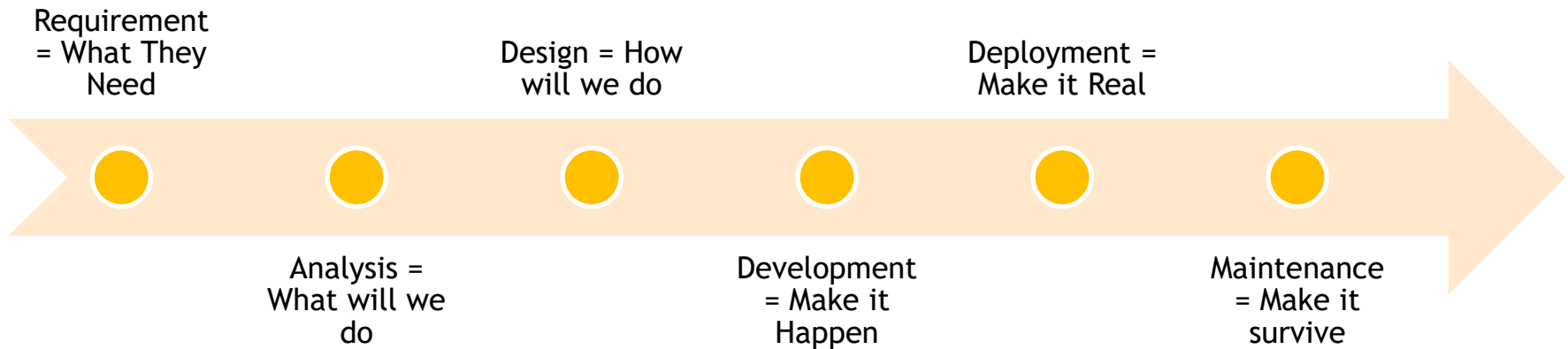


# Open Oriented Analysis and Design

Ridi Ferdiana  
[ridi@acm.org](mailto:ridi@acm.org)  
Version 1.0.0

# Requirement, Analysis, and Design



# Analysis and Design

- Shared vision between customer and the development team
- Visualize the customer need to understand the big picture of solution
- Translating between business domain with technical domain

# Analysis and Design Concept

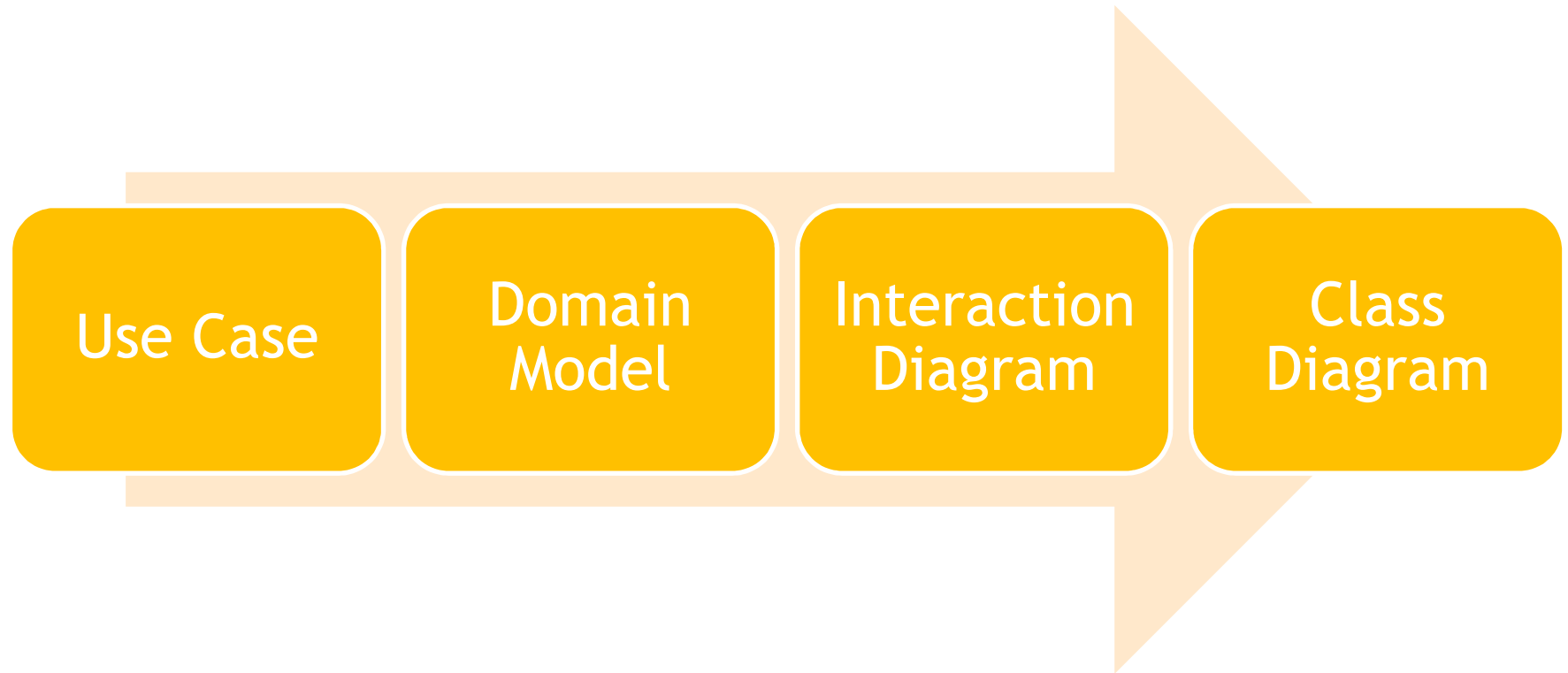
- *Analysis* emphasizes an *investigation* of the problem and requirements
- Design emphasizes a conceptual solution (in software and hardware) that fulfills the requirements

