A red and purple logo

Description automatically generated

**Course Code Course Name**

ICT1214-Software Testing

**Activity Name**

Pre-assignment

Lanka Nippon Biztech Institute

UOG Batch 05

K.L.Y.M. Rupasingha

UOG0523032

**Completion of Course(Without Last step)**

A screenshot of a computer

Description automatically generatedWeek 1

A screenshot of a computer

Description automatically generatedWeek 2

**01.Explain the importance of Software testing ?**

1. Ensures Quality: Identifies bugs and ensures the software meets quality standards.

2. Prevents Costly Issues: Detects problems early, reducing the cost of fixing them later.

3. Improves Reliability: Ensures the software performs well under various conditions.

4. Protects Users: Prevents critical failures, especially in industries like healthcare and aviation.

5. Enhances User Experience: Ensures the software is functional and user-friendly.

6. Ensures Compliance: Verifies adherence to industry standards and regulations.

**02. Explain the role of software testing in the software development lifecycle**

The role of software testing in the Software Development Lifecycle (SDLC) is crucial to ensuring the quality and reliability of the software product. Testing is integrated into every phase of the SDLC, from requirements to maintenance and serves multiple functions:

1. Requirements Phase: Testing validates and verifies that the requirements are clear, complete, and testable, helping to ensure that stakeholders have defined accurate expectations.

2. Design Phase: Testers review design documents for accuracy and consistency, providing feedback on potential risks that may arise during implementation and testing.

3. Implementation Phase: Testers create test cases and perform unit testing to ensure individual software components function correctly.

4. Testing Phase: Testers execute test cases to identify defects, working with developers to ensure all bugs are fixed before release.

5. Deployment and Maintenance Phases: Testers verify successful deployment and continue to test new features, bug fixes, and updates to maintain software quality.

**03. Find an example of real-world software failures and their consequences.**

Knight Capital Group (2012)

- Failure: Knight Capital Group, a large American financial services firm, experienced a financial catastrophe due to a software bug.

- Cause: They released a new software version to a computer server without removing the outdated one, causing both programs to run simultaneously. This led to millions of flawed trades within 45 minutes.

- Consequences: The company's stocks plunged, resulting in a loss of $440 million. Knight Capital nearly went bankrupt and had to seek emergency funding.

- Lesson: This issue highlights the importance of thorough software testing and proper version control, especially in the financial sector where transactions are high-speed and high-volume.

**04. Describe the four levels of testing in the SDLC.**

1. Unit Testing

- Overview: This level involves testing individual components or units of a software application.

- Objective: The goal is to verify that each part functions as intended when isolated.

- Example: Testing a small function to ensure that adding two numbers, like 2 and 3, returns 5.

- Challenges: Identifying edge cases and isolating dependencies within the code.

2. Integration Testing

- Overview: This phase tests the interaction between combined software units or components.

- Objective: The aim is to detect faults in how these units work together.

- Example: Checking if the front-end of a website properly communicates with the back-end database during a user login.

- Challenges: Ensuring smooth communication between different components, such as the front-end and back-end.

3. System Testing

- Overview: This level tests the complete integrated system to verify it meets specified requirements.

- Objective: It assesses both functional and non-functional requirements, ensuring the system performs as expected and efficiently.

- Example: Testing an application to confirm it fulfills all specified requirements and operates well under various conditions.

- Challenges: Evaluating both what the system does and how well it performs against its requirements.

4. Acceptance Testing

- Overview: This final level assesses the software for acceptability and readiness for delivery.

- Objective: The goal is to ensure the software meets the needs and expectations of end users.

- Example: Testing a mobile app’s user experience, responsiveness, and security before release.

- Challenges: Gathering feedback from a diverse group of users to ensure the software meets broad expectations.

**05.  Explain functional and non-functional testing types. Instructions.**

Functional testing focuses on verifying whether the software functions and features work as per the specified requirements. It ensures that the software behaves as expected based on its functional specifications. Examples include checking if the login functionality of a website works or whether data is accurately retrieved from the database. This type of testing validates the application's conformance to requirements by testing against different user scenarios.

On the other hand, non-functional testing examines aspects like performance, usability, and security. It evaluates how well the software operates under various conditions. For instance, performance testing measures the system's response time under high traffic, while usability testing checks the user-friendliness of the interface. Security testing focuses on identifying vulnerabilities and ensuring sensitive data is protected. Non-functional testing ensures that the software not only works but performs optimally across different environments and scenarios.