**OVERVIEW OF TESTING ESSENTIALS**

**by- Rupal Manwatkar**

Testing essentials refer to the fundamental skills, knowledge, and practices necessary for effective software testing. These include understanding different testing types and levels, test planning, test case design, test execution, defect logging, and test closure activities. Testing essentials provide the foundation for ensuring software quality and reliability throughout the development lifecycle.

**What Have We Learned in Testing Essentials?**

* **Testing Fundamentals:** Understanding the basic concepts, types, and levels of testing.
* **Test Planning:** Importance of planning, creating test plans, and defining scope and objectives.
* **Test Design:** Designing test cases, preparing test data, and understanding various test design techniques.
* **Test Execution:** Executing test cases, logging defects, and tracking test progress.
* **Test Closure:** Concluding the testing phase, analyzing results, and documenting lessons learned.

**Implementing Testing in a Real-Time Product: Example - Banking Application**

**Scenario:** Testing a banking application that includes functionalities such as account management, fund transfers, and transaction history.

**Steps to Implement:**

1. **Requirement Analysis:** Identify testable features like account creation, login, fund transfer, balance inquiry, and transaction history.
2. **Test Planning:** Define the testing scope, estimate effort, select tools (e.g., Selenium for automation), and plan resources.
3. **Test Case Design:** Create test cases for functionalities like:
   * Verifying user login with valid and invalid credentials.
   * Testing fund transfer between accounts.
   * Checking the accuracy of transaction history.
4. **Test Environment Setup:** Set up a test environment that mimics the production environment, including database, application server, and network configurations.
5. **Test Execution:** Run the designed test cases, log any defects, and retest after fixes.
6. **Test Cycle Closure:** Analyze test results, document lessons learned, and ensure all issues are resolved before release.

**Why Testing Essentials Are Needed**

1. **Quality Assurance:** Ensures that the product meets the required standards and functions correctly.
2. **Risk Mitigation:** Identifies and addresses defects early, reducing the risk of failures in the production environment.
3. **Cost Efficiency:** Detecting defects early in the development cycle is more cost-effective than fixing them after release.
4. **Customer Satisfaction:** Delivering a well-tested product enhances user experience and trust.

**The Journey Ahead in Testing**

1. **Deepening Knowledge:** Moving from basic testing concepts to advanced techniques.
2. **Specialized Testing:** Exploring different types of testing like performance, security, and usability testing.
3. **Automation:** Learning automation tools and frameworks to increase testing efficiency and coverage.
4. **Continuous Improvement:** Implementing testing best practices and continuously improving testing processes.

**Why We Need to Know Testing Essentials Before Automation Overview**

1. **Foundational Knowledge:** Understanding manual testing fundamentals is crucial before transitioning to automation.
2. **Effective Automation:** Knowing what and how to test manually helps in identifying test cases suitable for automation.
3. **Context and Scope:** Grasping the basics ensures you understand the context, scope, and limitations of automated tests.
4. **Quality Automation:** Ensures the development of robust and maintainable automated test scripts.

**Overview of Testing Essentials Before Automation**

1. **Testing Basics:** Introduction to software testing, its importance, and types of testing (unit, integration, system, acceptance).
2. **Test Planning:** Creating effective test plans, defining scope, objectives, resources, and scheduling.
3. **Test Design:** Designing comprehensive test cases, preparing test data, and understanding various test design techniques (e.g., boundary value analysis, equivalence partitioning).
4. **Test Execution:** Executing test cases, logging defects, retesting, and tracking progress.
5. **Test Closure:** Concluding the testing phase, analyzing results, and documenting lessons learned.