*do the right thing (analysis), and do the thing right (design).*

# OOA&D

- During *object-oriented analysis* there is an emphasis on finding and describing the problem into an object
- During *object-oriented design* (or simply, object design) there is an emphasis on defining software objects and how they collaborate to fulfill the requirements

# Practical OO&D on UML



Get deep understanding on this concept on Software Engineering Course

# *use case*

Use case is a list of actions or event steps, typically defining the interactions between a role

# *domain model*

A domain model is a visual representation of conceptual classes or real-situation objects in a domain



# *Interaction diagram*

interaction diagrams to illustrate how objects interact via messages

# *class diagram*

class diagrams to illustrate classes, interfaces, and their associations

# Practical OOA&D without UML



Features

User  
Story

Tasks

# Features

- A tech-business features for the solution

All of these features are about functionality... they focus on what the system has to do, not on what principles or patterns you use to build the system.

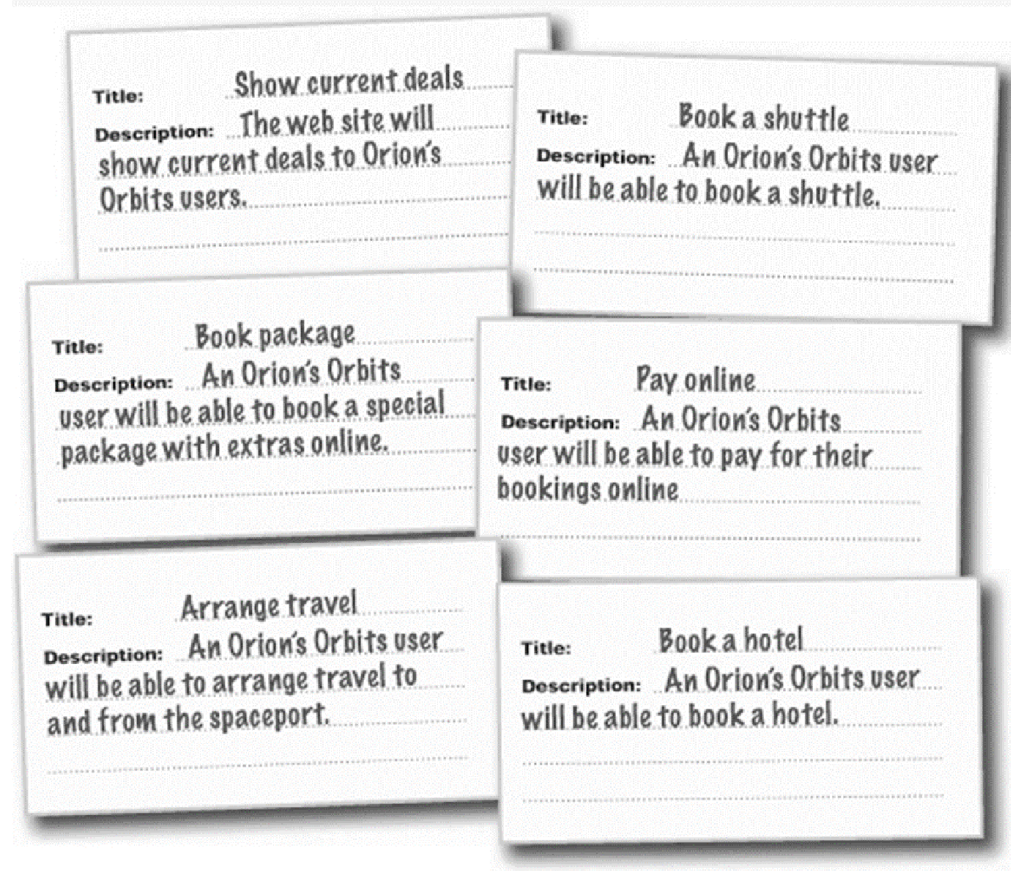
## Gary's Game System Framework Feature List

