IQRA NATIONAL UNIVERSITY

INU

DEFINING POSSIBILITIES, CREATING OPPORTUNITIES

Summer-2020 Final Term

Software Engineering

Submitted By: Mudassir Ahmad Khan
ID# 14086
BS(Software Engineering)

Submitted To: Engr. Ghassan Husnain
(Lecturer)

Date: September 25, 2020

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. (6 marks)

Answer:

In a conventional waterfall or specfication-based process the final software is usually delivered to the customer long after it was originally specified. In a business environment, this can cause problems. By the time the software is available for use, the original requirements may have changed drastically. Therefore, business systems focus on rapid software development and delivery.

Question No: 02

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

Answer:

The principles underlying agile development are:

1. Individual and interactions over processes and tools.
Overheads of formal communication and process assurance are avoided by letting the development team know what others are doing and taking advantage of individual skills. This means that the team can focus on the development of working software.

2. Working software over comprehensive documentation.
Time is not spent developing, checking and managing documentation. Time is focused on the development and testing of code.

3. Customer collaboration over contract negotiation
Rather than analyzing and negotiating what should be in a system contract, agile developers get feedback from customer’s directly during the development. This allows useful functionality to be developed and delivered earlier than would be possible if contracts were required.

4. Responding to change over following a plan
Being responsive to change is more effective than following a plan-based process because change is inevitable. The inflexibility of a plan causes significant overhead when changes are required and finished work may be discarded.

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. (6 marks)

Answer:

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. (8 marks)

Answer:

If the company decided to close down a number of offices that werespecialized in using agile methods they may face a multitude ofdifficulties. When a company is driven by a close team and is dividedthey will be unable to have daily meetings, which can cause issueswith communication, programming in pairs would not be possible, acommunication gab would be created, productivity will slow downdue to communication issues, and detecting errors would be quitedifficult. These problems can be avoided by creating merging officestogether so pair programming and daily communication can beestablished. If that is not possible, a communication platformconsisting of webcams, desktop viewing software, and microphonesshould be created to allow better communication.

Question No: 05

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

Answer:

Types of requirements for a computer based system: Generally, system requirements are includedto communicate the functions that the system should provide. And every computer based systemsconsists of many requirements. They are:

1. User requirements:

are an abstract description of services the system is expected to provide to system users, and the constraints under which they must operate.

2. System requirements:

are similar to user requirements but offer a more detailed description of the system's functions and services. These requirements are used by the system developers to develop the system's services.

3. Security Requirements:

require that a system user must be authenticated before they are granted access to the system. This keeps the integrity of the system intact and prevents unauthorized use of the system.

4. Dependability requirements:

this requires that the system acts as expected at all times the system is in use. This is a must because without dependability the system would be useless.

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. (10 marks)

Answer:

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. (8 marks)

Answer:

A simple way to keep track of the relationships between functional and non-functional requirements might be to first list the requirements separately in a columnar format. After completing each list, place the lists side by side and search for redundancies or dependencies. One could then highlight those requirements from each side that are related, using the separate colors or styles to denote those that are related on each side. This is a situation where diagramming could also be useful. One could create different diagrams for each component, and add the functional and non-functional requirements for each component, thereby grouping those that are related together.

Thank you!