SAHARO ALI – SCT221-C004-0405/2022
NAEEMA SULEIMAN SHEIKH – SCT221-C004-0414/2022 SOFTWARE ENGINEERING 1
IBRAHIM ZAKIYA – SCT221-C004-0722/2022

GROUP ASSIGNMENT

ASSIGNMENT ONE:

Read carefully and understand the case study below to answer the two questions. Be as innovative as possible.

The railway reservation system functions as follows;
The passenger is required to fill in a reservation form giving detail of his journey. The counter clerk ensures whether the place is available. If so, entries are made in a register, tickets are prepared, amount is computed and cash is accepted.
A booking statement is prepared in triplicate format from the reservation register. One copy is retained as office copy, the other is pasted to the compartment and third is passed on to the train conductor. Besides booking statement, cash statement is prepared at the end of each shift.

Prepare System Require Specification and system specification for above system problem.

<u>System Requirements Specification (SRS)</u>

1. Introduction

The Railway Reservation System is designed to manage and facilitate the booking of train tickets. It allows passengers to make reservations, checks seat availability, and handles the ticketing process.

2. Functional Requirements

- Reservation Form

The system should provide a user-friendly reservation form.
Passengers must input journey details from the destination to the time.

- Seat Availability

The system must check the available seats for the requested journey.
The clerk should be able to confirm if there are seats available before proceeding with the booking.

- Ticket Booking

After confirming seat availability, the system should prepare a ticket.
The system must compute the ticket fare based on the class of travel.

- Booking Statement

A booking statement should be prepared including the details of the reservation.
One copy is retained as the office copy.
Another copy is pasted inside the train compartment.
The third copy is handed over to the train conductor.

3. System Architecture

The Railway Reservation System should consist of the following components:

User Interface: Provides a reservation form for passengers.
Seat Availability Module: Checks seat availability.
Ticketing Module: Handles ticket booking and fare calculation.
Booking Statement Generator: Prepares booking statements.
Cash Statement Generator: Prepares cash statements.
Database: Stores reservation and ticketing data.

4. Rules and Constraints

The system must comply with railway rules and regulations. The hardware and software infrastructure should meet minimum system requirements.

System Specification

The System Specification document would delve into technical details such as hardware and software requirements, database schema, and architectural diagrams. It would provide a blueprint for how to implement the system based on the requirements outlined in the SRS.

ASSIGNMENT TWO

Group One: Discuss the following software testing techniques (walkthroughs, reviews and inspections, dynamic testing, Traceability matrices. Debugging environments)

- Walkthroughs:

  Walkthroughs are collaborative and manual processes where developers and testers, review a software product or specific components.

  The aim is to understand the design or documentation and identifying potential issues or ways to improve.

  Walkthroughs used to ensure the team has a common understanding of the software's functionality
- Reviews and Inspections:

  Reviews and inspections are formal and systematic methods for examining software requirements, design or code.

These techniques involve a structured process, documented checklists and a team of individuals who analyse thoroughly the software artifacts for defects and compliances.

The objective is to find defects early in the development process, improving overall software quality.

- Dynamic Testing:

  Dynamic testing involves the execution of the software to assess its functionality.

  Test cases are designed to ensure the software performances are as expected and to identify defects or issues.

  This category includes various testing types, such as unit testing, integration testing, system testing and user acceptance testing

- Traceability Matrices:

  Traceability matrices are documents that maintain links between different phases of the software development life cycle

  They help ensure that each requirement has corresponding test cases and that all defects are linked back to specific requirements or design elements.

  Traceability matrices facilitates traceability, impact analysis and requirements coverage.

- Debugging Environments:

  Debugging environments are tools used by developers to identify and fix defects in software.

  These environments typically include integrated development environments (IDEs), code editors and debugging tools .

  Debugging is important in the development process to find and correct programming errors.

Group Two:

Discuss Software Quality Evaluation – Problems, Software standards, Certification, Software Tools support for Systems Engineering

Problems

a) Subjectivity: Assessing software quality is subjective to some extent. What might be considered high quality by one person or team may not be the same for another.

b) Complexity: Software systems are often complex, making it challenging to assess their quality comprehensively. The interconnectedness of components and the potential for unforeseen interactions create difficulties.

c) Changing Requirements: Software projects often deal with evolving requirements. Evaluating quality becomes more complex when requirements are not well-defined or change frequently

d) Resource Constraints: Limited resources, such as time and budget, can impact the depth of quality evaluation. Comprehensive testing may be limited due to resource constraints.

