

# AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Assignment Title: Assignment on Test Strategy

Date of Submission: 3  
July, 2021

Course Title:  
Software Quality  
And Testing

Course Code: CSC4133

Section: C

Semester: Summer

2020-2021

Course Teacher: Abhijit Bhowmik

## Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

\* Student(s) must complete all details except the faculty use part.

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

Submitted By: Hoque, Fardin

No	Name	ID	Program	Signature
1	Hoque, Fardin	18-37245-1	CSE	
2	Bari, SK Tasnim	18-37201-1	CSE	
3	Patwary, Tanjib	18-37230-1	CSE	
4	Rizvee, Mayin Uddin Ahmed	18-36628-1	CSE	
5				
6				
7				
8				
9				
10				

## Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

### **Question 1: Describe from your point of view how you can state the term quality?**

**Answer:** The quality of software can be defined as how well the software is properly designed as per user requirement. Quality software is bug free, developed within the budget and time and satisfy user needs. Also, the main aim of software quality is to build the functionalities and features of a software correctly.

As a project manager for this given project first need to know the user requirements properly to build a quality-full software. Without proper information of this project, it is not possible to develop the product correctly. Also, we have to check the way that we used to implement the software is good enough or not, the product we are building is correctly designed or not, after completing the implementation check the software has any defects. If any defect detected then solve the defects. Also, we have to check the security issues. Is the software able to block the malwares or not? All these are dependent for building a quality-full software. Every well-structured and bug-full software follow the ISO 9126 certification rules which also related to the term quality. As a project manager, I must have to follow the ISO certifications for build the software. For this there are some quality attributes. They are:

**Functionality:** It defines that the software performs and functions correctly according to its requirements. It verifies the software is able to give the correct output or not, how accurately it performing the operations, how much it is able to protect the malwares and how much the software is secure etc. Also, it verifies that the software that we made is able to couple with another system or not.

**Reliability:** The probability of the software executing without failure for a specific period of time is known as reliability. Reliability also measures the errors, bugs of a software product. Based on these things it measures the quality.

**Usability:** It defines that how easily the user can use the software and how easy to understand the features of the software. Without user satisfaction, we can not say that the software is quality-full. Mainly the software is properly designed so that both the consumers and customers can easily use this system.

**Portability:** Portability is the effort to transfer a program from one environment to another. So, we have to check the software is performing well in all environment or not.

**Maintainability:** It verifies how easily developer or tester can fix the bugs of a software. Well-designed software is easy to maintain and easy to fix if any error occurs. Developer can easily change the functions according to the user needs if the software is well maintained.

**Efficiency:** The amount of time requires to give an accurate result by the software determines the efficiency of a software. How much memory the software is using for operating any task etc.

**Software Quality Assurance** is also an important part that related to the term quality. It creates an effective plan for developing a quality-full software. The processes, functions and the standards that helps to develop the plan. All these are involved in SQA. Before developing any software,

designing its structure is very much important. A set of activities designed so that it will easy to ensure the system's quality. High quality software is fully dependent on this effective plan. As a project manager I also have to follow SQA for developing better quality software.

Umbrella activity is very useful in software quality assurance. The methods which use in umbrella activity is very much effective for developing any quality-full software.

**Umbrella activities:**

- Software Project Tracking and Control
- Formal Technical Reviews
- Software Quality Assurance
- Software Configuring Management
- Document Preparation and Production
- Reusability Management
- Measurement
- Risk Management



Fig: Umbrella Activities

**Software Quality Control** is another important part that also related to the term quality. The main aim of SQA is to develop an effective plan where as in SQC, its main aim is to execute that plan effectively. It follows the plans, standards for developing the system. Its aim is to identify and solve the defects. Software quality control is mainly performed by testing team. It ensures the outputs that what we have done and what we have expected. For developing our project, we have to follow the software quality control attributes for better quality.

**SQC follows some activities:**

- Requirement Review
- Design Review
- Code Review

- Deployment Plan Review
- Test Plan Review
- Test Cases Review

**Software Testing:** Software testing is the most important part which finalizes the quality of the product. Firstly, software testing is the process of executing a system under some fixed conditions and finding the defects and solving them. There are different types of testing methods that are used to verify the quality of the software. Unit testing is one of them. Under unit testing, there are two types of testing criteria: i) Static Unit Testing. ii) Dynamic Unit Testing. Static Unit testing is used when compiling the codes and Dynamic Unit Testing is used in run time to ensure the quality of the software. There are some code review processes and also debugging techniques that are responsible for verifying the quality of a software.

