JAVA INSTITUTE
for Advanced Technology
INNOVATIONS FOR THE KNOWLEDGE SOCIETY
ISO 9001:2008 Certified Institute

Java Institute for Advanced Technology

UNIT NAME: SOFTWARE ENGINEERING II (SOFTWARE TESTING, QUALITY ASSURANCE & MAINTENANCE)

UNIT ID: HF2W 04

ASSESSMENT NAME: RESEARCH ASSIGNMENT

NAME: M.R.P.N.THARUKSHA RAJAPAKSHA

STUDENT ID: 2019/2020/CO/SE/I2/029

SCN NO: 207977608

NIC: 200019401866

BRANCH: JAVA INSTITUTE, COLOMBO

Answer All questions

01.

I. What is STLC? List down all the steps of STLC?

STLC (Software Testing Life Cycle) is a procedure for testing software and ensuring that quality criteria are satisfied. The tests are carried out in a methodical manner throughout numerous periods. During product development, the steps of STLC can be repeated until the product is considered ready for release.

The steps of STLC

1. Requirement Analysis
2. Test Planning
3. Test Case Development
4. Environment setup
5. Test Execution
6. Test Cycle Closure

II. Briefly explain the first three test levels used in software testing?

1. Unit Testing
   A unit is the most minor tested component of a system or application that can be built, liked, loaded, and executed. This type of testing helps to test each part independently. The goal is to test each component of the software separately. Determines whether an element meets its capabilities or not. Developers are responsible for this type of testing.

2. Integration Testing
   Integration is merging. In this test step, separate software modules are assembled and evaluated as a group to ensure that the integrated system is capable of testing the systems. Integrating testing ensures that data flows from one module to the next. These types of tests are conducted by testers.

3. System Testing
   System testing is performed on a fully integrated system. It enables the assessment of compliance with system requirements. It analyzes the complete interaction of the components. Load, performance, reliability, and security testing are all part of the process. System testing is often the final test to ensure that the system meets specifications. It assesses both functional and non-functional testing requirements.

III. Write short notes on the following terms:

a) Test case
   A test case is a document containing a set of test data, preconditions, expected results, and postconditions prepared for a given test instance to verify compliance with a

particular requirement. Test cases act as the starting point of the testing process, and after applying a set of input values, the application has a specified result and exits the system at some end state, also known as the execution postcondition.

b) Software bug
A software bug is a problem that causes a program to crash or produce incorrect results. The situation is due to inadequacy or faulty logic. A bug is an error, mistake, flaw, or fault that can cause failure or deviation from intended results. Most bugs are caused by human mistakes in the source code or its design. A program is said to be buggy when it contains a large number of bugs that impair its functionality and produce incorrect results.

c) Test Automation
Automated testing is a software testing approach that compares the actual result with the predicted result. This can be done using a test script or other automated testing tool. Test automation is used to automate repetitive operations and other difficult-to-perform testing tasks.

d) Quality Audit
An independent assessment of a software product or process to determine conformance with specifications, standards, contractual obligations, or other criteria is called a software audit.

Any of the following can be audited in the software field:

- A software quality assurance process that the program is tested for quality.
- A Physical Configuration Audit (PCA) is a formal examination to validate the manufacturing basis of a configuration item.
- A software license audit is when a software user is audited for license compliance.

02.

I. What is UAT Testing?

UAT, often called application testing, is the final stage of any software design or update lifecycle before going live. It is the final stage of any design process to ensure that the software performs as intended in real-world settings. Real users test the program in real-world scenarios to see if it's achieving what it's supposed to do, verify updates, and analyze adherence to their organization's business needs. The primary goal of acceptance testing is to validate the entire business flow.

II. Briefly explain about the Alpha testing and Beta testing?

1. Alpha testing
   Alpha testing is a type of software testing that is done to find bugs before the product is released to real customers or the general public. This is a type of user acceptance test. This is called alpha testing because it is done during the software development process, near the conclusion. Alpha testing is often performed by in-house software engineers or quality assurance staff. It is the final level of testing before the program is released into the real world.

2. Beta testing
   Beta testing is done by real users in the real world. This is also a type of user acceptance testing. A beta version of the program is distributed to a small number of production end users to obtain feedback on product quality. Beta testing reduces the likelihood of product failure and improves product quality through customer validation. It is the final test performed before delivering a product to customers. One of the primary benefits of beta testing is direct client feedback.

