



*Summer 2020 Final-term*

*Software Engineering*

**Submitted By:** Daniyal Malik

ID# 13709  
BSSE

**Submitted To:** Engr. Ghassan Husnain

(Lecturer)

## **Question No: 01**

**Explain why the rapid delivery and deployment of new systems is often more important to businesses than the detailed functionality of these systems.**

**Ans:** Now a day's many businesses were willing to trade off software quality and compromise on requirements to achieve faster deployment of the software. In this way, Rapid development and delivery is the most critical requirement for software system. Because these business are operating in a changing environment. They have to respond to new opportunities and markets, changing economic conditions and the emergence of competing products and services. So, Rapid delivery focuses on the delivery of the system. It is good for a system that is required to show the result of the system.

And, it is good for business since the system can be used early if the essential functionality is available and be later improved as the user requirements change. Rapid delivery can make profit swiftly.

Rapid software development processes are designed to produce useful software quickly. At the same time, it is not good for critical-system development since the delivery of the system needs to be perfect and without failures.

## **Question No: 02**

**Explain how the principles underlying agile methods lead to the accelerated development and deployment of software.**

**Ans: Agile methods:**

Agile methods develop the system in increments by using the incremental development processes. These methods made the change of the system in the small increments depending on the requirements of the customer. In this, the customers are included in the development process. The agile methods focus on the software of the system instead of designing and documentation. These methods allow the developers to deliver the software quickly to the customer.

The principles of agile methods has contributed to the accelerated development and deployment of software in the following ways:

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 important principle to develop efficient software is to recognize the technical skill of the people working in the development process of the system. It is necessary to implement and deliver the product quickly 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.

### **Question No: 03**

**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.**

**Ans: Advantages of stories:**

1. They represent real situations that often happen so the system will support the most common user operations.
2. It is easy for users to understand and critique the stories.
3. They represent increments of functionality - implementing a story delivers some value to the user.

### **Disadvantages of stories**

1. They may be incomplete and their informal nature makes this incompleteness difficult to see.
2. They focus on functional requirements rather than non-functional requirements.
3. Representing cross-cutting system requirements such as performance and reliability is impossible when stories are used.
4. The relationship between the system architecture and the user stories is unclear so architectural design is difficult.

### **Question No: 04**

**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.**

**Ans: Use of technology:**

- Video conferencing
- Scrum master should be located with development team to be aware of everyday problems

- Product owner should visit developers to establish good relationships with them
- Real time communication through informal communication, I.E. instant message and video calls.
- Continuous integration, all team members can be aware of state of product at any time
- Common development environment for all teams

### **Problems:**

- Development requirement daily meetings but not possible in distributed environment
- Communication gap can occur between members
- Changes lead to slow the entire development of project
- Pair programming benefits detection and evaluation of errors.

### **Question No: 05**

**Identify and briefly describe four types of requirement that may be defined for a computer-based system**

**Ans: User Requirements** – User requirements are natural language statements that may be accompanied by diagrams that show both the services the system is expected to provide the users, as well as any constraints the system will operate under.

**System Requirements** – System requirements are more detailed descriptions of the software system's functions, services, and operational constraints. The exact implementation should be defined and this document can serve as part of a contract between the system buyer and the developers.

**Functional Requirements** – Functional requirements are the statements of the services that the system should provide, how the system handles specific input and how the system should behave in a given situation.

**Nonfunctional Requirements** – Nonfunctional requirements refer to the constraints on the system's services or functions. These can include industry standards that must be followed such as timing, security, etc. An example being that bank transactions most likely have to have encrypted communications for certain operations, if not all. These constraints usually effect the entire system rather than certain features or services.

## **Question No: 06**

**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.**

**Ans:** Given the variety of systems used, there can't be one unified set of use cases for all ATMs, but many share common functionality. A few examples are:

### **1. Withdraw funds:**

- Actors – user, ATM, user account/s
- Input – user card, user PIN, account select, withdrawal amount
- Output – user card, transaction receipt, account/s balance

### **Exceptions:**

- Card invalid – user advised to seek assistance; card retained
- Invalid PIN – transaction rejected after predetermined number of attempts; card retained; user advised to seek assistance
- Insufficient funds – transaction cancelled; user advised to seek assistance

### **Use case:**

- User inserts card into card reader. ATM prompts user to enter PIN. User enters PIN. If PIN is correct, ATM prompts user to select transaction type. User selects withdrawal. ATM prompts user to select account for withdrawal. User selects account. ATM prompts user to enter amount of funds for withdrawal. User enters amount for withdrawal. If sufficient funds exist for withdrawal, funds are

dispensed. ATM prompt asks user if they would like a receipt. User makes selection. If yes, receipt is printed. Card is returned to user.

