**TASK 14**

**What is the difference between automated and manual testing in software development**

| **Parameter** | **Automation Testing** | **Manual Testing** |
| --- | --- | --- |
| Definition | Automation Testing uses automation tools to execute test cases. | In manual testing, test cases are executed by a human tester and software. |
| Processing time | Automated testing is significantly faster than a manual approach. | Manual testing is time-consuming and takes up human resources. |
| Exploratory Testing | Automation does not allow random testing | Exploratory testing is possible in Manual Testing |
| Initial investment | The initial investment in the automated testing is higher. Though the ROI is better in the long run. | The initial investment in the Manual testing is comparatively lower. ROI is lower compared to Automation testing in the long run. |
| Reliability | Automated testing is a reliable method, as it is performed by tools and scripts. There is no testing Fatigue. | Manual testing is not as accurate because of the possibility of the human errors. |
| UI Change | For even a trivial change in the UI of the AUT, Automated Test Scripts need to be modified to work as expected | Small changes like change in id, class, etc. of a button wouldn’t thwart execution of a manual tester. |
| Investment | Investment is required for testing tools as well as automation engineers | Investment is needed for human resources. |
| Cost-effective | Not cost effective for low volume regression | Not cost effective for high volume regression. |
| Test Report Visibility | With automation testing, all stakeholders can login into the automation system and check test execution results | Manual Tests are usually recorded in an Excel or Word, and test results are not readily/ readily available. |
| Human observation | Automated testing does not involve human consideration. So it can never give assurance of user-friendliness and positive customer experience. | The manual testing method allows human observation, which may be useful to offer user-friendly system. |
| Performance Testing | Performance Tests like Load Testing, Stress Testing, Spike Testing, etc. have to be tested by an automation tool compulsorily. | Performance Testing is not feasible manually |
| Parallel Execution | This testing can be executed on different operating platforms in parallel and reduce test execution time. | Manual tests can be executed in parallel but would need to increase your human resource which is expensive |
| Batch testing | You can Batch multiple Test Scripts for nightly execution. | Manual tests cannot be batched. |
| Programming knowledge | Programming knowledge is a must in automation testing. | No need for programming in Manual Testing. |
| Set up | Automation test requires less complex test execution set up. | Manual testing needs have a more straightforward test execution setup |
| Engagement | Done by tools. Its accurate and never gets bored! | Repetitive Manual Test Execution can get boring and error-prone. |
| Ideal approach | Automation testing is useful when frequently executing the same set of test cases | Manual testing proves useful when the test case only needs to run once or twice. |
| Build Verification Testing | Automation testing is useful for Build Verification Testing (BVT). | Executing the Build Verification Testing (BVT) is very difficult and time-consuming in manual testing. |
| Deadlines | Automated Tests have zero risks of missing out a pre-decided test. | Manual Testing has a higher risk of missing out the pre-decided test deadline. |
| Framework | Automation testing uses frameworks like Data Drive, Keyword, Hybrid to accelerate the automation process. | Manual Testing does not use frameworks but may use guidelines, checklists, stringent processes to draft certain test cases. |
| Documentation | Automated Tests acts as a document provides training value especially for automated unit test cases. A new developer can look into a unit test cases and understand the code base quickly. | Manual Test cases provide no training value |
| Test Design | Automated Unit Tests enforce/drive Test Driven Development Design. | Manual Unit Tests do not drive design into the coding process |
| Devops | Automated Tests help in Build Verification Testing and are an integral part of DevOps Cycle | Manual Testing defeats the automated build principle of DevOps |
| When to Use? | Automated Testing is suited for Regression Testing, Performance Testing, Load Testing or highly repeatable functional test cases. | Manual Testing is suitable for Exploratory, Usability and Adhoc Testing. It should also be used where the AUT changes frequently. |

**Explore some of the most common automation testing tools available in the market**

Free automation tools like Selenium or Protractor are quite popular in the community of testers. There are a lot of benefits to using open-source software, such as better pricing for licensing and the ability to create tailored solutions or extended collaboration.

1. Selenium

2. Lambda Test

3. Cucumber

4. Protractor

5. Cypress.io

6. Apache Meter

7. Appium

**What is cross browser testing**

Cross-browser testing is a quality assurance (QA) process that verifies that web applications work as intended across different browsers, operating systems, and devices. The goal is to identify errors in frontend functionality before users encounter them.

Cross-browser testing can help create a positive user experience on a website, which can result in higher conversion rates and more revenue.

* Cross-browser testing involves:
* Identifying which features to test
* Writing steps to specify the scenarios
* Identifying the browsers and platforms to test on
* Picking how to execute the test scenarios

Cross-browser testing can be done manually or as part of a test automation strategy

**What is blog on TDD and BDD**

TDD (Test Driven Development) and BDD (Behavior Driven Development) are fairly similar development approaches that both emphasize testing and collaboration, yet have major differences in focus and methodology

**TDD** is a software development process that involves writing tests before writing code. It's an iterative process that includes:

* Writing a test
* Running the test
* Writing the code
* Running all tests
* Refactoring

TDD helps develop simple, clean, and extensible code. It also results in more thorough testing and fewer bugs. TDD is a good opportunity to detect bugs and crashes quickly and resolve the issue immediately. This reduces the costs of tedious debugging if the errors were discovered later.

TDD was developed by Kent Beck in the late 1990's as part of Extreme Programming

**BDD** is a software development process based on the Agile methodology. It uses examples in plain text to define a feature's behavior, which are then used as acceptance criteria. BDD can improve communication between technical and business teams, and help ensure that software development meets business goals. Some benefits of BDD include:

* Traceability: All development work can be traced back to business objectives
* User needs: Software development meets user needs
* Prioritization: Business-critical features are delivered first