Final Assessment│Summer 2023

Md. Shafayet Hossain

CSE - 21st Batch │ **Course Title:** Software Engineering

Course Code: CSI - 321 │ **ID:** 2121210071

Answer to the Question no- 1

(a)

Software Engineering

Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications. Software engineers apply engineering principles and knowledge of programming languages to build software solutions for end users.

(b)

The Waterfall Model

The waterfall model is the basic software development life cycle model. It is very simple but idealistic. Earlier this model was very popular but nowadays it is not used. But it is very important because all the other software development life cycle models are based on the classical waterfall model.

The waterfall model is useful in situations where the project requirements are well-defined and the project goals are clear. It is often used for large-scale projects with long timelines, where there is little room for error and the project stakeholders need to have a high level of confidence in the outcome.

The Validation Process in Waterfall Model

The Waterfall Model is a traditional and linear software development methodology that consists of several distinct phases, and validation is an essential part of this process. The validation process in the Waterfall Model typically occurs in the following manner:

1. **Requirements Gathering and Analysis:** The first phase involves gathering requirements from stakeholders and analyzing them to understand the scope and objectives of the project.
2. **Design:** Once the requirements are understood, the design phase begins. This involves creating a detailed design document that outlines the software architecture, user interface, and system components.

3. **Implementation:** The implementation phase involves coding the software based on the design specifications. This phase also includes unit testing to ensure that each component of the software is working as expected.
4. **Testing:** In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects.
5. **Deployment:** Once the software has been tested and approved, it is deployed to the production environment.
6. **Maintenance:** The final phase of the Waterfall Model is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.

(c)

Following are the characteristics of a good SRS document:

- **Clarity:** The SRS should be written in clear and concise language so that it is easily understandable by all stakeholders, including developers, testers, project managers, and clients.
- **Completeness:** It should include all necessary requirements, both functional (what the software should do) and non-functional (how it should perform), to ensure that the software meets the user's needs.
- **Consistency:** The SRS should be internally consistent, meaning that there should be no conflicting or contradictory requirements within the document.
- **Uniqueness:** Each requirement should be unique and not duplicated in the document. Duplicate requirements can lead to confusion and misinterpretation.
- **Use Cases and Scenarios:** The SRS may include use cases, scenarios, or flow diagrams to illustrate how the software will behave in different situations, helping to provide a more detailed understanding of the system's functionality.
- **Review and Approval:** The SRS should go through a formal review and approval process involving all relevant stakeholders to ensure that everyone agrees on the documented requirements.

Answer to the Question no- 2

(a)

Categories of Software Maintenance-

- **Corrective Maintenance:** This involves fixing errors and bugs in the software system.
- **Patching:** It is an emergency fix implemented mainly due to pressure from management. Patching is done for corrective maintenance but it gives rise to unforeseen future errors due to lack of proper impact analysis.
- **Adaptive Maintenance:** This involves modifying the software system to adapt it to changes in the environment, such as changes in hardware or software, government policies, and business rules.
- **Perfective Maintenance**: This involves improving functionality, performance, and reliability, and restructuring the software system to improve changeability.
- **Preventive Maintenance:** This involves taking measures to prevent future problems, such as optimization, updating documentation, reviewing and testing the system, and implementing preventive measures such as backups.

(b)

Black box testing and white box testing are two different approaches to software testing, and their differences are as follows:

Black box testing is a testing technique in which the internal workings of the software are not known to the tester. The tester only focuses on the input and output of the software. Whereas, White box testing is a testing technique in which the tester has knowledge of the internal workings of the software, and can test individual code snippets, algorithms and methods.

Testing objectives:
Black box testing is mainly focused on testing the functionality of the software, ensuring that it meets the requirements and specifications. White box testing is mainly focused on ensuring that the internal code of the software is correct and efficient.

Knowledge level:
Black box testing does not require any knowledge of the internal workings of the software, and can be performed by testers who are not familiar with programming

languages. White box testing requires knowledge of programming languages, software architecture and design patterns.

Testing methods:
Black box testing uses methods like equivalence partitioning, boundary value analysis, and error guessing to create test cases. Whereas, white box testing uses methods like control flow testing, data flow testing and statement coverage.

Scope:
Black box testing is generally used for testing the software at the functional level. White box testing is used for testing the software at the unit level, integration level and system level.

(c)

Types of Documents in SQA

Software Quality Assurance (SQA) involves various processes and activities to ensure that software products meet specified quality standards. As part of these activities, several types of documents are typically created and maintained to document and manage the quality assurance process. Here are some common types of documents in SQA:

- Quality Assurance Plan (QAP)
- Test Plan
- Test Cases
- Test Scripts
- Test Data
- Defect Reports
- Traceability Matrix
- SQA Audit Reports
- Configuration Management Documents
- Quality Metrics and Reports
- Release Notes
- Training Materials

Answer to the Question no- 3

(a)

Software Project Management

Software Project Management (SPM) is a proper way of planning and leading software projects. It is a part of project management in which software projects are planned, implemented, monitored and controlled.