III. How can you conduct risk analysis?

Risk analysis in software testing is a methodology for analyzing and measuring software risk. Traditional software testing often focuses on very simple functional testing. A software vulnerability analysis checks for code violations that could jeopardize the code's stability, security, or efficiency.

Implementing risk analysis during software testing usually requires careful examination of the source code to see how it interacts with other components of the overall program. This assessment examines various code components and maps how the code interacts. Transactions can be identified and analyzed using this map. Architectural and structural guidelines can be used to map to determine where software defects exist and which ones are the most critical given the application's transactions.

IV. Explain the significant of software testing plan including 4 components of test plan?

A test plan is a document that describes all upcoming test operations. It is created at the project level and outlines the work items to be tested, how they will be tested, and the distribution of test types among testers. A test manager will prepare a test plan before the commencement of testing. In every organization, after starting a new project, the team's test manager creates a test plan before the tester starts testing.

Importance of a test plan

- The test plan is important because it guides thinking. This is like a rulebook to be obeyed.
- The test plan documents important components such as the test schedule, test strategy, and test scope so that the management team can evaluate and reuse them for later comparable projects.
- The test plan helps developers, business managers, customers, and others understand the test specifications as they relate to the outside world.
- The test plan helps establish the efforts required to ensure the quality of the software application being tested.

Components

1. Estimate: An estimate refers to the amount of money and time that will be invested in the process of carrying out test activities.

2. Schedule: A summary of the schedule including the milestones to be met as well as the time period for completing the test activities.

3. Test Environment: The test environment is a significant consideration in testing. Test environment components such as hardware, software, network, etc., complete configuration method, and any relevant test tools should be specified.

4. Staffing and Training Requirements: Specify the type of training that should be provided to the staff or team members to hone their skills or provide the skills needed to perform test activities.

03.

I. What are the categories of defects?

- Errors of Clarity: The most common error in natural language. This mistake occurs as a result of miscommunication between the developer and the client. It takes more time to move from requirements to program.

- Errors of Omission: As the name suggests, an error of omission is anything that happens by chance. Something omitted or executed is called an omission.

- Errors of Commission: Commission refers to a given direction or order. The mistake in commission now indicates an error in command or instruction.

- Errors of Speed or Capacity: This suggests that the program is functioning properly but not within an appropriate time frame. This is a speed error. When it comes to memory capacity, it can be significant.

II. What is the difference between a defect and a failure?

A defect is a difference between actual and expected output. During the development process of SDLC, the defect is found by the testers and fixed by the developers.

A failure is a collection of multiple defects that eventually lead to software failure and loss of information in essential modules, making the system inaccessible. The problem is discovered by the test engineer during the SDLC development cycle.

III. Are there more defects in the design phase or in the coding phase? Explain.

The design phase is more prone to mistakes than the implementation phase. One of the most common design flaws is that the product fails to meet overall customer needs. The second point is that poor architectural and technical decisions make the following phase, execution, more prone to flaws. The design phase is the most critical to test as it drives the execution phase. A thorough review can help with design phase testing. On average, 60% of faults arise during the design phase, whereas 40% occur during the execution phase.

IV. What is a test log?

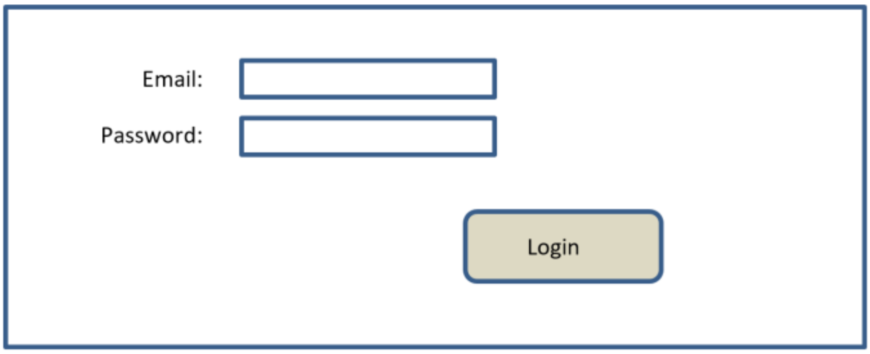
A test log provides a complete overview of the entire test run and identifies which tests succeeded and which failed. Furthermore, the test log provides data and information about various test activities, such as the source of problems and reasons for failed operations. The purpose of this report/document is to allow post-execution diagnosis of software problems and bugs.