1. The framework supports different types of terrain.
2. The framework supports different time periods, including fictional periods like sci-fi and fantasy.
3. The framework supports multiple types of troops or units that are game-specific.
4. The framework supports add-on modules for additional campaigns or battle scenarios.
5. The framework provides a board made up of square tiles, and each tile has a terrain type.
6. The framework keeps up with whose turn it is.
7. The framework coordinates basic movement.

# User Story

- user story is a description consisting of one or more sentences in the everyday or business language

As a (actor)  
I want to  
(action)  
So that  
(benefit)



The image shows six overlapping user story cards for Orion's Orbits. Each card is a white rectangle with a thin black border and contains text in a typewriter-style font. The cards are arranged in a staggered, overlapping manner. Each card has a 'Title:' field followed by a 'Description:' field. The text on the cards is as follows:

- Title:** Show current deals  
**Description:** The web site will show current deals to Orion's Orbits users.
- Title:** Book a shuttle  
**Description:** An Orion's Orbits user will be able to book a shuttle.
- Title:** Book package  
**Description:** An Orion's Orbits user will be able to book a special package with extras online.
- Title:** Pay online  
**Description:** An Orion's Orbits user will be able to pay for their bookings online.
- Title:** Arrange travel  
**Description:** An Orion's Orbits user will be able to arrange travel to and from the spaceport.
- Title:** Book a hotel  
**Description:** An Orion's Orbits user will be able to book a hotel.

