



Computer Science Department
Software Engineering (COMP 433)
1st Semester 2013/2014

Midterm Exam

(75 minutes)

5/01/2014

Student Name: _____ ID: _____ Section: Dr. Yousef Dr. Mamoun

[Question 1: 20%]

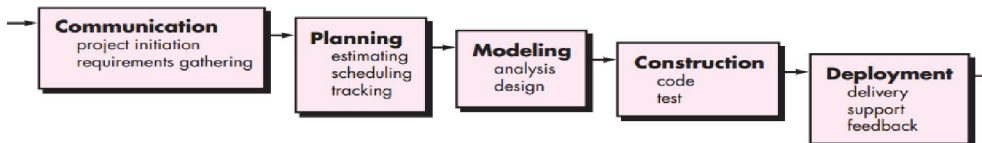
a) List three ways that software engineering differs from programming:

- A Programmer writes a **complete** program
 - a software engineer writes a software component that will be combined with components written by other software engineers to build a system
- Programming is primarily a **personal** activity
 - Software engineering is essentially a team activity
- Programming is just one aspect of software development
 - Large software systems must be developed similar to other engineering practices

b) Define “Agile software engineering”:

- **Effective** (rapid and adaptive) **response to change**.
- **Effective communication** in structure and attitudes among all team members, technological and business people, software engineers and managers.
- **Drawing the customer into the team**.
- **Planning** in an uncertain world has its limits and plan must be **flexible**.
- **Organizing a team** so that it is in control of the work performed.

c) List the activities of waterfall model:



d) Discuss how waterfall model and iterative models such as Rational Process deal with changing in customer requirements:

In waterfall model, the inflexible partitioning of the project into distinct stages makes it **difficult to respond to changing** customer requirements. Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.

However, the iterative models have been explicitly designed to accommodate a product that evolved over time. This enables you to develop increasingly more complete version of the software as requirements often change as development proceeds.

a) What is the distinction (difference) between functional and non-functional requirements?

- ❖ Functional requirements: Describe **user tasks** which the system needs to support.
“An operator must be able to define a new game”
- ❖ Nonfunctional requirements: Describe **properties of the system** or the domain.
“The response time must be less than 1 second”

b) Based on your deliverable for phase one of the Municipality project, select a use case and specify one functional and one non-functional requirements, your specification should be clear, correct, feasible, and verifiable:

c) Consider an ATM system. Identify three different actors that interact with this system.

1	Bank Customer
2	ATM Maintainer
3	Central Bank Computer
	Thief

d) What is the difference between a scenario and a use case? When do you use each?

A scenario is an actual sequence of interactions (i.e., an instance) describing one specific situation;

A use case is a general sequence of interactions (i.e., a class) describing all possible scenarios associated with a situation.

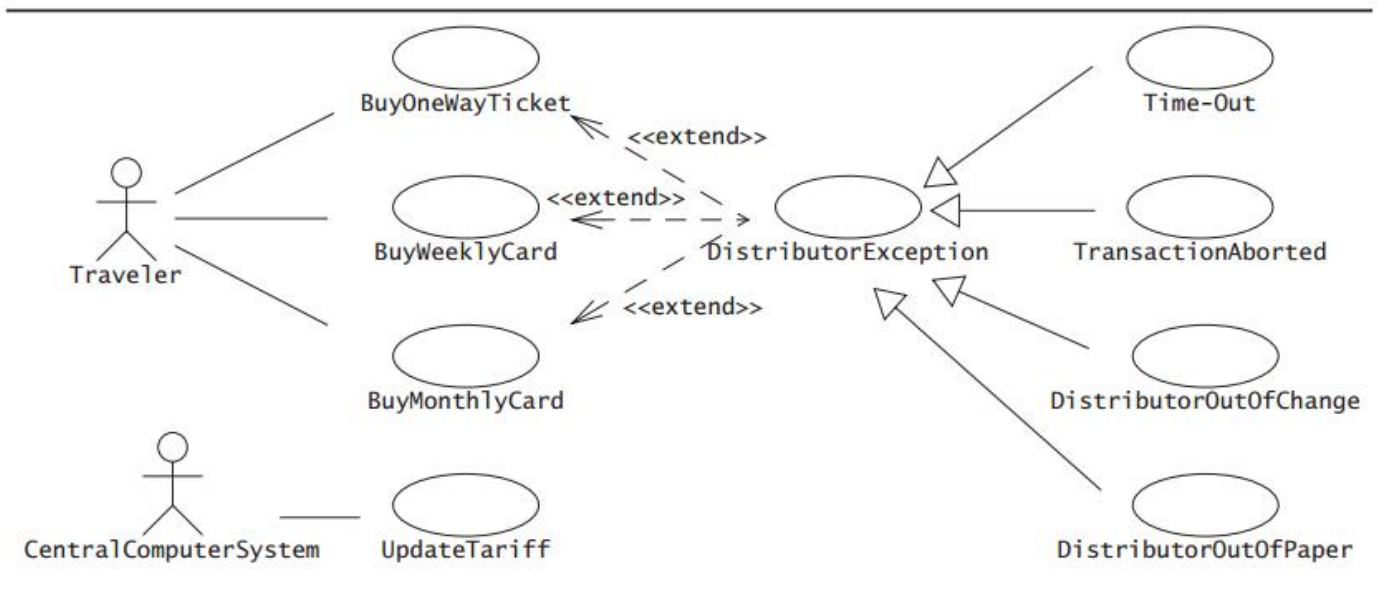
Scenarios are used as examples and for clarifying details with the client.

Use cases are used as complete descriptions to specify a user task or a set of related system features.

[Question 3: 20%]

Draw a use case diagram for a ticket distributor for a train system. The system includes **two actors**: a **traveler**, who purchases different types of tickets, and a **central computer system**, which maintains a reference database for the tariff (i.e. rate, price). Use cases should include: **BuyOneWayTicket**, **BuyWeeklyCard**, **BuyMonthlyCard**, **UpdateTariff**. Also include the following exceptional cases: **Time-Out** (i.e., traveler took too long to insert the right amount), **TransactionAborted** (i.e., traveler selected the cancel button without completing the transaction), **DistributorOutOfChange**, and **DistributorOutOfPaper**.

This question can have several correct answers; the following figure is a possible answer:



The following elements should be present:

- The relationship between an actor and a use case is a communication relationship (undirected solid line).
- The relationship between exceptional use cases and common use cases is an «extend» relationship.
- The exceptional use cases described in the exercise only apply to the use cases invoked by the traveler.

The following elements should be present in a “good” answer:

- All exceptions apply to all traveler use cases. Instead of drawing 3x4 relationships between these use cases, an abstract use case from which the exceptional use case inherit can be used, thus reducing the number of «extend» relationships to 3 at the cost of introducing 4 generalization relationships.