Tutorial #1 – Requirements Elicitation – Functional / Non-Functional Requirements

1. Atomise the following descriptions for an Inventory and Asset Tracking System as functional requirements. Check that the atomized requirements are also verifiable.

   The System maybe queried. A query may contain a user number or a serial number. If the query contains a user number, all equipment assigned to that user is reported. If the query contains a serial number, the assignment for that computer is reported. All query results show the name, office and user number of a user, followed by the serial numbers and types of all computers assigned to that user, and the date of each assignment. If a user is not assigned a computer or a computer not assigned to a user, this is reported.

2. Identify several non-functional requirements for the above Inventory and Asset Tracking System.

Tutorial #2 – Requirements Analysis – Use Case & Conceptual Model in a Class Diagram

1. Refer to the description of the Grain Elevator System (GES).

   (a) Identify the Actors and their main Use Cases and draw a complete Use Case diagram for the Grain Elevator System. Wherever appropriate, use generalization, include and extend relationships.

   (b) Write the Use Case description for the use case(s) that encompasses the grain shipment arrival and subsequent allocation functionality.

   (c) Determine and document important terms in the Data Dictionary

2. Refer to the shipment arrival and allocation functionality obtained in the above question.

   (a) From the Use Case description and Data Dictionary, identify the classes to produce a conceptual model.

   (b) Depict the classes identified in part (a) and the associations/dependencies between them in a Class Diagram.

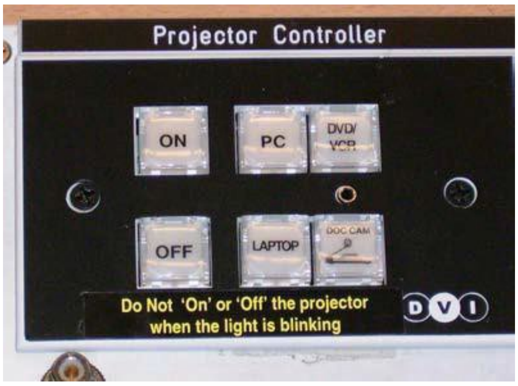
Tutorial #3 – Requirements Analysis –
Dynamic Models in Sequence Diagram and State Machine Diagram

1. Grain Elevator System (GES). Refer to the shipment arrival and allocation functionality obtained in Tutorial #4.

   (a) In a Sequence Diagram, model the interaction amongst boundary / control / entity classes to enact the shipment arrival functionality.
   Demonstrate how the messages passed between objects enact the Use Case's flow of events.

   (b) Refine the conceptual model obtained in Tutorial #4 using details uncovered in question 1(a) above by adding:
   - associations,
   - generalization relationships,
   - properties / attributes,
   - operations, and
   - new classes (if any)

2. 8:30am lectures are usually frustrating for the lecturer. Projectors, having been turned off at the end of the previous day and take a long time to become ready for use.


Projector Controller
ON
PC
DVD/VCR
OFF
LAPTOP
DOC CAM
Do Not 'On' or 'Off' the projector when the light is blinking
DVI

Here's what Professor Jane does at the Projector Controller Panel, in order to project slides from her laptop:-

- Turn on the projector
- As the projector warms up, it first gives a beep (sound) and a red indicator light continues to blink
- When ready, the indicator light turns to green; the projector takes input from the PC by default
- Jane presses the "LAPTOP" switch

After the lecture, Jane could turn off the projector by pressing the "OFF" button. However, to save her colleagues the angst, she decides to leave it ON.
Model the projector's system behavior using a State Machine diagram (can also be called a dialog map.