# User Story Card

- Usage Scenario

- User Story Description

- The Business Process of user story
    - The Business rules of user story

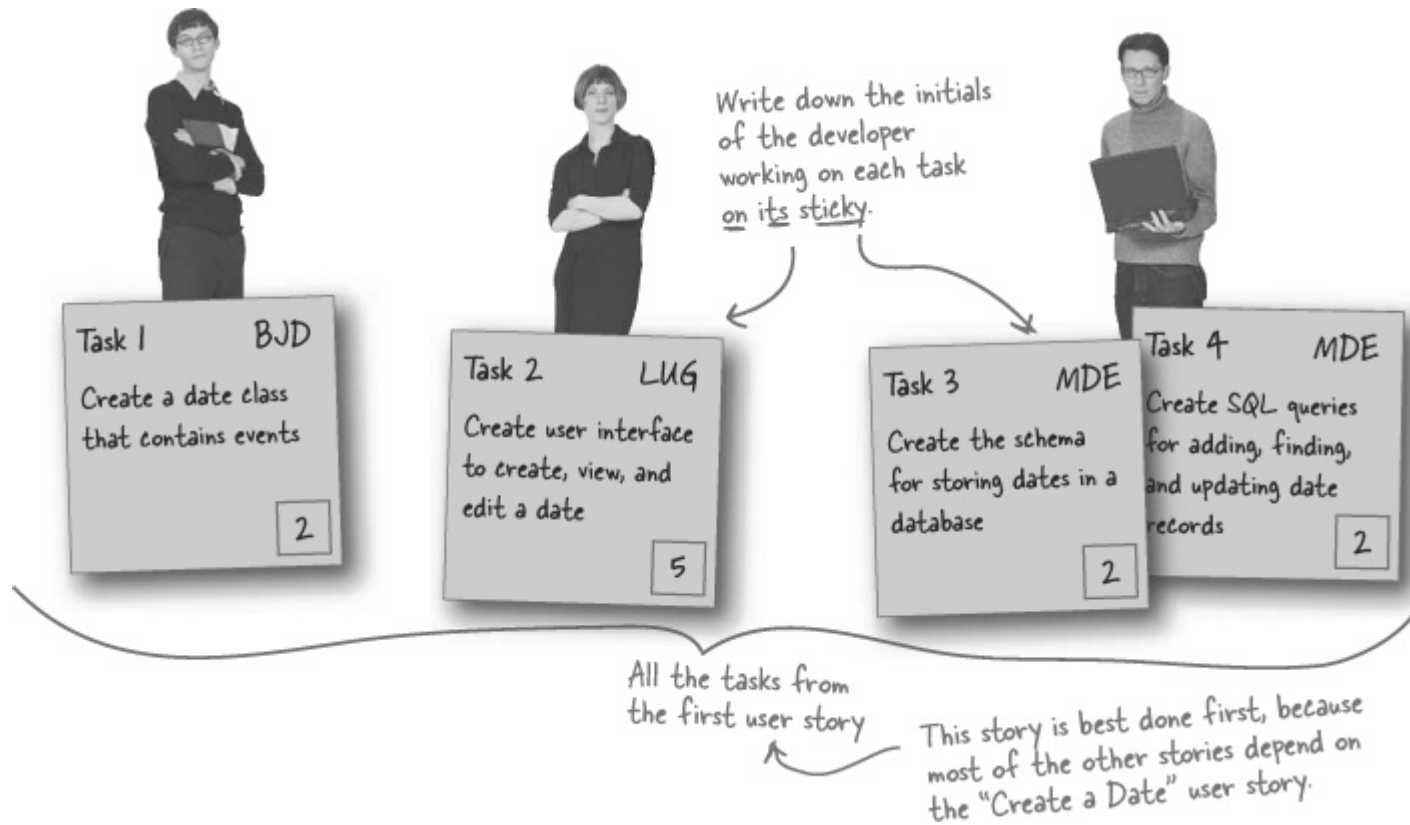
- User Story Acceptance

- The measuring factor of user story

*Test with Visa, MasterCard and American Express (pass).  
Test with Diner's Club (fail).  
Test with good, bad and missing card ID numbers.  
Test with expired cards.  
Test with over \$100 and under \$100.*

# Tasks

- Converting and Splitting User Story into a real and technical tasks





# *User story*

User story on Scrum Method



# Design Principles on OOA&D

- Noun Analysis (NA): Converting sentences into nouns and verbs. Noun goes to Class, Verbs goes to Method
- Single Responsibility Principle (SRP): Every object in your system should have a single responsibility, and all the object's services should be focused on carrying out that single responsibility.
- Don't Repeat Yourself (DRY): Avoid duplicate code by abstracting or separating out things that are common and placing those things in a single location

# Architecture

- Architecture is the organizational structure of a system, including its decomposition into parts, their connectivity, interaction mechanisms, and the guiding principles and decisions that you use in the design of a system.
- Architecture patterns, for example
  - Model View Control
  - Layered Architecture
  - Tier Architecture
  - Etc.

