use case diagram
  - Output : A sequence diagram



# Activity 2036. Define Operation Contracts

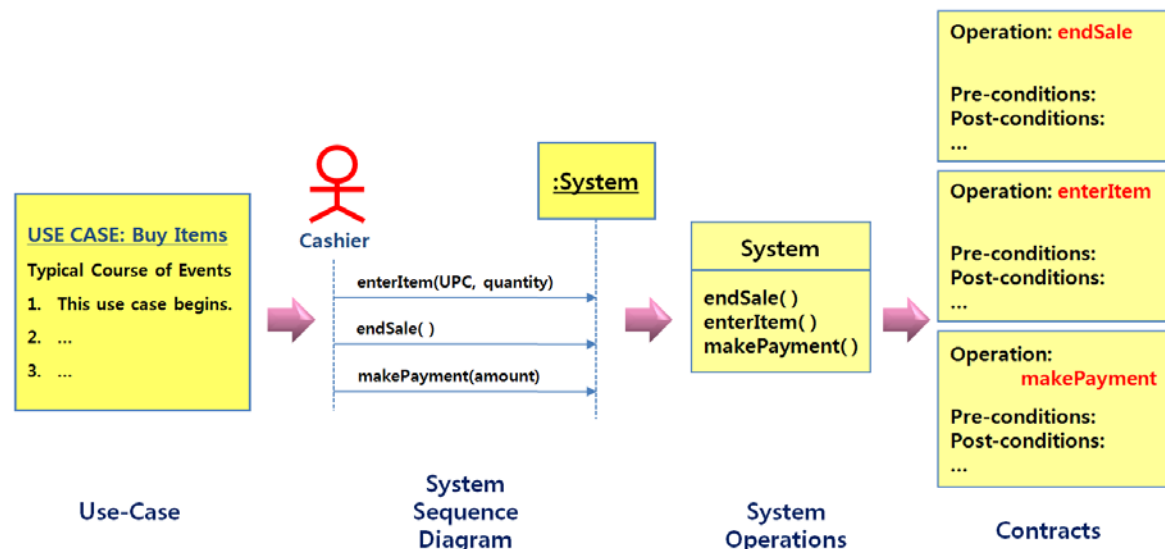
2135 Define System Sequence Diagrams

2136 Define Operation Contracts

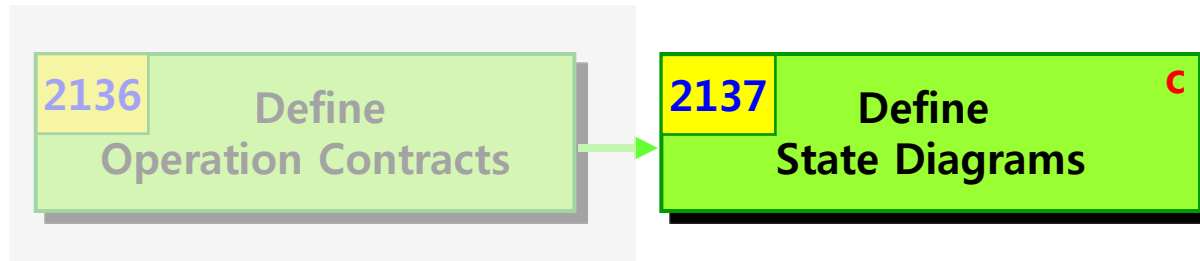
2137 Define State Diagrams C

- Description

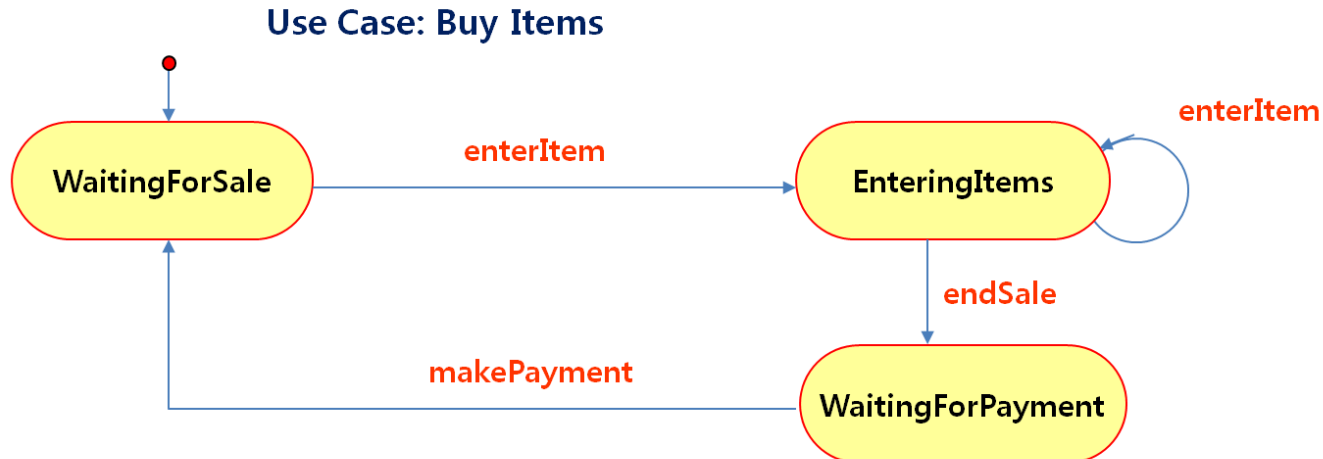
- Define contracts for system operations
- Input : system sequence diagram, conceptual class diagram
- Output : Operation Contracts



