**STLC & QA Testing**

**1.List down all the Models of SDLC**.

SDLC models is lifecycle models which defined and designed to follow certain rules and regulations while developing software.

**List of Models:**  
Waterfall Model

Prototype Model

Iterative Model

Relation Unified Process

Time Boxing Model

Spiral Model

V-Model

Extreme Programming

Agile Program.

**2.What is STLC? Also, Explain All Stages of STLC.**

STLC stands for Software Testing Lifecycle. STLC is a sequence of different activities performed by the testing team to ensure the quality of the software or the Product.

**Stages of STLC:**

**1.Requirement Analysis:** In this initial stage, the testing team closely analyses the software requirements and documentation to gain a deep understanding of what the software is expected to do.

**2.Test Planning:** During this phase, the testing team creates a detailed test plan that includes test objectives, scope, resources, schedule, and the overall strategy for testing.

**3.Test Analysis:** The test analysis, which includes hardware, Software, and network Configurations, is set up according to the test plan’s specifications. This ensures that the testing team has a controlled analysis to execute the tests.

**4.** **Test** **Design:** In this stage, the testing team designs the test cases and test scripts based on the requirements and the test plan. Test cases outline specific test scenarios, including input data, expected outcomes, and any preconditions. Test data and test environment setup procedures are also defined during this phase.

**5.Test Execution:** This is where the actual testing takes place. Testers execute the test cases and scripts in the defined test environment. They record the test results, including any defects or issues discovered during testing. Automated testing tools may be used to expedite this phase.

**6.Test Closure**: Once all planned tests have been executed, and the software meets the predefined exit criteria, the testing team prepares a test summary report.

3.As a test lead for a web-based application, your manager has asked you to identify and

explain the different risk factors that should be included in the test plan. Can you provide

a list of the potential risks and their explanations that you would include in the test plan?

1. Performance and Scalability Risk:  
This risk relates to the application’s ability to handle a high volume of concurrent users or large datasets efficiently. It includes concerns about response times, load balancing, and database performance.

2. Security Risk:  
Security risks involve vulnerabilities that could lead to data breaches, unauthorized access, or other security issues. This includes concerns about authentication, authorization, and data encryption.

3. Compatibility Risk:  
Compatibility risks involve the application’s ability to function correctly on various browsers, devices, and operating systems. Ensure compatibility testing covers different combinations.

4. Integration Risk:  
This risk pertains to the integration of the web application with other systems or third-party services. Issues may arise related to data exchange, API compatibility, or version conflicts.

5. Data Loss and Recovery Risk:  
There’s a risk of data loss due to system failures or unexpected errors. A plan for data backup, recovery, and disaster response should be in place.

6. Usability Risk:  
Usability risks involve the user-friendliness of the application. Issues like poor navigation, unclear instructions, or unintuitive user interfaces can negatively impact user satisfaction.

7. Regression Risk:  
Changes or updates to the application may introduce unintended side effects or regressions in existing functionality. Regression testing should be comprehensive to mitigate this risk.

4.Your Team Lead (TL)asked you to explain the difference between quality assurance (QA) and quality control {QC) responsibilities. while QC activities aim to identify defects in actual products your TL is interested in processes that can prevent defects How would u explain the distinction between QA and QC responsibilities to your boss?

Quality Assurance (QA) and Quality Control (QC) are two essential components of the quality management process in any organization, but they focus on different aspects and have distinct responsibilities:

**1.Quality Assurance (QA):**  
Preventive in Nature: QA is a proactive approach aimed at preventing defects and ensuring that processes are designed and executed correctly from the beginning. It’s focused on improving processes and methodologies.

**\*Process-Oriented**: QA involves defining and implementing standards, guidelines, and best practices for the entire development or production process. This includes establishing process documentation, training, and process improvement initiatives.  
Continuous Improvement: QA teams work on optimizing processes to reduce the likelihood of defects or issues occurring in the first place. They may conduct process audits and reviews to identify areas for improvement.  
Long-Term Focus: QA is concerned with the long-term stability and consistency of the process, which helps maintain product quality over time.

**2. Quality Control (QC):**  
Detective in Nature: QC is a reactive approach primarily concerned with identifying defects or issues in the final product or output. It involves inspecting and testing the end product to ensure it meets the predefined quality standards.  
**\*Product-Oriented**: QC activities are focused on the end result, such as products, services, or deliverables. This can include testing, inspections, and reviews of the final output.  
Immediate Feedback: QC provides immediate feedback on the quality of the product and helps in making decisions about whether it meets the required quality standards or needs rework.  
Short-Term Focus: QC is concerned with ensuring that the current output is of acceptable quality. It doesn’t necessarily address the root causes of defects in the process.

5.Difference between Manual Testing and Automation Testing?

1. **Execution Process**:

**Manual Testing**: In manual testing, human testers interact with the software application, following test cases and using various inputs to evaluate its behaviour. Testers perform actions like clicking buttons, inputting data, and validating results manually.

**Automation Testing**: In automation testing, testing tools and scripts are used to execute predefined test cases automatically. Testers write scripts to automate repetitive and time-consuming test scenarios, and testing tools execute these scripts.

**2. Speed and Efficiency:**

**Manual Testing**: Manual testing can be time-consuming and less efficient, especially for repetitive and regression testing. Testers may spend a significant amount of time executing test cases manually.

**Automation Testing**: Automation testing is faster and more efficient for repetitive test cases. Once automation scripts are created, they can be executed quickly and repeatedly, allowing for rapid feedback on changes to the application.

**3.Human Judgment**:

**Manual Testing**: Manual testers can use their creativity and judgment to identify unexpected issues, usability problems, and visual defects. They can adapt to changing test scenarios on the fly.

**Automation Testing**: Automation is rigid and follows predefined scripts. It cannot detect issues that are not explicitly programmed into the scripts. Human judgment and intuition are not a part of automated testing.

**4. Exploratory Testing**:

**Manual Testing:** Manual testers can perform exploratory testing to discover new issues or to investigate complex scenarios that are difficult to automate. This is particularly valuable for uncovering usability and edge-case issues.

**Automation Testing**: Automation is not well-suited for exploratory testing, as it requires predefined test cases. Exploratory testing relies on a tester’s creativity and adaptability, which automation lacks.

**5. Initial Investment:**

**Manual Testing:** Manual testing typically requires lower initial investment as it doesn’t involve the creation of automation scripts and the setup of testing tools. Testers need domain knowledge and testing skills.

**Automation Testing**: Automation testing requires an initial investment in terms of time and resources to develop and maintain automation scripts and infrastructure. However, it can yield significant long-term benefits in terms of regression testing and repetitive tasks.

**6. Maintenance**:

**Manual Testing**: Test cases are executed manually, so there’s minimal maintenance effort. Testers need to update test cases as the application evolves or requirements change.

**Automation Testing**: Automated scripts require regular maintenance to keep them up-to-date with changes in the application. Maintenance can be time-consuming, especially for complex applications.