**Chapter 7: System Quality & Testing**

## System Testing

According to the common process framework (CPF), the software testing is the final activity that has to initiate after developing. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. A quality management software system that is automated and connects all departments is essential for a regulated or ISQ-compliant company. A QMS or a TQM (total quality management) system can connect each phase in a products development lifecycle with every department in a company. This gives everyone an opportunity to provide feedback. Automated, routing, with escalation, ensure the rapid responses to inputs needed from the department. By building quality into products as opposed to forcing QA to bear the burden of the responsibility, everyone wins, engineering, regulatory, QA, manufacturing, sales and marketing

The quality of software is assessed by a number of variables. These variables can be divided into external and internal quality criteria. External quality is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code-dependent, and that are not visible to the end-user. External quality is critical to the user, while internal quality is meaningful to the developer only.

Some quality criteria are objective, and can be measured accordingly. Some quality criteria are subjective, and are therefore captured with more arbitrary measurement.

There are mainly two types of quality

* Internal Quality
* External Quality
* Internal quality:
* Test coverage
* Testability
* Portability
* Thread-safeness
* Conciseness
* Maintainability
* Documentation
* Legibility
* Scalability
* External quality
* Features
* Speed
* Space
* Network usage
* Stability
* Robustness
* Ease-of-use
* Determinism
* Back-compatibility
* Security
* Power consumption

**7.1.1 Software Quality Management Process**

* The aim of Software Quality Management (SQM) is to manage the quality of software and development and of its development process
* A quality product is one which meets its requirements and satisfies the user
* A quality culture is an organization environment where quality is viewed as everyone’s responsibility

Some of the specific SQM processes defined in standard:

* **Quality assurance process**

Quality Assurance makes sure the project will be completed based on the previously agreed specifications, standards and functionality required without defects and possible problems. Itsmonitors and tries to improve the development process from the beginning of the project to ensure this.it is oriented to ‘’prevention’’

* **Verification & validation process**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it full fill its intended purpose. It is normally the responsibility of software testers as part of the software development lifecycle.

In the verification, a client will either view the software, or see it implemented in a test situation. At this point it is imperative that the client who is needed of the software is able to ascertain that this software is hitting all the parameters initially requested or desired .only when this assurance is made should the next part of the verification and validation process be started. While this is not the last chance to “tweak” the software into doing the tasks required it is part of the last steps before a project is completed, and in being too quick to approve the software as this could cause problems later, and could also result in more money required for the software’s later changes.

The next step of verifications &validation of software is simple. ClientCompany will approve the software and validate it as being what is required. This stage usually means a systematic checking off various requirements. While this might sound tedious, it is necessary part of the procedure to insure that again, the result is exactly to the specifications of all concerned. The entire verification and validation process is part of a normal sequence of quality control for software.

* **Review& Audit Process**

A software audit review, or software audit, is a type of software review in which one or more auditors who are not members of the software development organization conduct “An independent examination of a software product, softwareprocess, or set of software processes to assess compliance with specifications,standards,contractual agreements, or other criteria”

Five types of reviews or audits presented in the standard:

1. **Management reviews**

The purpose of a management review is to monitor progress, determine the status of plans and schedules, confirm requirements and their system allocation, or evaluate the effectiveness of management approaches used to achieve fitness for purpose. This support decisions about changes and corrective actions that are required during a software project.

1. **Technical reviews**

The purpose of technical review is to evaluate a software product to determine its suitability for its intended use. The objective is to identify discrepancies from approved specifications and standards. The result should provide management with evidence confirming (ornot) that the product meets the specifications and adheres to standards and that changes are controlled”. A technical review requires that mandatory inputs be in place in order to proceed:

* Statements of objectives
* A specific software product
* The specific project management plan
* The issues list associated with this product

1. **Inspections**

The purpose of an inspection is to detect and identify software product anomalies. Two important differentiators of inspections as opposed to reviews are as follows:

* An individual holding a management position over any member of the inspection team shall not participate in the inspection
* An inspection is too led by an impartial facilitator who is trained in inspection techniques.

The inspection exit must correspond to one of the following three criteria:

* Accept with no or at minor reworking
* Accept with rework verification
* Re inspect

Inspection meetings typically last a few hours, whereas technical reviews and audits are usually broader in scope and take longer.