# Activity 2037. Define State Diagrams



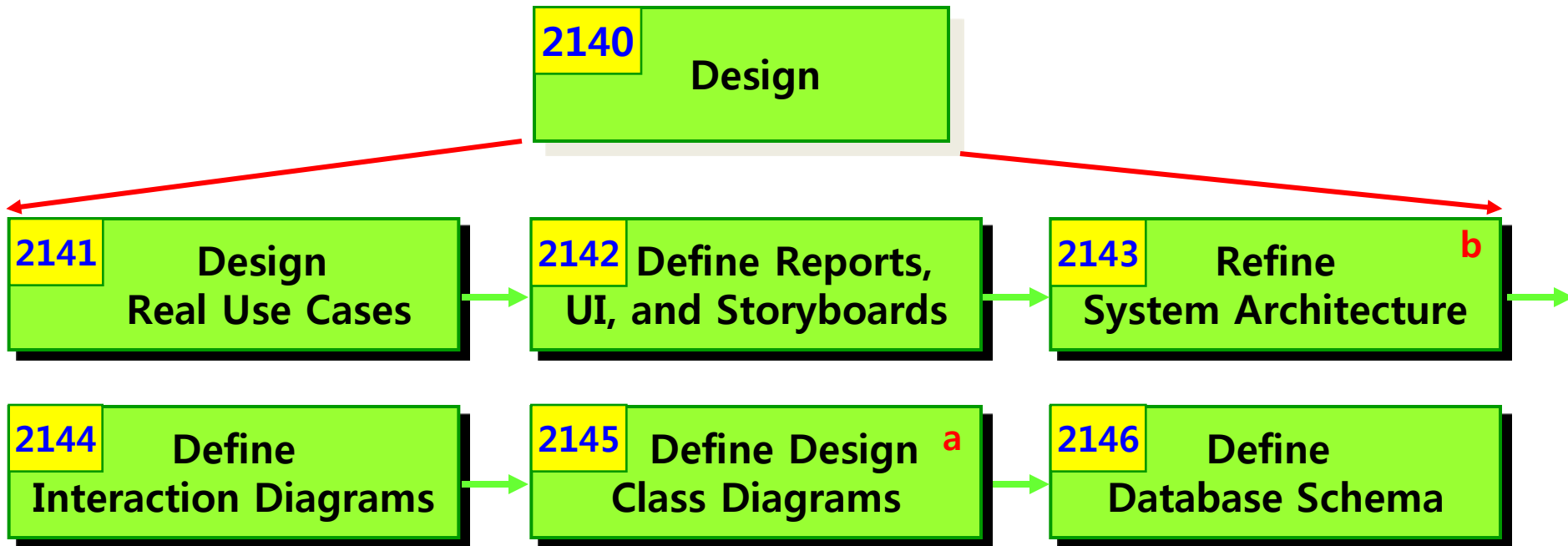
- Description
  - Describes all possible states of the system, use cases, or objects
  - Input : essential use case diagram, conceptual class diagram
  - Output : A state diagrams (for Use case, System, Class)



