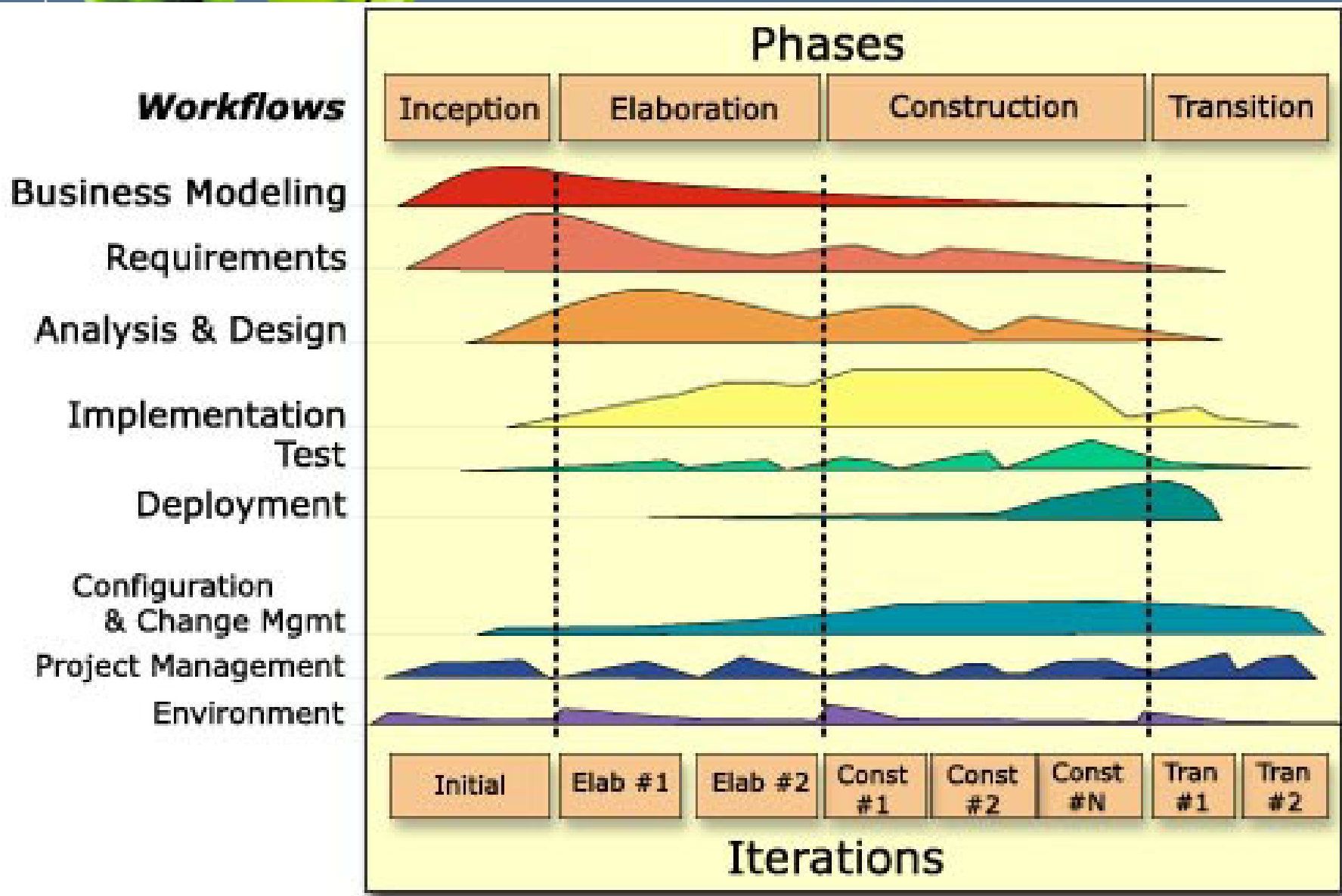


# Product Life cycle (RUP)





# Requirements workflow

CSC532:



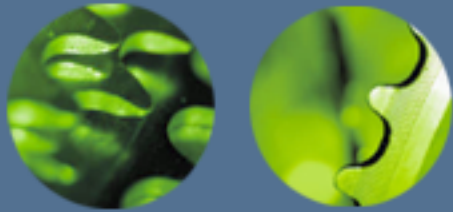
## Outline

- Introduction.
- Capture requirements.
- Artifacts.
- Workers.
- Activities.
- Next step.