CE2006/CZ2006 Software Engineering

Tutorial #4 – Software Processes

1. Why are software development costs so high?

2. Why do we spend so much time and effort maintaining existing software programs?

3. Discuss on the characteristics of software that affect its development.

4. What are the two distinct differences between software development and hardware manufacturing?

5. Sum up Agile manifesto in four sentences, then discuss the key Agile principles.

6. Discuss the following characteristics of eXtreme Programming (XP):
    i. When to use XP
    ii. Pair programming
    iii. Test-driven development (TDD)
    iv. Code refactoring

Tutorial #5 – Scrum

1. What are the considerations crucial to deciding whether to adopt Agile/Scrum development in a project?

2. Discuss the ways of estimating Velocity in Scrum.

3. What are the characteristics of a good product backlog?

4. What does the INVEST mnemonic of Agile software development stand for?

5. Under Scrum methodology, a Product Increment is a piece of software that is both complete and potentially shippable. Why is it important to have a working software delivered at the end of each Sprint?

6. The total size (in points) of a list of product backlog items is 64. The Scrum Team consists of Product Owner, Scrum Master and six members in the Development Team. The cost of each member in the Scrum Team for each Sprint is $2K. Eight Sprints is determined to be needed to complete the project. Calculate the velocity and cost of this project.

Tutorial #6 – Software Architecture and Strategy Pattern

1. Grain Elevator System (GES)

   a) Refer to the conceptual model developed in Question 1 of Tutorial#4.

   Use proper architectural styles to model dependencies among boundary, control, and entity objects.

2. Grain Elevator System (GES)

   The elevator manager uses a mobile device to process shipment and allocation. This mobile device must deal with a variety of network access protocols (LAN, WiFi, Bluetooth, 3G, 4G). Furthermore, we want to be able to deal with further network protocols with minimal impacts on the application.

   a) Identify the design problem and apply appropriate design pattern to address this problem

   b) Draw a UML class diagram depicting the classes in the design pattern and explain their roles

Tutorial #7 –Observer Pattern and Factory Pattern

1. Grain Elevator System (GES)

   Once grain arrives or leaves, GES will notify processing plants. The processing plants can register their interests in specific types of grains. GES will notify the registered processing plants using the plant preferred communication means, including email, SMS, or the combination of these means.

   Note that the interests and preferred communication means of the processing plants vary greatly, and the processing plants can change their interests and preferred communication means over time. Furthermore, GES should be easily extended to support new communication means once they become available.

   a) Please identify the design problem in this feature and suggest a design pattern for addressing this problem.

   b) Illustrate your solution in a class diagram and explain the roles of each class.

2. Refer to the design pattern solution developed in Question 2 of Tutorial#6.

   a) Discuss what is missing in the solution

   b) Add a relevant design pattern to the design solution to complete the design

Tutorial #8 – Equivalence Class and Boundary Value Testing

1. Grain Elevator System

   A silo cannot accept any more grain if its temperature or humidity exceeds normal levels. Humidity ranges 0-100%, the normal humidity is 35 – 50%.
   Temperature ranges -10 – 100 C, the normal temperature is 40-60 C.

   a) Determine equivalence classes and boundary values for humidity and temperature

   b) Systematically design a set of test cases to test whether silo can accept grain or not based on its temperature and humidity.

Tutorial #9 – Control Flow Testing

1. A Java method to compute the sum of 1 .. n is shown in the following Java code below:

```
public int sum(int n, int upperbound){
   1  int result, i;
   2  result = 0;
   3 i =0;
   4 if(n < 0) {
   5    n = -n;
   6  }
   7 while(i<n && result <= upperbound){
   8    i = i + 1;
   9    result = result + i;
   10 }
   11 if(result <= upperbound){
   12   System.out.println("The sum is " + result);
   13  } else {
   14   System.out.println("The sum is too large!");
   15 }
   16  return result;
}
```

a) Draw the corresponding control flow graph for the `sum(int n, int upperbound)` method

b) Calculate the Cyclomatic Complexity of the `sum(int)` method

c) Design test cases to achieve 100% statement coverage and 100% branch coverage

d) List the basis set of linearly independent paths, along with a test case (input parameters) and expected outcome for each of the path in the basis path set