# Zen Architecture

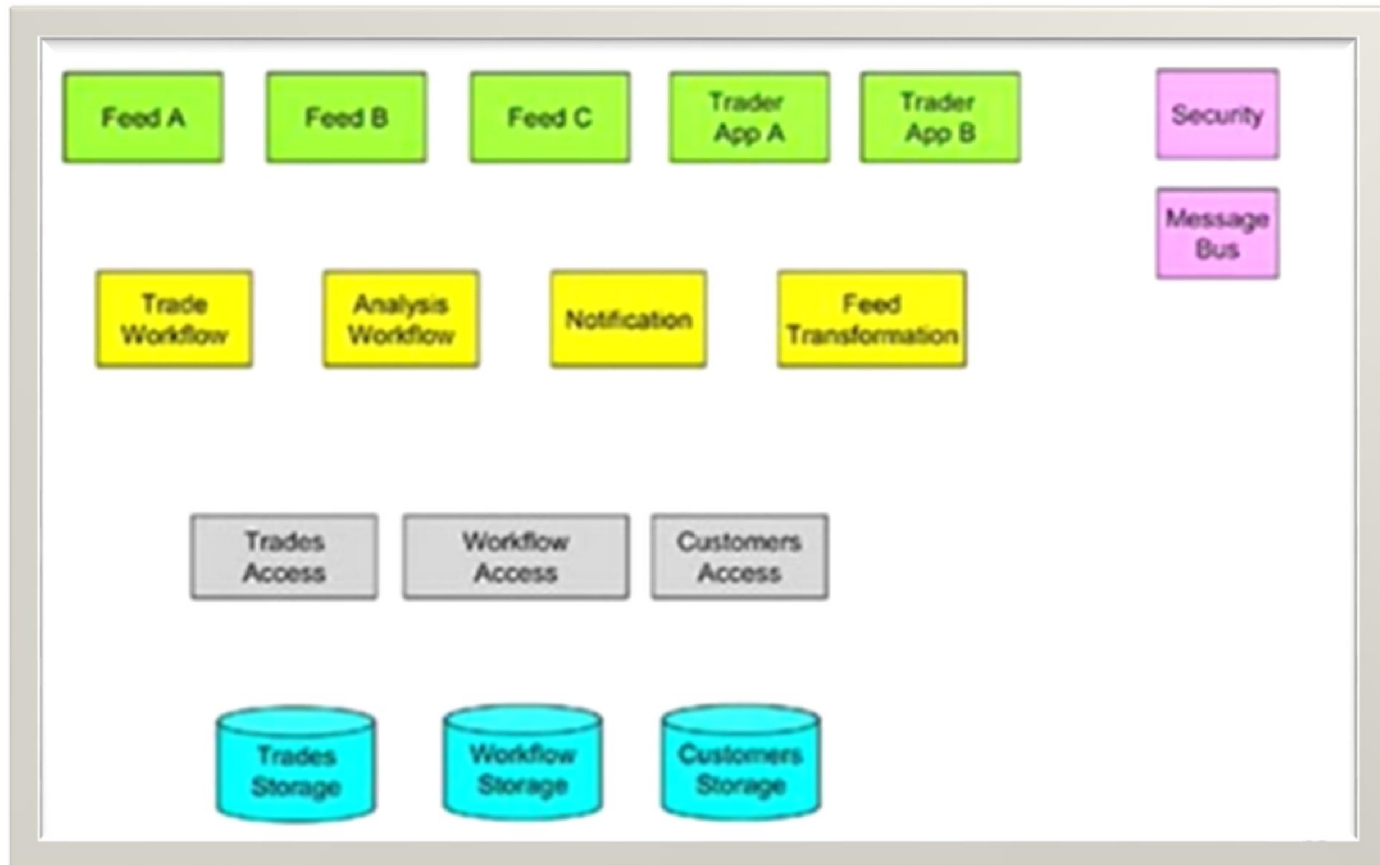
## ● Functional Decomposition

- Example: stocks trading system
- Requirements
  - Enable traders to buy and sell stocks
  - Schedule trades
  - Issue reports
  - Process market data feeds
  - Analyze trades
  - Client is browser
    - ▲ Connected sessions
  - After a trade/report sends email confirming request or results
  - Data is stored in database



# Zen Architecture

- Functional Decomposition



# Key points

- Analysis: understanding the customer problem into possible solution
- Design: proposing 'enough solution' through technical approach
- OOA&D: an object oriented style of A&D
- Architecture: visualization of the design

# References

- Russ Miles; Dan Pilone. 2007. Head First Software Development. O'Reilly Media
- Craig Larman. 2004. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition. Prentice Hall
- Mike Cohn. 2006. Agile Estimating and Planning, Prentice Hall
- Tilak Mitra. 2015. Practical Software Architecture: Moving from System Context to Deployment. IBM Press
- Juval Löwy. 2015. The Zen of Software Architecture. O'Reilly Media