## Introduction

- The fundamental principles.
- Difficulties.
  - communication.
  - articulation.
  - clarity.



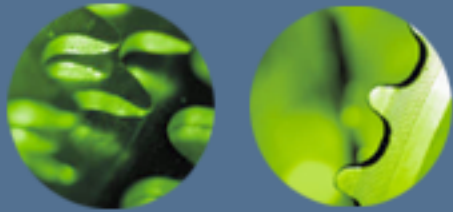
## What are requirements?

- “What customers or users expect from the system”
- Two types
  - Functional Requirements
    - Features (more tangible)
  - Non-functional requirements
    - Reliability and performance (equally if not more)



## Why important?

- Standish (1995) reports from pfleeger's book,
  - Incomplete requirement (13.1%)
  - Lack of user involvement (12.4%)
  - Lack of resources (10.6%)
  - Unrealistic expectations (9.9%)
  - Lack of executive support (9.3%)
  - Changing req and spec (8.7%)
  - Lack of planning (8.1%)



# Capture requirement

- Reach agreement on system context
  - provided by customers
  - Vision statement (e.g from marketing/product team)
  - Survey or research
- Come up with Abstractions of a given problem domain
- Arrive at actions representing/involving the abstractions (USE-CASES)



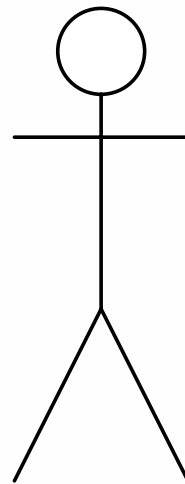
## USE CASE

- A series of actions that an actor performs in conjunction with a system to achieve a particular goal
- It only describes what but not the how a system needs to do.



## USE CASE : An Actor

Represents either a role (user) or an entity that interacts but is outside the system.