Testers are critical in producing test logs, which should be produced when tests are conducted or when the team implements test scripts. Also, in addition to providing information about various tests, the test log includes photographs of the test program, URLs to files, and various other items.

V. Explain requirement traceability and its importance?

Requirements Traceability allows product teams to correlate a single requirement and other requirements, both forward and backward, with all related project artifacts, so that anyone can understand how activities relate to a requirement and vice versa, at any time during development. Also known as Live Traceability, this feature promotes team communication and allows for the quick discovery of potential production issues.

04.

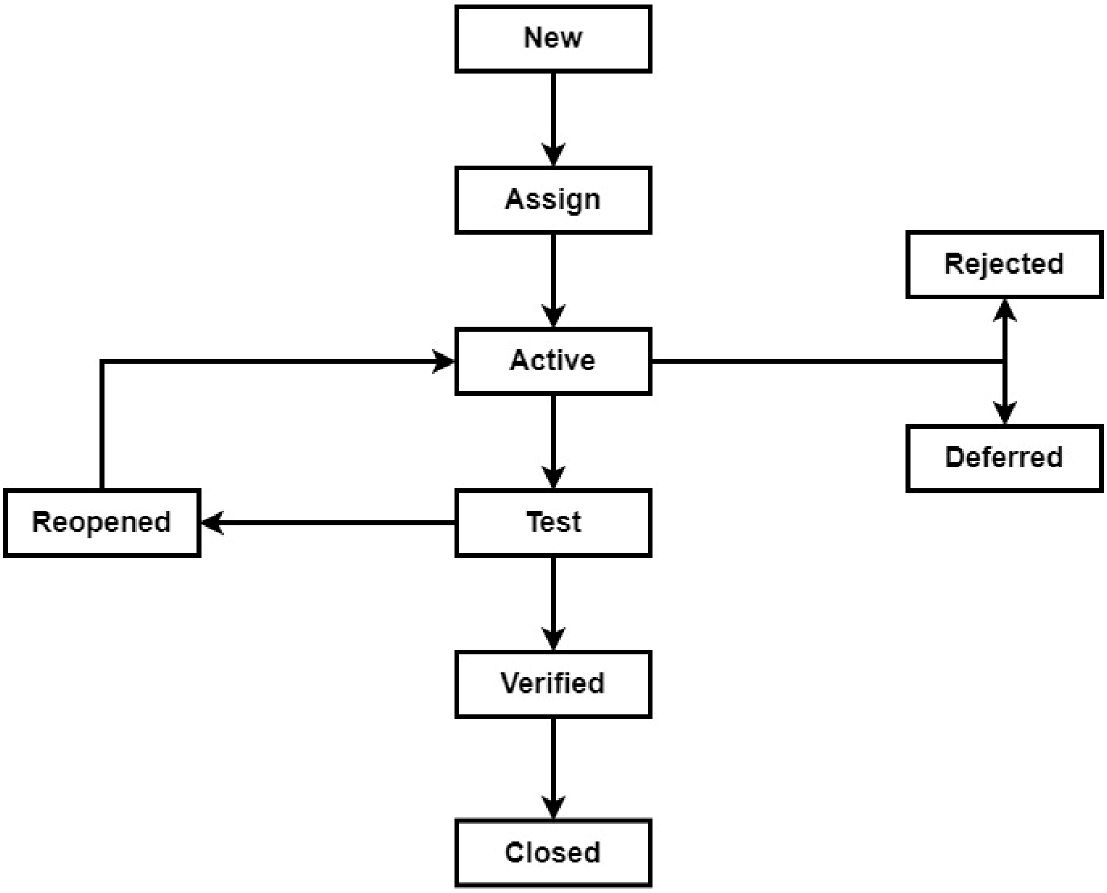


I. Write down 5 Test cases for the above Login Function.

Test Case No.	Test Case Name	Description	Pre-Condition	Test Data	Step Name	Step Description	Expected Result
1	Verify Email	Email Verification Passed		email@gmail.com	Enter Email	Verification Passed	Show the email input field border in green color
2	Verify Email	Email Verification Failed		email@gmail.com	Enter Email	Verification Failed	Show the email input field border in red color
3	Verify Password	Password Verification Passed		password 1234	Enter Password	Verification Passed	Show the password input field

							border in green color
4	Verify Password	Password Verification Failed		password 1234	Enter Password	Verification Failed	Show the password input field border in red color
5	Check Login Button	Login Successful	Email & Password Verification Passed	Button Value = 1	Press Button	Login Successful	Redirect to the home page

II. Draw the bug life cycle.



05.