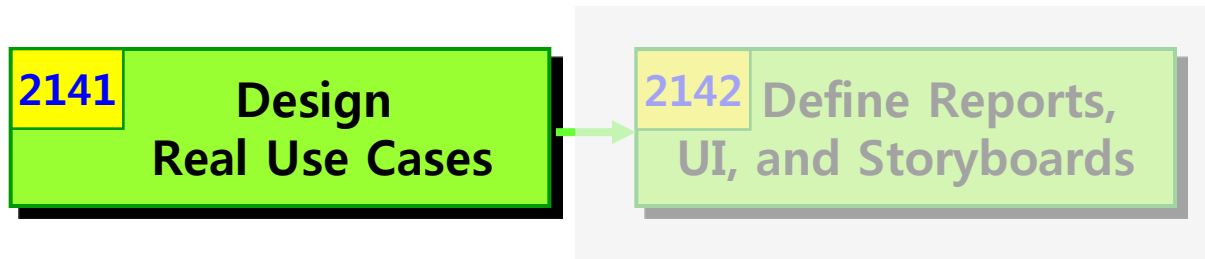
# Phase 2040. Design

- Phase 2040 Activities

a. In parallel with interaction diagrams  
b. Varied order



# Activity 2041. Design Real Use Cases



- Description
  - It describes real/actual design of the use case in terms of concrete input and output technology and its overall implementation.
  - If a graphical user interface is involved, the real use case will include diagrams of the GUI and discussion of the low-level interactions with interface widgets.

- Input
  - Essential Use Case Descriptions

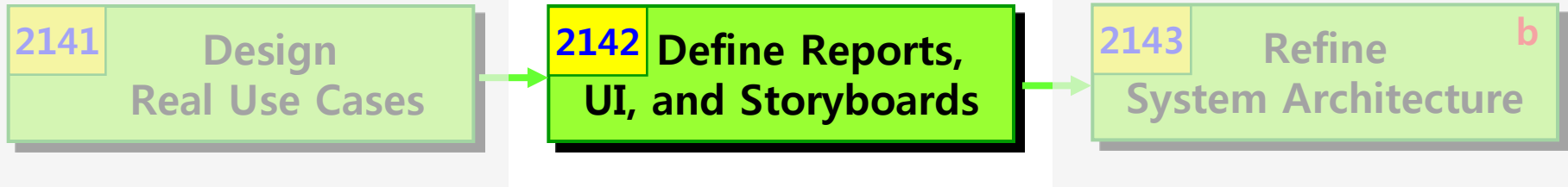
- Output
  - Real Use Case Descriptions



Use Case	Buy Items – Version 1 (Cash only)
Actor	Customer, Cashier
Purpose	Capture a sale and its cash payment
Overview	A Customer arrives at a checkout with items to purchase. The Cashier records the items and collects cash payment, which may be authorized. On completion, the Customer leaves with the items.
Type	Primary and Real
Cross Reference	Functions: R1.1, R1.2, R1.3, R1.7, R1.9, R2.1 Use Cases: Log In use case
Pre-Requisites	N/A
UI Widgets	Window-1
Typical Courses of Events	(A) : Actor, (S) : System 1. (A) This use case begins when a customer arrives at the POST to checkout with items to purchase. 2. (A) For each item, the Cashier types an UPC in A of Window-1. If there is more than one of an item, the quantity may optionally be entered in E. They press B after each item entry. (E1) 3. (S) Adds the item information to the running sales transaction. The description and price of the current item are displayed in B and F of Window1. 4. (A) The Cashier tells the customer the total.
Alternative Courses of Events	...
Exceptional Courses of Events	E1: If an invalid UPC is entered, indicate an error.

# Activity 2042.

## Define Reports, UI, and Storyboards



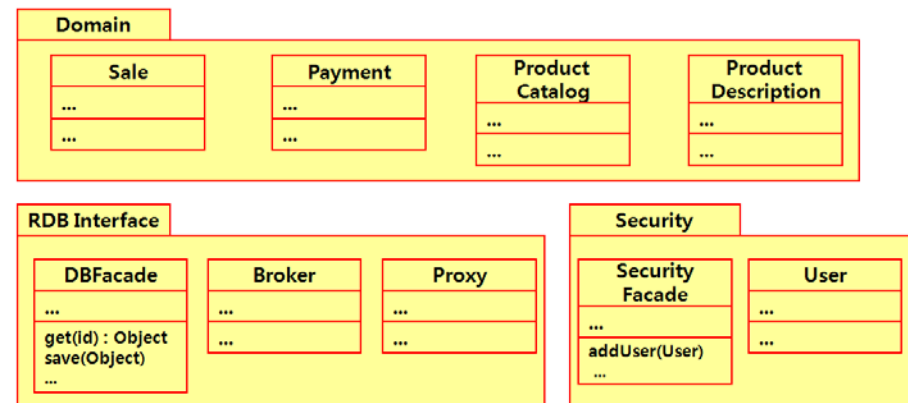
- Description
  - Design UI storyboard and UI components.
- Input
  - Requirements Specification
  - Real Use Case Descriptions
- Output
  - UI Storyboard
  - UI Component Design Specification