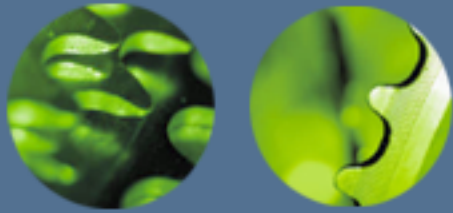


Actor1



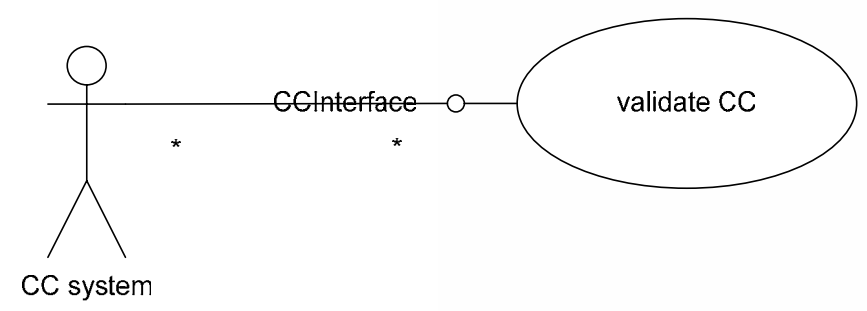
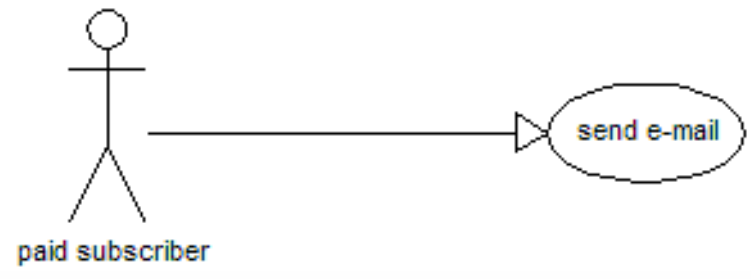
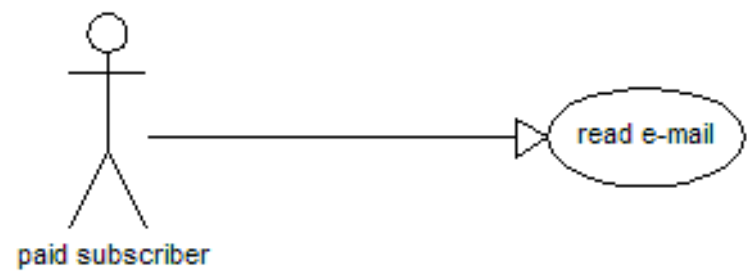
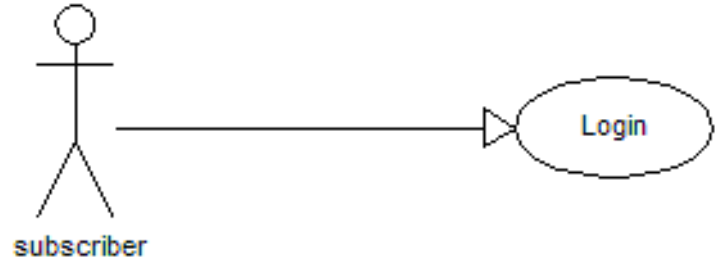
## USE CASE types

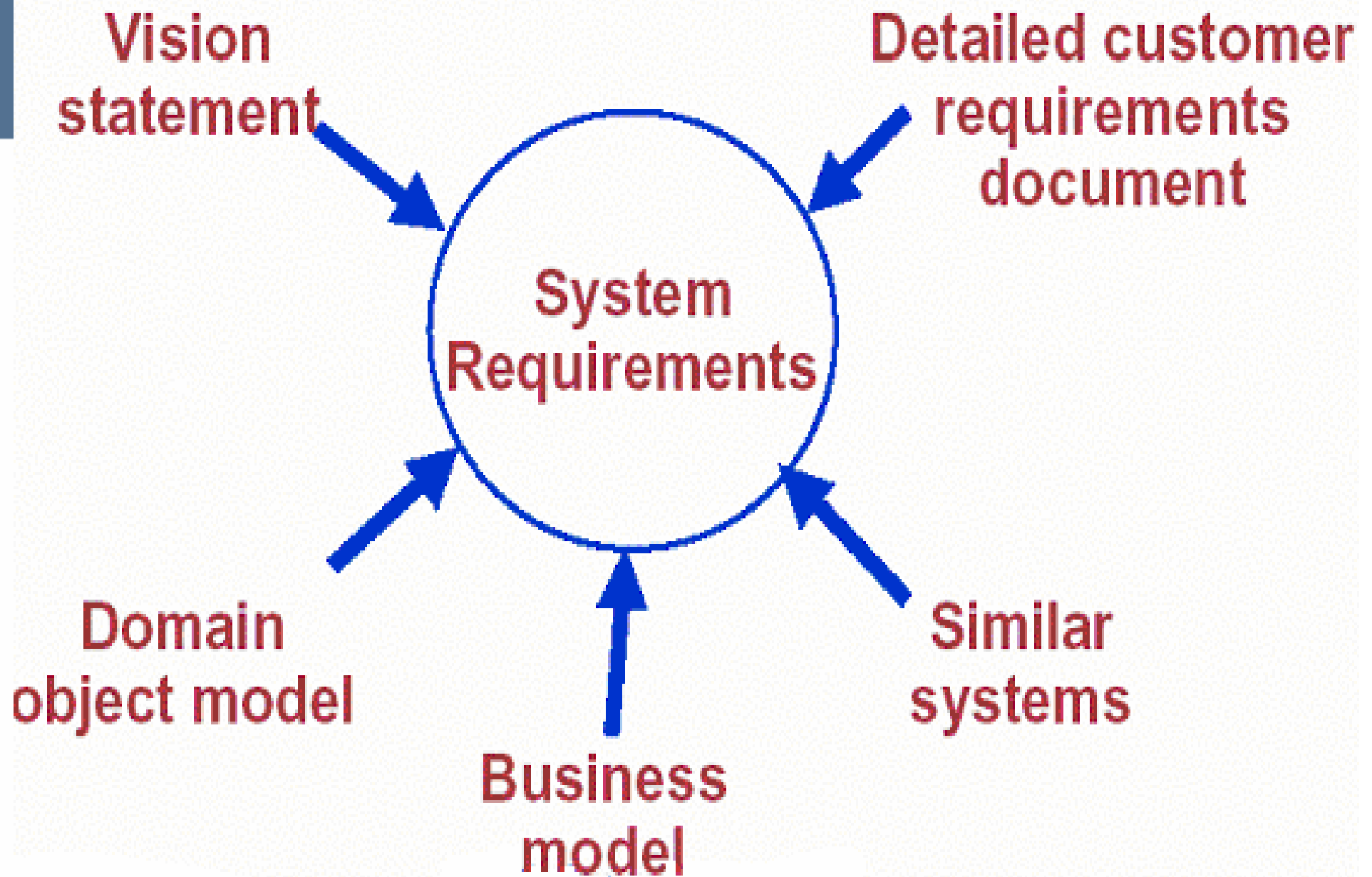
- Main flow of events
- Exceptional flow of events

