



Iqra National University, Peshawar
Department of Computer Science
Summer Semester 2020
Final – Term Examination

Course Code: 102007052

Course Title: Software Engineering

Instructor: Engr. Ghassan Husnain

Program: BS CS (Software Engineering)

Total Marks: 50 Time Allowed: 4 Hours

Note: Attempt all Questions

Name: Alamgir Khan	Id: 13379
BS(SE) Section (A)	Course: Software Engineering
Final-Term Exam	Date: 25/09/2020

Q.1: Explain why the rapid delivery and deployment of new systems is often more important to businesses than the detailed functionality of these systems. (6 marks)

Answer # 01:

Many companies today are willing to compromise on the quality of software and the need for faster delivery of software. As such, rapid development and delivery is the most important requirement for software systems. Because these companies operate in a changing environment. They have to respond to new opportunities and markets, changing economic conditions and the emergence of competitive products and services. Therefore, focus on providing fast delivery system. This is useful for a system that needs to display system results. This is also useful for businesses as the system can be put to initial use when key features become available and can be improved later as the user needs to change. Rapid delivery can make profit swiftly. The rapid software development process is designed to produce useful software. At the same time, it is not good for system maintenance because the system must be error-free.

Q.2: Explain how the principles underlying agile methods lead to the accelerated development and deployment of software. (6 marks)

Answer # 02:

The principles underlying agile methods lead to the accelerated development and deployment of software.

Agile methods:

The system develops step by step with the help of agile development process. As a result of these methods, the system has changed in small steps according to the requirements of the users. Consumers are involved in the development process. Agile methods focus on system software rather than design and documentation. In these ways, developers can quickly deliver software to the user.

1. Incremental delivery:

In this process, the software is delivered in small increments to the customer. Depends on the customer feedback and requirements, the developers made the increments in the system. In every increment, the new functionalities are developed and deployed into the system.

2. Customer involvement:

The customers are involved in the development process of the system. The agile methods involve regular discussions with the customer. As the development of the system is done in small increments, the customer must involve in the development process to provide the requirements of the new features in the system.

3. People, not process:

One of the most important principles for developing effective software is to recognize the technical skills of those working in the system development process. Products must be implemented and delivered quickly and with high quality.

4. Embrace change:

As technology is developing, the changes must be done to the existing system. The system must be designed as per the requirements of the changes to deploy the new features in the system.

5. Maintain simplicity:

As the changes in the system done frequently in small increments, the code used in the system must be simple. If changes are required for the system, the existing code must be restructured as per the requirements of the changes. The simplicity of the code and the development process must be maintained without any complexity in the system.

Q.3: Extreme programming expresses user requirements as stories, with each story written on a card. Discuss the advantages and disadvantages of this approach to requirements description. (6 marks)

Answer # 03:

Extreme programming (XP) is probably the most understood and well-known method. Beck developed the name because the approach was developed by promoting recognized best practices.

In extreme programming, the requirements are presented as scenarios that apply directly to a series of activities. The program includes additional planning, small mini-publications, simple design, trial first development, refactoring, pair programming, collective ownership, and sustainable rhythm and site users.

Advantages and Disadvantages of Extreme Programming user requirements:

Advantages:

1. Scenarios cope with most of common operation. It is easy to identify what type of operation that is required in the user's stories.
2. Customer focus in the scenario card increase the chance that the software produced will actually meet the needs of the users.

Disadvantages:

1. Using scenarios on a card can bring to a function overlooked or omission which can be a time-consuming process to complete the system.
2. Two different scenarios can lead to the same function as it will be conflicted each other. Crossing out redundant scenarios can be a cumbersome task.

Q.4: To reduce costs and the environmental impact of commuting, your company decides to close a number of offices and to provide support for staff to work from home. However, the senior management who introduce the policy are unaware that software is developed using agile methods, which rely on close team working and pair programming. Discuss the difficulties that this new policy might cause and how you might get around these problems. *(8 marks)*

Answer # 04:

The difficulties that this new policy might cause are:

There would definitely be communication gap among the team members. There are numerous benefits which are obtained through agile methods. If the company want to decide to close down many of their offices that were specialized in using agile methods they may face a many of difficulties. When a company is to close team and is divided they will be unable to have daily meetings, which can cause issues with communication, programming and many other issues, and if the company send project to their employer so the employer does not fulfil the requirement which the company wants and also communication gab would be created, productivity will slow down due to communication issues, and detecting errors would be quite difficult.

These problems are avoided by:

These problems can be avoided by creating merging offices together so pair programming and daily communication can be established. If that is not possible, a communication platform consisting of webcams, desktop viewing software, and microphones should be created to allow better communication.

Q.5: Identify and briefly describe four types of requirement that may be defined for a computer-based system. (6 marks)

Answer # 05:

Software requirements:

Requirement is a condition or capability possessed by the software or system component in order to solve a real world problem. The problems can be to automate a part of a system, to correct shortcomings of an existing system, to control a device, and so on.