(b)

Role and Responsibilities of a Software Project Manager

A software project manager is the most important person inside a team who takes the overall responsibilities to manage the software projects and plays an important role in the successful completion of the projects. Most of the managers take responsibility for writing the project proposal, project cost estimation, scheduling, project staffing, software process tailoring, project monitoring and control, software configuration management, risk management, managerial report writing and presentation and interfacing with clients. The task of a project manager are classified into two major types:

1. Project planning
2. Project monitoring and control

- **Project planning-**

Project planning is undertaken immediately after the feasibility study phase and before the starting of the requirement analysis and specification phase. During the project planning the project manager performs the following activities:

1. **Project Estimation:** Project Size Estimation is the most important parameter based on which all other estimations like cost, duration and effort are made.
   - **Cost Estimation:** Total expenses to develop the software product is estimated.
   - **Time Estimation:** The total time required to complete the project.
   - **Effort Estimation:** The effort needed to complete the project is estimated.
2. **Scheduling:** After the completion of the estimation of all the project parameters, scheduling for manpower and other resources is done.
3. **Staffing:** Team structure and staffing plans are made.
4. **Risk Management:** The project manager should identify the unanticipated risks that may occur during project development risk, analyze the damage that might cause these risks, and take a risk reduction plan to cope with these risks.

- **Project monitoring and control-**

This includes making several other plans such as quality assurance plans, configuration management plans, etc.

1. **Lead the team:** The project manager must be a good leader who makes a team of different members of various skills and can complete their individual tasks.
2. **Motivate the team-member:** One of the key roles of a software project manager is to encourage team members to work properly for the successful completion of the project.
3. **Tracking the progress:** The project manager should keep an eye on the progress of the project.
4. **Liaison:** The project manager is the link between the development team and the customer. Moreover, the project manager checks whether the project is fulfilling the customer's requirements or not.
5. **Monitoring and reviewing:** Project monitoring is a continuous process that lasts the whole time a product is being developed.
6. **Documenting project report:** The project manager prepares the documentation of the project for future purposes.

(c)

Advantages and Disadvantages of Iterative Model

Advantages of Iterative Model:

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

Disadvantages of Iterative Model:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

Answer to the Question no- 4

(a)

Software Quality

Quality software refers to a software which is reasonably bug or defect free, is delivered in time and within the specified budget, meets the requirements and/or expectations, and is maintainable.

The Importance of Software Quality

The importance of software quality cannot be overstated in today's technology-driven world. Software quality refers to the degree to which a software product meets its specified requirements, is free from defects or errors, and satisfies the needs and expectations of its users. Here are several reasons why software quality is crucial:

User Satisfaction: High-quality software ensures that users have a positive experience, leading to increased user satisfaction. Satisfied users are more likely to continue using the software, recommend it to others, and remain loyal to the brand.

Reliability: Quality software is reliable and trustworthy. It operates consistently and predictably without unexpected crashes, errors, or data corruption.

Cost Savings: High-quality software reduces the need for costly post-release bug fixes and support.

Competitive Advantage: High-quality software can set a company apart from its competitors by providing a better user experience and more reliable functionality.

Reduced Risk: Low-quality software can pose significant risks to an organization. It can lead to data breaches, security vulnerabilities, legal issues, and reputational damage. Ensuring software quality helps mitigate these risks.

Customer Retention: High-quality software can lead to increased customer loyalty.

Reputation and Brand Image: Software quality reflects on an organization's reputation and brand image. Poor-quality software can damage a company's image, while high-quality software enhances its reputation.

(b)

Major Software Quality Assurance Activities:

1. **SQA Management Plan:**
   Make a plan for how you will carry out the SQA through out the project. Think about which set of software engineering activities are the best for project.

2. **Set The Check Points:**
   SQA team should set checkpoints. Evaluate the performance of the project on the basis of collected data on different check points.

3. **Multi-testing Strategy:**
   Do not depend on a single testing approach. When you have a lot of testing approaches available use them.

4. **Measure Change Impact:**
   The changes for making the correction of an error sometimes re introduces more errors keep the measure of impact of change on project. Reset the new change to change check the compatibility of this fix with whole project.

5. **Manage Good Relations:**
   In the working environment managing good relations with other teams involved in the project development is mandatory. Bad relation of SQA team with programmers team will impact directly and badly on project. Don't play politics.