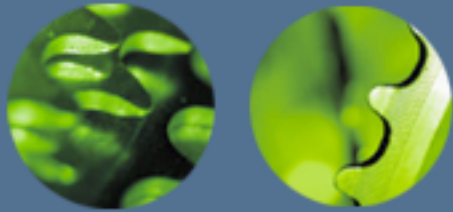


# Sample of e-mail system use cases/requirements

Yahoo.c

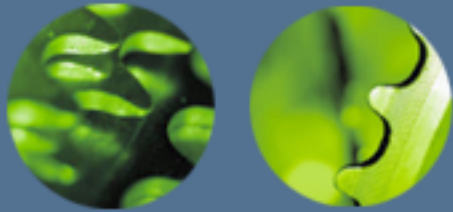






## Capture requirements

- Reach agreement on system context
  - Domain model
    - Abstraction of a given problem domain
  - Business Model
    - Use case diagrams and business actor
- List candidate requirements
- Identify and negotiate functional requirements –  
**USECASES**
- Specify non-functional requirements
  - Expressed in a supplemental document and/or as constraints in the UML diagrams



## Artifacts.

- Domain model
- Business model
- Glossary
- Actor
- Use case
- User-interface prototype
- Use case model
- Architecture description
- Supplementary requirements



# Artifacts and workers



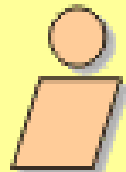
Requirements Management Plan



Stakeholder Requests



Glossary



System Analyst



Vision



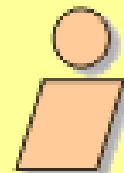
Use-Case Model



Supplementary Specification



Requirements Attributes



Requirements Specifier



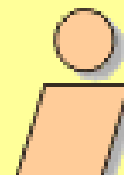
Use Case



Use-Case Package



Software Requirements Specification



User-Interface Designer



Actor (human)