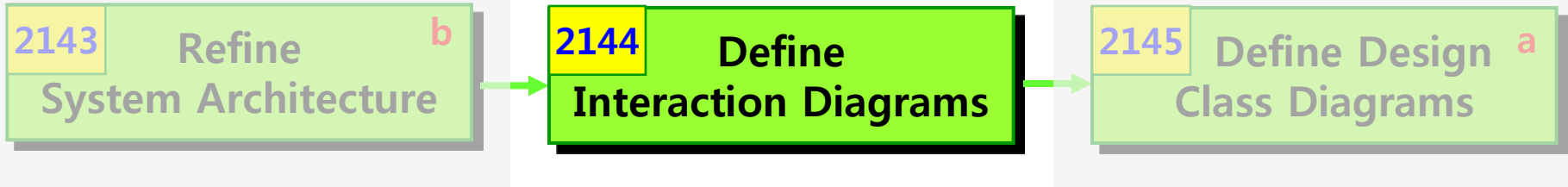
# Activity 2043. Refine System Architecture



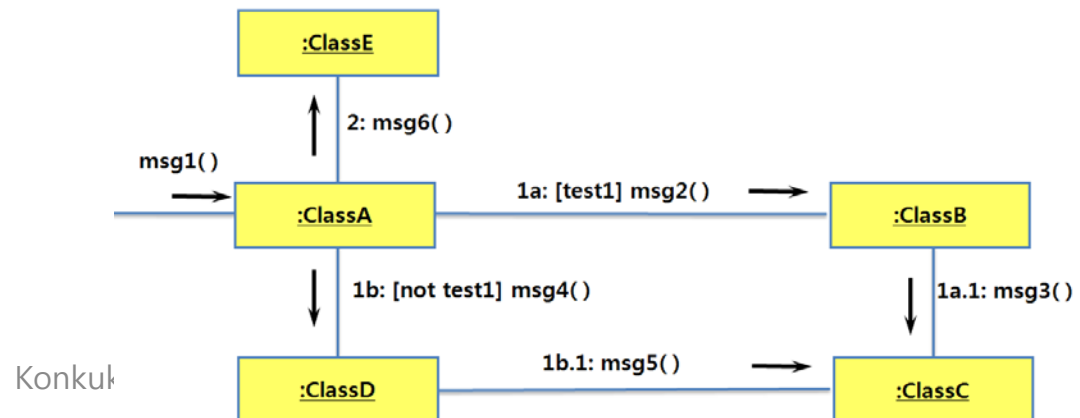
- Description
  - Refine draft system architecture developed in the plan stage
- Input : Draft System Architecture
- Output : A package diagram, a deployment diagram
- Standards Applied
  - UML's Package Diagram
  - UML's Deployment Diagram



# Activity 2044. Define Interaction Diagrams



- Description
  - Collaboration diagrams illustrate object interactions in a graph or network format.
  - To illustrate how objects interactions via messages to fulfill tasks.
- Input : Real Use Case Descriptions
- Output : An interaction diagram
  - Collaboration diagram
  - Sequence diagram