Software Standards

Software standards provide guidelines and best practices for the development, maintenance, and evaluation of software. They ensure consistency, reliability, and interoperability. Examples of software standards include ISO/IEC 25010 for software product quality, ISO/IEC 12207 for software life cycle processes, and ISO/IEC 27001 for information security.

Certification:

Software certification is the process of evaluating and verifying that a software product or system
meets specified standards, requirements, or regulations. Certification provides confidence to users and customers that the software is reliable and of high quality. Examples of software certifications include CMMI (Capability Maturity Model Integration) and ISO/IEC 9001.

Software tools support

a) Requirements Management Tools: These tools help in defining, managing, and tracking software requirements, ensuring that they are clear, complete, and consistent.

b) Testing and Quality Assurance Tools: Tools like automated testing frameworks, static code analysis, and continuous integration systems help assess and improve software quality.

c) Configuration Management Tools: These tools ensure version control, change tracking, and consistency in software development.

d) Modeling and Simulation Tools: Tools like UML modeling software assist in system design, architecture, and simulation, enabling better understanding of complex systems.

e) Project Management Tools: Project management and collaboration tools support project planning, tracking, and coordination, contributing to successful systems engineering and quality evaluation.

Group Three: Discuss CASE tools as used in Web engineering techniques and process, standards and guidelines. CASE (Computer-Aided Software Engineering) tools play a vital role in the web engineering process, aiding in various stages of web application development. Here's a discussion on their usage, process, standards, and guidelines:

Usage of CASE Tools in Web Engineering:

Requirements Analysis: CASE tools assist in gathering, documenting, and managing user requirements for web applications. They can create use case diagrams, data flow diagrams, and entity-relationship diagrams.

Design and Modelling: These tools help in creating visual representations of the web application's architecture and data models. UML diagrams, such as class diagrams and sequence diagrams, can be generated using CASE tools.

Code Generation: Some CASE tools offer code generation features, which can significantly speed up the development process by producing boilerplate code and reducing manual coding efforts.

Testing and Quality Assurance: They facilitate test case management, tracking, and reporting, ensuring the quality of web applications through various testing phases.

Documentation: CASE tools help in generating documentation for web projects, including technical specifications, user manuals, and system documentation.

Web Engineering Process:
CASE tools are integrated into the web engineering process, which typically includes phases like requirements analysis, design, implementation, testing, deployment, and maintenance. Throughout these phases, CASE tools assist in project management, version control, and collaboration among team members.
Standards and Guidelines:
CASE tools adhere to various standards and guidelines to ensure consistency and compatibility:
UML (Unified Modeling Language): For modeling and diagramming.
Software Development Life Cycle (SDLC): Following established phases and best practices.
Web Development Frameworks: CASE tools should support popular web development frameworks, such as Django, Ruby on Rails, or React.
Security Standards: Adhering to security standards like OWASP (Open Web Application

Security Project) to ensure secure web application development.

Group Four: Discuss Process Improvement (PI): - Quality and process standards and guidelines

Defining Standards and Guidelines:

Quality and process standards provide a clear set of criteria and expectations for how work should be done. These standards may come from various sources, including industry best practices, regulatory requirements, or internal organizational policies.

Ensuring Consistency:

Standards and guidelines help ensure that processes are executed consistently across the organization. This consistency reduces the risk of errors, defects, and variations in output.

Benchmarking:

Quality and process standards often serve as benchmarks against which an organization can measure its performance. By comparing current processes and outcomes to established standards, organizations can identify areas for improvement.

Identifying Weaknesses:

By adhering to quality and process standards, organizations can identify weaknesses or gaps in their current processes. These standards highlight areas that may require improvement or modification.

Root Cause Analysis:

When a process deviates from established standards, it prompts organizations to conduct root cause analysis to understand why the deviation occurred. This analysis can uncover systemic issues that need to be addressed to prevent future deviations.

Continuous Monitoring:

Quality and process standards require continuous monitoring and assessment. This ongoing evaluation helps organizations maintain the desired level of quality and make adjustments as needed.

Compliance and Regulatory Requirements:

In some industries, adherence to specific quality and process standards is required by law or regulatory authorities. Failing to comply with these standards can result in legal and financial consequences.

Customer Satisfaction:

Meeting or exceeding quality standards is often directly related to customer satisfaction. By consistently delivering high-quality products or services, organizations can build trust and loyalty among their customers.

Cost Reduction:

Adhering to well-defined process guidelines can lead to cost reduction. By eliminating waste, redundancy, and inefficiencies in processes, organizations can optimize resource utilization and reduce operational expenses.

Documentation and Training:

Quality and process standards and guidelines provide a basis for creating documentation and training materials. These resources help employees understand how to perform their tasks effectively and consistently.