Boundary Class



User-Interface Prototype



Use-Case Storyboard



## Workers

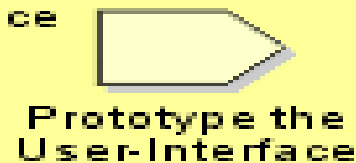
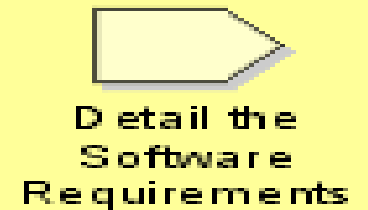
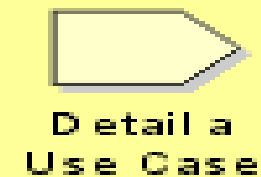
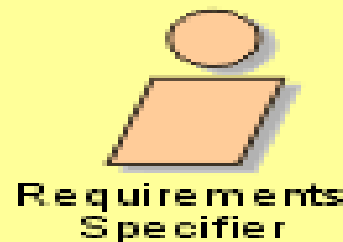
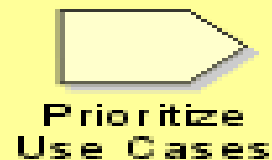
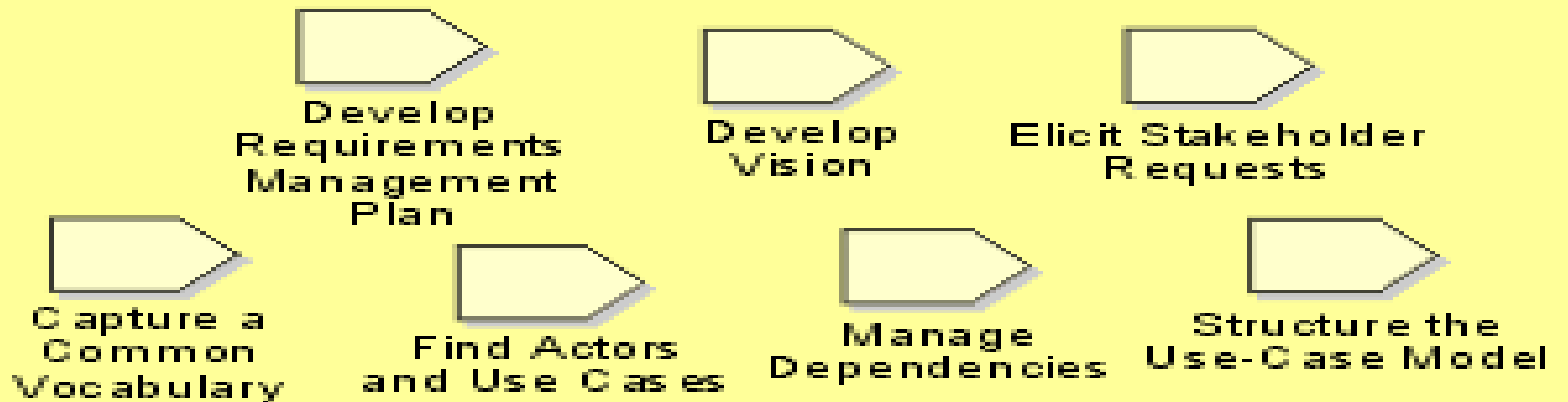
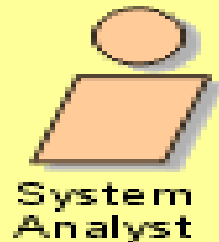
- System Analyst.
- Use case specifier
- User-interface designer
- architect