Also, for developing a quality-full software, verification and validation are two important terms. All team members of a company are responsible for verifying the software quality and only testing team members are responsible for validating the quality. Without verification and validation, we cannot say that the software meets all the requirements correctly. So as a project manager, if my team members follow all the above plans and activities properly when implementing the ERP software project for this poultry and frozen food company, then we will be able to develop a good quality software.

## **2. Measuring what you can say the software you developed assured quality or not and how?**

**Answer:**

First of all, we need to know what really quality assurance is. SQA (Software Quality Assurance) is a technique for managing quality that focuses on assuring that quality requests will be met. In this case, quality will be assured if the automation of our client's company meets with desired functionalities.

Now, we need to consider the benchmarks for measuring the software quality. And they are :

### **Efficiency**

The amount of computational resources and code that a program needs to complete a task.

### **Security**

The extent to which illegal access to software or data can be controlled.

### **Usability**

The time it takes to learn how to use the program, how to operate it, how to gather input, and how to evaluate its output.

## **Reliability**

Reliability refers to the likelihood of software running without errors for a specified period of time.

**How does the QA team measure Software Quality is the main question that arises.**

QA testers will examine all aspects of software quality, including the product code's validity and standard through manual and automated testing. Base metrics and Calculated Metrics are the two types of manual test metrics. Basis metrics are made up of raw, unprocessed data acquired, whereas calculated metrics are produced from the data obtained in the base metrics.

As the job is to automate the company's daily works we need to consider important metrics for automation testing and they are:

- Total time spent on the test
- Coverage of unit tests
- Coverage of the path (how many linearly independent paths of the program the test covers)
- Coverage of requirements
- Percentage of tests that pass/fail
- The number of flaws
- Coverage of automated tests as a percentage (against the total test coverage which includes manual testing)
- Execution of the test (total tests executed during the build)
- Results that are useful vs. those that aren't
- Manufacturing flaws
- The percentage of built structures that are broken, and so forth.

Some other QA alternatives to testing the system are:

- Defect Prevention (Strategy to identify root causes of defects)
- Formal Verification (A suitable mathematical model or can be formal mathematical proofs)
- Fault Tolerance (Software's ability to detect and recover from a fault that is occurring or has already occurred)

So as a project manager, I can ensure that my software company's developers as well as QA team are going to assure our client's Poultry and Frozen Food Company's ERP based automation software's quality by measuring the key points described here.

**3. Is it possible to assure top quality for every quality attributes? Define your answer.**

**Ans:**

A product must go through some quality criteria to ensure high quality.

As an example:

**Functionality:** The conformity of a software-driven system to actual needs and specifications is determined by this attribute. Most Software Testing specialists consider this feature to be critical and a top requirement of a modern application, and therefore recommend performing tests that evaluate the desired functionality of a system early in the Software Testing process.

**Reliability:** Product reliability is determined by how well a project performs in various working environments and settings. It is check to see if the product is durable enough to sustain any situation. Basically, the probability of the software executing without failure for a specific period of time is known as reliability.

**Usability:** The effort required to learn, operate, prepare input, and interpret output of a program is usability. This also can be measured in terms of ease of use. The application should be simple to use. It should be simple to understand. The navigation should be straightforward. The system must be simple to use in terms of input preparation, operation, and output interpretation. It ensures that our other frequently used systems have consistent user interface standards or conventions. Usability refers to how easy it is for new, infrequent, or any type of user to learn to use the system.

**Portability:** Technical challenges, costing issues, behavioral issues, these all are related to portability.

**Maintainability:** The effort required to locate and fix a defect in an operational program. Basically, this is the process of determining if the system can update and modify the application if necessary. Because the system is subjected to modifications throughout the software life cycle, this is a critical component. The software must be maintained once the system has been deployed in a production environment. Maintenance should be both inexpensive and simple. The system should simple to maintain, and adding changes to the software or resolving flaws is simple.

**Efficiency:** It's a major system quality attribute. Measured in terms of the time it takes for the system to execute any given task. So, the amount of computing resources and code required by program to perform a function. For example, the system should make optimal use of CPU capability, disk space, and memory. If the system consumes all available resources, the user's performance will suffer, and the system would be deemed inefficient. It is impossible to employ a system in real-time applications if it is inefficient.

**Correctness:** In terms of functionality, internal computations, and navigation, the app should be flawless. This implies that the application must meet all functional criteria.

**Performance:** This characteristic refers to a software-driven system capacity to follow to the time constraints. It suggests that Software Testing engineers must evaluate whether the system responds to various events within established time constraints from a fixed testing time. Clock events, process interruptions, communications, and requests from various users, among other things, are examples of these events.