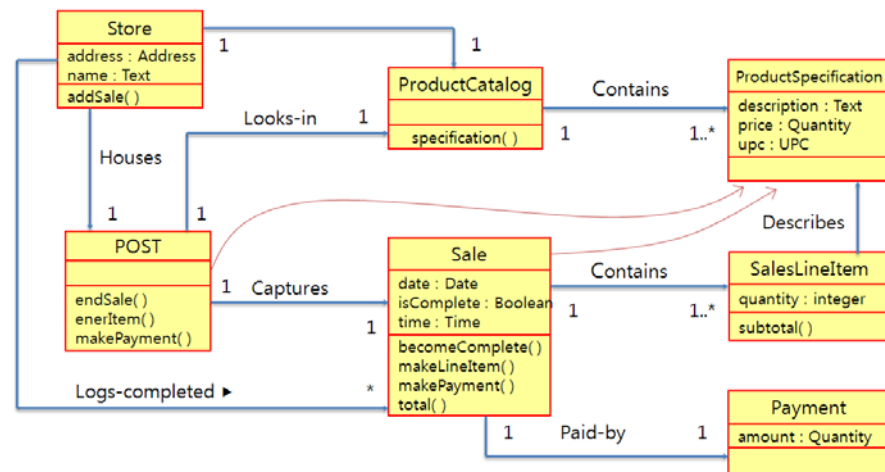


# Activity 2045.

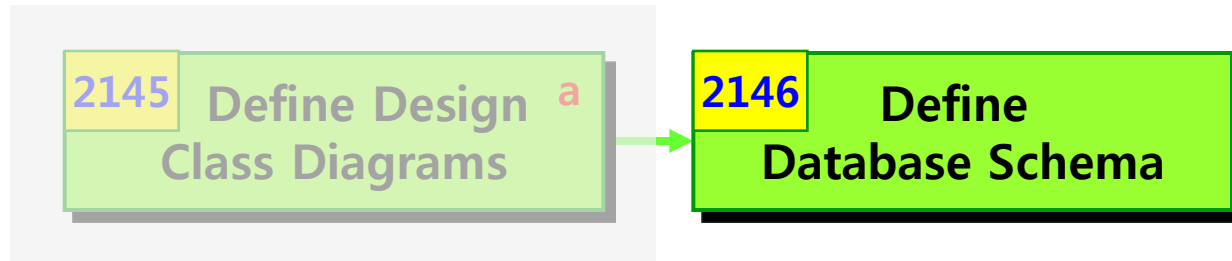
## Define Design Class Diagrams



- Description
  - Describes more details in conceptual class diagram
  - Add navigability, dependency, data type, operation signature, parameters, return types, and so on.
- Input :
  - Interaction Diagram
  - Conceptual Class Diagram
- Output : A Design Class Diagram



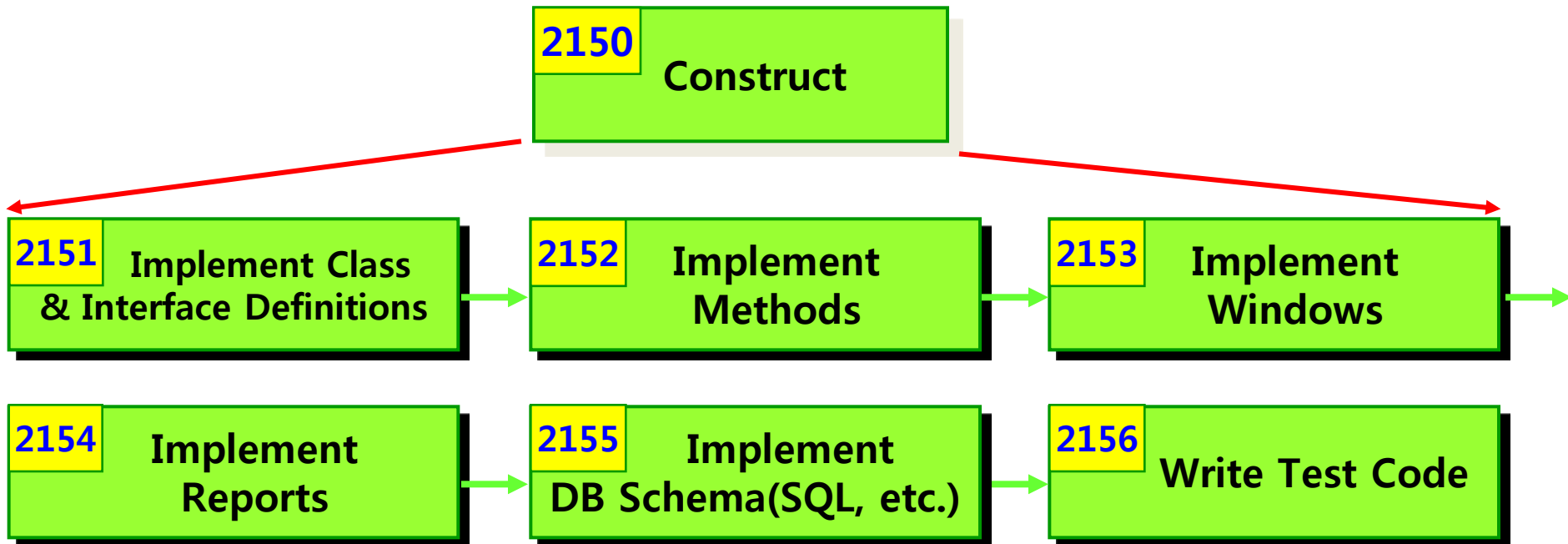
# Activity 2046. Define Database Schema



- Description
  - Design database, table, and records
  - Map classes into tables
- Input : Design Class Diagram
- Output : A Database Schema
- Steps:
  1. Map classes into tables
  2. Map relationships between classes into relations between tables
  3. Map attributes into fields of tables
  4. Design Schema

# Phase 2050. Construct

- Phase 2050 Activities



# Phase 2060. Test

- Phase 2060 Activities