I. Explain the difference between white box and Black box testing?

	White Box Testing	Black Box Testing
1	White box testing is a software testing approach where the tester knows the internal structure of the software.	Black box testing is a software testing approach that tests the software without understanding the internal design of the code or program.
2	Applies to lower-level such as unit testing and integration testing.	Applies to high-level testing such as system testing and acceptance testing.
3	Structural or interior test.	External or functional test.
4	Can be started based on the Detail Design document.	Can be started based on Requirement Specifications documents.
5	White box testing requires implementation knowledge.	No knowledge of implementing black box testing is required.
6	White Box testing requires programming knowledge.	Black box testing does not require programming knowledge.
7	This type of testing is often done by software engineers.	Testers are responsible for this kind of testing.
8	White Box testing focuses on the program code of the system, i.e. code structure, conditionals, branches, loops, etc.	Black box testing focuses on system functionality.
9	White box testing includes structural testing, path testing, logic testing, loop testing, code coverage testing, and open box testing.	Black box testing includes behavior testing, functional testing, and close box testing.

II. What is the Test Automation?

Automated testing is a method of evaluating software products using specialized testing frameworks and tools to minimize human involvement while maximizing quality. Automated testing is performed with the help of automated software that controls the flow of test execution in accordance with written test protocols. They are compared against the expected results to guarantee the quality and dependability of the application. Automated testing allows one to perform important repetitive activities as well as those that are difficult to complete manually. As a result, this form of testing is essential for both continuous delivery and continuous testing.

III. What are the advantages of Test Automation?

- Automation testing includes all application functions.
- It is capable of running a large number of test cases simultaneously.
- It allows test cases to be reused and replayed.
- Automated testing eliminates the need for human interaction when executing test scripts.
- It improves accuracy as it eliminates the potential for human mistakes.
- 70% faster than manual testing, saving testers and the company significant time.
- It is reliable as it is done using effective testing equipment.
- The rapid feedback mechanisms of automation tests create an easy flow of information between developers and testers, allowing errors to be detected and fixed early.
- It improves and optimizes the software testing process.

IV. What are the standard associated with software testing?

- ANSI - American National Standards Institute

- ISO/IEC/IEEE 29119 series of standards for software testing.
- ISO/IEC/IEEE 29119-1: Concepts & Definitions (published Sept. 2013)
- ISO/IEC/IEEE 29119-2: Test Processes (published Sept. 2013)
- ISO/IEC/IEEE 29119-3: Test Documentation (published Sept. 2013)
- ISO/IEC/IEEE 29119-4: Test Techniques (expected publication late 2014)
- ISO/IEC/IEEE 29119-5: Keyword Driven Testing (expected publication 2015)

- Q9001-2008 - Quality Management Systems: Requirements;
- Q9000-2005 - Quality Management Systems: Fundamentals and Vocabulary;
- Q9004-2009 - Quality Management Systems: Guidelines for Performance Improvements;

- IEEE - Institute of Electrical and Electronics Engineers

- IEEE Standard for Software Test Documentation' (IEEE/ANSI Standard 829),
- IEEE Standard of Software Unit Testing (IEEE/ANSI Standard 1008),
- IEEE Standard for Software Quality Assurance Plans' (IEEE/ANSI Standard 730), and Others.

References

www.tutorialspoint.com. (n.d.). *STLC - Overview*. [online] Available at: https://www.tutorialspoint.com/stlc/stlc_overview.htm#:~:text=STLC%20Phases&text=Test%20Planning%20%E2%88%92%20Test%20Team%20plans [Accessed 19 Dec. 2022].

Guru99.com. (2019). *STLC - Software Testing Life Cycle Phases & Entry, Exit Criteria*. [online] Available at: https://www.guru99.com/software-testing-life-cycle.html.