## Activities

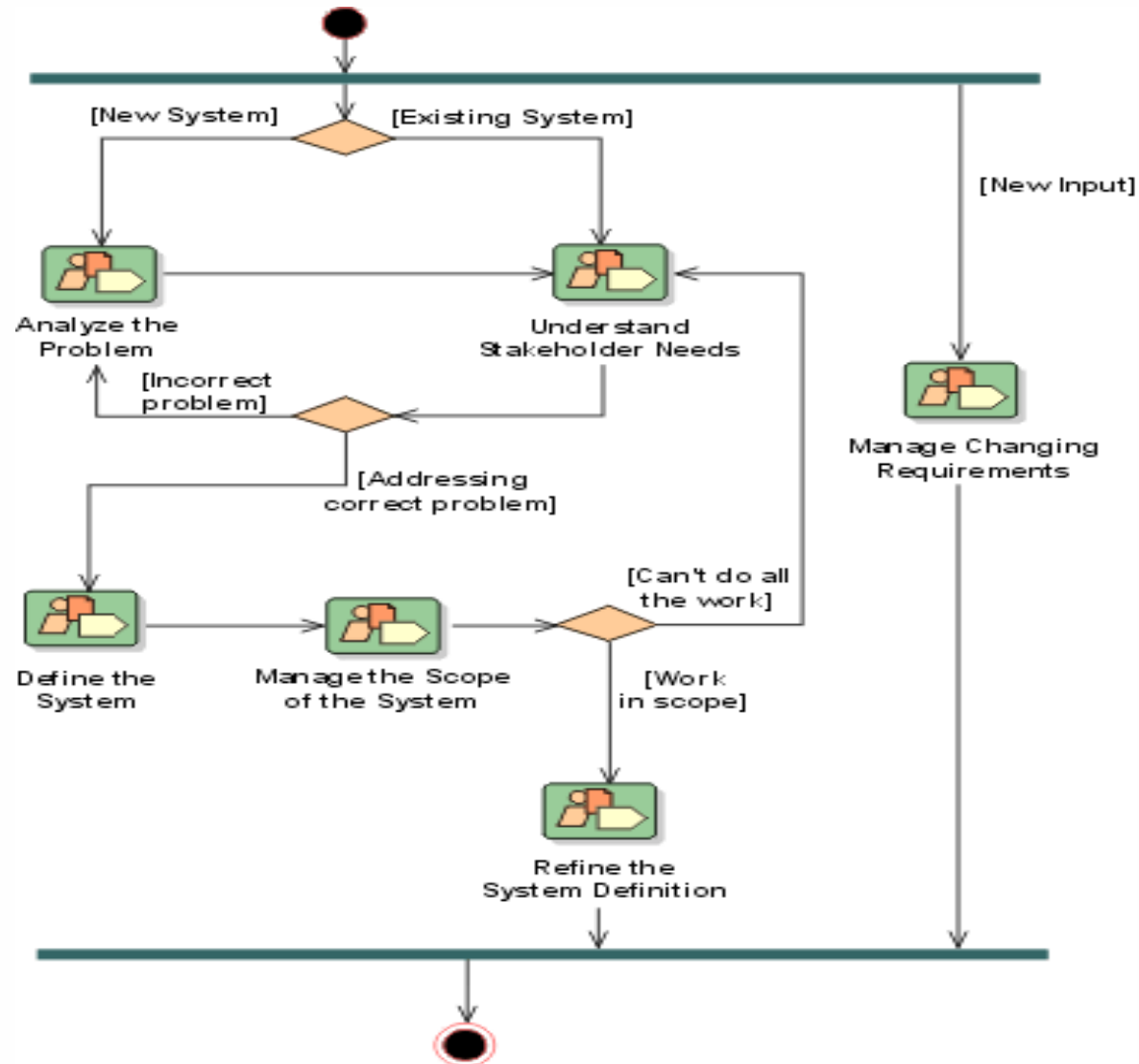
- Build domain model
- Build business model
- Find actors and use cases
- Prototype the user interface
- Prioritize the use cases
- Detail a use case
- Structure the use case model

# Activities





# Requirement workflow





## Exercise

- Group follows the req workflow guideline and works toward the project requirements
  - Brainstorm
  - Finding usecases