Types of requirements for computer-based system.

System requirements are intended to communicate the functions that the system should provide. And every computer based systems consists of many requirements. They are:

1. User requirements:

These requirements are the natural language declaration and the services provided by the system and its operational limits.

2. System requirements:

An organized document containing a detailed description of system features, services, and operating restrictions. Explains that this needs to be implemented. This may be part of an agreement between the customer and the contractor.

3. Functional requirements:

These are the statements about which the system would like to provide services, how the system reacts to some input. Should act and how the system should behave in a particular situation.

4. Non-functional requirements:

Constraints on services or functions offered by the system such as punctuality, constraints on development process, quality, etc.

These requirements generally apply to the entire system and not to individual functions or services.

Q.6: Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system. (10 marks)

Answer # 06:

Use case:

Use case are to generate the desired output are usually a series of interactions between the user / admin and the system. The use case diagram is a graphical representation of all the issues that interact with the system. The use case diagram is part of a unified modeling language, also called UML.

Understanding the requirements for an ATM system:

Set of Use cases that are used in this ATM use case diagram to understand the requirements of the ATM are given below:

1. Insert ATM card
2. Enter pin
3. Perform required transaction

- Withdrawal
 - Deposit
 - Transfer
 - Change pin
4. Exit

Customer and Bank are treated as actor. Actors are the one who interacts with the system.

Withdraw cash:

- **Actors:** Customer, ATM, accounting system.
- **Inputs:** Customer's card, PIN, Bank account details.
- **Outputs:** Customer's card, Receipt, bank account details.
- **Normal operation:** Customer inputs his card into the machine He is prompted for a pin which is entered on the keypad if correct, he is presented with a menu of options. The withdraw cash option is selected. The customer is prompted with a request for the amount of cash required and enters the amount if there is sufficient funds in the account, the cash is dispensed, a receipt is printed and the account balance is updated. The card is returned to the customer who is prompted by the machine to take the card.

Exception:

- **Invalid card:** Card is retained by machine. Customer is advised to seek advice
- **Incorrect pin:** customer is requested to rekey Pin. If incorrect after 3attempts, card is retained by machine and customer is advised to seek advice
- **Insufficient balance:** Available balance is displayed to customer and allow customer to rener a valid amount.

Display balance:

- **Actor:** Customer, ATM, Accounting system.
- **Inputs:** Customer's card, PIN, Bank account details.

Outputs: Customer's card, Receipt, bank account details.

Normal operation: The customer authenticates using his card and associated PIN Customer select Display balance option. The Current balance of their account is displayed on the screen. Option is given to print the balance on a receipt. Other menu options displayed to customer or terminate the transaction. The card is returned to the customer.

Exception:

- **Invalid card:** Card is retained by machine. Customer is advised to seek advice
- **Incorrect pin:** customer is requested to rekey Pin. If incorrect after 3attempts, card is retained by machine and customer is advised to seek advice.

Print statement:

- **Actors:** Customer, ATM, Accounting system.
- **Inputs:** Customer's card, PIN, Bank account details.
- **Outputs:** Customer's card, Printed statement, bank account details.
- **Normal operation:** The customer authenticates using his card and associated PIN Customer select Print statement option. The last five transaction on their

account is printed. Receipt with transaction is issued to the customer. Other menu options displayed to customer or terminate the transaction. The card is returned to the customer.

Exception:

- **Invalid card:** Card is retained by machine. Customer is advised to seek advice.
- **Incorrect pin:** customer is requested to rekey Pin. If incorrect after 3 attempts, card is retained by machine and customer is advised to seek advice.

Change PIN:

- **Actors:** Customer, ATM, Accounting system.
- **Inputs:** Customer's card, PIN, Bank account details.
- **Outputs:** Customer's card, Printed statement, bank account details.
- **Normal operation:** The customer authenticates using his card and associated PIN. Customer select Change PIN option. He is prompted twice to input the new PIN. System validate that the input are the same.

Q.7: Suggest how an engineer responsible for drawing up a system requirements specification might keep track of the relationships between functional and non-functional requirements. (8 marks)

Answer # 07:

Functional requirements: Describe what the system will do. For example inputs and outputs.

Non-functional requirements: Describe the expectations but it is not concerned with the system. For example security.

It is a functional requirement: The user needs to search for the candidate list for the interview.

It is a non-functional requirement: That the search should return all the list of candidates who are attending the interview.

One way to track the relationship between functional and non-functional requirements may be to include some kind of flow chart. If other such requirements exist, this graph may move from one practical need to the next logical practical need with the possibility of diversification. Non-functional requirements needs can then be written along with the practical need with which they are associated. Passive needs can be repeated if they refer to more than one practical need. And other words: An engineer can produce a table or diagram of functional and non-functional requirements separately. The relationships between these two requirements are hard to manage because one might work off from another. First, listing and identifying system level non-functional requirements that are linked with functional requirements. Then listing the system requirements that are related for each functional requirement.