## **2. Obtain Balance:**

- Actors – user, ATM, user account/s
- Input – user card, user PIN, account select
- Output – user card, transaction receipt, account/s balance

### **Exceptions:**

- Card invalid – user advised to seek assistance; card retained
- Invalid PIN – transaction rejected after predetermined number of attempts; card retained; user advised to seek assistance

### **Use Case:**

- User inserts card into card reader. ATM prompts user to enter PIN. User enters PIN. If PIN is correct,

ATM prompts user to select transaction type. User selects account balance. ATM prompts user to select account for balance inquiry. User selects account. ATM displays account balance for selected account. ATM prompt asks user if they would like a receipt. User makes selection. If yes, receipt is printed displaying account balance. Card is returned to user.

## **3. Deposit Funds:**

- Actors – user, ATM, user account/s
- Input – user card, user PIN, account select, cash/check for deposit
- Output – user card, transaction receipt, account/s balance

### **Exceptions:**

- Card invalid – user advised to seek assistance; card retained
- Invalid PIN – transaction rejected after predetermined number of attempts; card retained; user advised to seek assistance
- Deposit failure – transaction cancelled if user fails to input funds for deposit within predetermined amount of time

### **Use Case:**

- User inserts card into card reader. ATM prompts user to enter PIN. User enters PIN. If PIN is correct, ATM prompts user to select transaction type. User selects deposit. ATM prompts user to select account for deposit. User selects account. ATM prompts user to select cash or check deposit. User makes selection. If cash, ATM prompts user to insert cash into cash receptacle. ATM reads bills as they enter and counts total. ATM displays amount and asks user to confirm total. If check, ATM reads scans check and reads total. ATM displays check scan and total for deposit. ATM asks user to confirm total. User can confirm total or reject total. If user rejects total, ATM asks user to enter amount for deposit and confirm. ATM displays total deposit and new account balance. ATM prompt asks user if they would like a receipt. User makes selection. If yes, receipt is printed displaying account balance. Card is returned to user.

### **Question No: 07**

**Suggest how an engineer responsible for drawing up a system requirements specification might keep track of the relationships between functional and non-functional requirements.**

**Ans:** Functional requirements describe what the system will do.

Ex: inputs and outputs.

Non-functional requirements describe the expectations but it is not concerned with the system.

Ex: security

While drawing up a system requirements specification, an engineer might keep track of the functional and non-functional requirements by ensuring the following:

- The requirements needed to design meets the requirements such as compatibility, portability etc.
- Design the system so that it ensures the safety and security.
- Implementing the system in an efficient manner.



- The cost and time required for the development should not affect the design and implementation of the system

Here, the non-functional requirement defines what are the expectation to get out and the user requirements.

The functional requirement defines the use of the developer knowledge.

It does not conflict with each other.

The first step is to make the Systems Requirement Document.

It is engineer responsible to prepare documents to each functional and non-functional requirement.

- The engineer needs to prepare the document depending on this; Non-functional requirements need the natural language and functional requirements need the structured language to understand better.
- It gives the matrix that shows each requirement related to each other.
- It is very difficult to manage because the functional and non-functional requirements put efforts with each other on track of relationships.
- Non-functional requirements linked with functional requirements to list, identify the system levels that have related each other.
- The engineer needs to prepare the way to link the functional to non-functional to implement it.
- The functional requirements enforce the non-functional requirements that shall be recorded and tracked.

For example, The user needs to search for the candidate list for the interview.

- It is a functional requirement.

That the search should return all the list of candidates who are attending the interview.

- It is a non-functional requirement.

Therefore, it helps the engineer to avoid overlap and that relates to each other.

And it keeps track the relationships between functional and non-functional requirements.