**Integrity/Security:** System integrity or security should be sufficient to prevent unauthorized access to system functions, information loss, and virus infection of software, as well as to protect the privacy of data entered into the system. Actually, Integrity comes with security.

**Testability:** Software testability refers to how well a software-driven system enables Software Testing specialists to perform tests according to the set of criteria. This criterion also considers how easy it is for Software Quality Assurance engineers to create test criteria for a system and its components. Engineers can evaluate a system testability using variety of techniques, including encapsulation, interfaces, patterns, reduced coupling, and more.

**Flexibility:** This attribute ensures that software should be flexible enough to modify in terms of needs. It can be adapted to work with other goods with which it needs to interface. It should be simple to integrate with other common third-party components.

**Reusability:** The extent to which a software system component can be reused in other applications. Software reuse is a smart technique to produce software that is both cost-effective and time-saving. Classes in distinct code libraries should be generic enough to be used in a variety of application modules. Organizing the app into separate modules so that modules can be utilized throughout the program.

**Interoperability:** To complete particular tasks, software-driven systems may be necessary to communicate. The capacity of two systems to communicate information via specific interfaces is referred to as interoperability. As a result, Software Quality Assurance engineers must consider both syntactic and semantic interoperability when evaluating the interoperability characteristic.

We can not say that the software that we are developing is 100% bug-full. As, top quality software depends on this for it's quality so that if our team members follow all the above attributes properly then we can ensure the top quality software as much as possible.

**Question 4: According to your plan the software you are developing will be tested by whom?**

**Answer:** Software testing is very important. After developing a software, it needs to be tested to ensure it's quality. This software should be tested by full-time software testers as well as professionals. They have to perform various duties. Such as-software analysis, mitigate risk of software and system, prevent defected issues. There are several techniques that are used in each level of testing a software. As our software is an ERP software so that we have to perform some testing methods for finding the bugs. Both manual and automated testing will be performed for testing this software. According to our plan the software we are developing will be performed by them:

**Developers:** They will perform the unit testing.

**Unit Testing:** The purpose of unit testing is to validate that each unit of the software code performs as expected. Unit testing is done by developers during the development of a software. Unit testing

helps to fix the errors early in the development process. It helps the developers to understand the testing code easily and helps them to make changes quickly. Under unit testing static and dynamic unit testing will be performed.

**Static Unit Testing:** Static unit testing mainly checks the design and the documents to find the errors without code execution. Static testing involves both manual and automated reviews of the documents. In static unit testing code is reviewed by Walkthrough and Code Inspecting method.

**Dynamic Unit testing:** Dynamic testing is used to check the functional behavior and find the errors from a software. It mainly performs in the compiled and run time. The main purpose of this testing is to check the software product works correctly with the business requirements.

**QA:** The QA is responsible for designing and implementing the tests. They will perform regression testing when any bug will detect and resolve it. They will monitor debugging process result. They will create detailed and well-structured test plan and case for our project.

**Test Analyst:** They will responsible for identify and defines the require tests, monitor detailed testing progress and evaluate the overall quality of the software product. Also evaluate the outcome of each test cycle.

**Testers:** Software tester are responsible for quality software development. Testers execute the test operations, evaluate the results and document problems found. They monitor the testing and use different testing tools according to their requirements. Testing tools are:

- Documenters
- Code auditor
- Test data generator
- Test harness
- Performance monitors etc.

Software tester performs both manual and automated testing for finding the bugs and fixes those bugs.

**Manual Testing:** Manual testing is mainly performed by tester to identify the errors without using any automated tools. In this testing method Tester manually executes the test cases. Manual testing is labor dependent.

**Automated Testing:** Automation is a software testing method that use special types of tools for identify the bugs from the code. Test automation is used in automate repetitive tasks and other tasks which are difficult to perform manually.

**Developer/Programmer** will also perform the debugging technique in the testing phase. Debugging will perform by them which also important for finding the bugs from the software. The test engineers perform test cases, finding the bugs and they can report back to the development team for fixing those bugs. Then the programmer performs debugging technique.

**Debugging:** Debugging is the part of software testing process. It is the process to correct the bugs. In debugging process developer needs to identify the reason behind the selected bug and solve it. The developer changes the code and then rechecks whether the defect has been deleted whenever



the bug is found. Once the debugging is successfully then the system is again sent back to the test engineers for testing.

So, above engineers will be responsible for testing of our project that we are developing. After all the testing phase we will do the acceptance testing which will be performed by the End-User. Acceptance testing is very much important for our project. It is one of the final stages of the software's testing cycle and often occurs before a client accepts the new software. In this stage users will give their feedback that helps the developer to find and uncover issues and fix those problems.