GeeksforGeeks. (2019). *Software Testing Life Cycle (STLC) - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/software-testing-life-cycle-stlc/.

www.javatpoint.com. (n.d.). *Software Testing Life Cycle - javatpoint*. [online] Available at: https://www.javatpoint.com/software-testing-life-cycle [Accessed 19 Dec. 2022].

Hamilton, T. (2020). *Levels of Testing in Software Testing*. [online] www.guru99.com. Available at: https://www.guru99.com/levels-of-testing.html#:~:text=Unit%20Testing%20%3A%20checks%20if%20software [Accessed 19 Dec. 2022].

www.javatpoint.com. (n.d.). *Levels of Testing - javatpoint*. [online] Available at: https://www.javatpoint.com/levels-of-testing.

GeeksforGeeks. (2020). *Levels of Software Testing*. [online] Available at: https://www.geeksforgeeks.org/levels-of-software-testing/.

Hamilton, T. (2019). *How to Write Test Cases: Sample Template with Examples*. [online] Guru99.com. Available at: https://www.guru99.com/test-case.html.

Wikipedia. (2020). *Software bug*. [online] Available at: https://en.wikipedia.org/wiki/Software_bug.

www.javatpoint.com. (n.d.). *Bug in Software Testing - javatpoint*. [online] Available at: https://www.javatpoint.com/bug-in-software-testing.

Hamilton, T. (2019). *AUTOMATION TESTING Tutorial: What is, Process, Benefits & Tools*. [online] Guru99.com. Available at: https://www.guru99.com/automation-testing.html.

Wikipedia Contributors (2019). *Test automation*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Test_automation.

Guru99.com. (2020). *Software Quality Assurance(SQA): Plan, Audit & Review*. [online] Available at: https://www.guru99.com/software-quality-assurance-test-audit-review-makes-your-life-easy.html.

www.tutorialspoint.com. (n.d.). *Audit*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/audit.htm [Accessed 19 Dec. 2022].

Thomas Hamilton (2019). *What is User Acceptance Testing (UAT)? with Examples*. [online] Guru99.com. Available at: https://www.guru99.com/user-acceptance-testing.html.

Tutorialspoint.com. (2019). *User Acceptance Testing - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/use_acceptance_testing.htm.

GeeksforGeeks. (2019). *Difference between Alpha and Beta Testing*. [online] Available at: https://www.geeksforgeeks.org/difference-between-alpha-and-beta-testing/.

Guru99 (2019). *Alpha Testing Vs Beta Testing: What's the Difference?* [online] Guru99.com. Available at: https://www.guru99.com/alpha-beta-testing-demystified.html.

www.javatpoint.com. (n.d.). *Difference Between Alpha Testing and Beta Testing - javatpoint*. [online] Available at: https://www.javatpoint.com/alpha-testing-vs-beta-testing.

Indeed Career Guide. (n.d.). *How to Perform a Risk Analysis*. [online] Available at: https://www.indeed.com/career-advice/career-development/how-to-perform-a-risk-analysis.

Jennifer Bridges (2017). *How to Analyze Risks on Your Project - ProjectManager.com*. [online] ProjectManager.com. Available at: https://www.projectmanager.com/training/how-to-analyze-risks-project.

Hamilton, T. (2019). *How to Create a Test Plan (with Example)*. [online] Guru99.com. Available at: https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html.

www.tutorialspoint.com. (n.d.). *Test Plan - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/test_plan.htm.

www.javatpoint.com. (n.d.). *Test Plan - javatpoint*. [online] Available at: https://www.javatpoint.com/test-plan.

GeeksforGeeks. (2022). *Software Testing - Test Plan*. [online] Available at: https://www.geeksforgeeks.org/software-testing-test-plan/.

GeeksforGeeks. (2019). *Categories of Software Defect*. [online] Available at: https://www.geeksforgeeks.org/categories-of-software-defect/ [Accessed 19 Dec. 2022].

www.tutorialspoint.com. (n.d.). *Defect*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/defect.htm.

GeeksforGeeks. (2022). *Software Testing - Bug vs Defect vs Error vs Fault vs Failure*. [online] Available at: https://www.geeksforgeeks.org/software-testing-bug-vs-defect-vs-error-vs-fault-vs-failure/#:~:text=or%20execution%20fails%2C- [Accessed 19 Dec. 2022].

www.javatpoint.com. (n.d.). *Difference between Bug, Defect, Error, Fault and Failure - javatpoint*. [online] Available at: https://www.javatpoint.com/bug-vs-defect-vs-error-vs-fault-vs-failure.

Martin, M. (2019). *SDLC (Software Development Life Cycle) Tutorial: What is, Phases, Model*. [online] Guru99.com. Available at: https://www.guru99.com/software-development-life-cycle-tutorial.html.

GeeksforGeeks. (2019). *Types of Defects in Software Development*. [online] Available at: https://www.geeksforgeeks.org/types-of-defects-in-software-development/.

tutorialspoint (2019). *SDLC V-Model*. [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm.

www.ibm.com. (2017). *Test log overview*. [online] Available at: https://www.ibm.com/docs/en/rpt/9.1.0?topic=SSMMM5_9.1.0/com.ibm.rational.test.lt.common.doc/topics/ttestlogoverview.htm [Accessed 19 Dec. 2022].

www.professionalqa.com. (n.d.). *What is Test Log? |Professionalqa.com*. [online] Available at: https://www.professionalqa.com/test-log.

Hamilton, T. (2019). *What is Requirements Traceability Matrix (RTM)? Example Template*. [online] Guru99.com. Available at: https://www.guru99.com/traceability-matrix.html.

Jama Software. (n.d.). *What is Requirements Traceability*. [online] Available at: https://www.jamasoftware.com/requirements-management-guide/requirements-traceability/what-is-traceability-and-why-does-it-matter-for-product-teams#:~:text=Requirements%20traceability%20enables%20product%20teams [Accessed 19 Dec. 2022].

www.tutorialspoint.com. (n.d.). *Test Case - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/test_case.htm.

www.javatpoint.com. (n.d.). *Test Case - javatpoint*. [online] Available at: https://www.javatpoint.com/test-case.

Wikipedia Contributors (2019). *Test case*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Test_case.

www.tutorialspoint.com. (n.d.). *Defect Life Cycle*. [online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/defect_life_cycle.htm.

GeeksforGeeks. (2020). *Bug Life Cycle in Software Development*. [online] Available at: https://www.geeksforgeeks.org/bug-life-cycle-in-software-development/ [Accessed 19 Dec. 2022].

www.guru99.com. (n.d.). *Defect/Bug Life Cycle in Software Testing*. [online] Available at: https://www.guru99.com/defect-life-cycle.html.

Jain, M. (2018). *Differences between black box testing vs white box testing - geeksforgeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/.

Guru99.com. (2019). *Black Box Testing Vs. White Box Testing: Key Differences*. [online] Available at: https://www.guru99.com/back-box-vs-white-box-testing.html.

Anon, (n.d.). *Black Box vs. White Box Testing: Understanding 3 Key Differences |*. [online] Available at: https://www.spiceworks.com/tech/devops/articles/black-box-vs-white-box-testing/#:~:text=Black%20box%20testing%20is%20a.

Hamilton, T. (2019). *AUTOMATION TESTING Tutorial: What is, Process, Benefits & Tools*. [online] Guru99.com. Available at: https://www.guru99.com/automation-testing.html.

Wikipedia Contributors (2019). *Test automation*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Test_automation.

www.javatpoint.com. (n.d.). *Advantages of Automation Testing - Javatpoint*. [online] Available at: https://www.javatpoint.com/advantages-of-automation-testing [Accessed 19 Dec. 2022].

admin (2019). *Test Automation:Top 10 Benefits of Automation Testing By Testrig Technologies*. [online] Top QA & Software Testing Company. Available at: https://www.testrigtechnologies.com/top-10-benefits-of-automation-testing/.

Automation, A. (2021). *10 Benefits of Test Automation*. [online] Avo Automation. Available at: https://avoautomation.ai/10-benefits-of-test-automation/.

Softwaretestingstandard.org. (2013). *ISO/IEC/IEEE 29119 Software Testing*. [online] Available at: https://softwaretestingstandard.org/.

www.tutorialspoint.com. (n.d.). *Software Testing - ISO Standards - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/software_testing/software_testing_iso_standards.htm.

www.w3schools.in. (n.d.). *Software Testing Standards*. [online] Available at: https://www.w3schools.in/software-testing/standards.