1. **Walk-through**

The purpose of a walk-through is to evaluate a software product. A walk-through may conduct for educating an audience regarding a software product. The major objectives are to:

* Find anomalies
* Improve the software product
* Consider alternative implementations
* Evaluate conformance to standards and specifications

The purpose of a software audit is to provide an independent evaluation of the conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures. The audit is a formally organized activity, with participants having specific roles, such as lead auditor, another auditor, a recorder, or an initiator, and includes a representative of the audited organization. The audit will identify instances of nonconformance and produce a report requiring the team to take corrective action.

**7.2 System Testing**

* **Black box testing:** You don’t need to know the internal design in detail or have knowledge about the code for this test. It’s mainly based on functionality and specification, requirements.
* **White box testing**: This testing is based on detailed knowledge of the internal design and code. Tests are performed for specific code statements and coding styles.
* **Unit testing:**The most micro scale of testing to test specific functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code, may require developing test driver modules or test harnesses.
* **Incremental integration testing:** Continuous testing of an application as new functionality is added. Requires that various aspects of applications functionality be independent enough to work separately before all parts of the programmer are completed or that test drivers be developed as needed. Done by programmers or by testers.
* **Integration testing:** Testing of combined parts of an application to determine if they function together correctly. It can be any type of application which has several independent sub applications, modules.
* **Functional testing:** Black box type testing to test the functional requirement of an application. Typically done by software testers but software programmers should also check if their code works before releasing it.
* **System testing:** Black box type testing that is based on overall requirements specifications. Covers all combined parts of a system.
* **End to End testing:** It’s similar to system testing. Involves testing of a complete application environment similar to real world use. May require interacting with a database, using network communications, or interacting with other hardware, applications, or systems.
* **Sanity testing or smoke testing:** An initial testing to determine if a new sw version is performing well enough to start for a major software testing. For example, if the new software is crashing frequently or corrupting database then it is not a good idea to start testing before all these problems are solved first.
* **Regression testing:** Re-testing after software is updated to fix some problem. The challenge might to be determining what need to be tested, and all the interactions of the functions, especially near the end of the software cycle. Automated testing can be useful for this type of testing.
* **Accepting testing:** This the final testing done based on the agreements with the customer.
* **Load / stress / performance testing:** Testing an application under heavy loads. Such as simulating a heavy traffic condition in a voice or data network, or a web site to determine at what point the system start causing problems or fails.
* **Usability testing:** Testing to determine how user friendly the application is. It depends on the user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers.
* **Install / Uninstall testing:** Testing of full, partial, or upgrade install / uninstall processes.
* **Recovery / failover testing:** Testing to determine how well a system recovers from crashes, failures, or other major problems.
* **Security testing:** Testing to determine how well the system protects itself against unauthorized internal or external access and intentional damage. May require sophisticated testing techniques.
* **Compatibility testing:** Testing how well software performs in different environments. Particular hardware, software, operating system, network environment etc. Like testing a web site in different browsers and browsers and browsers versions.
* **Exploratory testing:** Often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it.
* **Ad-hoc testing:** Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.
* **Context driven testing:** Testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life critical medical equipment software would be completely different than for a low cost computer game.
* **Comparison testing:** Comparing software weakness and strengths to competing products.
* **Alpha testing:** Testing of an application when development when development is nearing completion. Minor design changes may still be made as a result of such testing. Typically done by end users or others, not by programmers or testers.
* **Beta testing:** Testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end users or others, not by programmers or testers.
* **Mutation testing:** A method for determining if a set data or test case is useful, by deliberately introducing various code changes and retesting with the originals test data/ cases to determine if the defects are detected. Proper implementation requires large computation resources.

**7.3 Testing Design**

|  |  |
| --- | --- |
| **Testing Scenario No:1** | |
| Scenario | Login testing scenario of our system |
| Input’s | Username and password of admin for login |
| Desired Output’s | When enter username, password then get access level define. |
| Actual Output’s | For login our system works properly. |
| Verdict | Getting result from desired outputs and actual outputs decided this system is successful for login. |

|  |  |
| --- | --- |
| **Testing Scenario No:2** | |
| Scenario | Student info insert testing scenario of our system |
| Input’s | Admin insert student details |
| Desired Output’s | Student info will show for start or stop job |
| Actual Output’s | We check this process and get actual outputs |
| Verdict | Our system is worked correctly and successfully. |

|  |  |
| --- | --- |
| **Testing Scenario No:3** | |
| Scenario | Result insert testing scenario of our system |
| Input’s | Admin insert their Marks |
| Desired Output’s | Start time will add into db |
| Actual Output’s | We check this process and get actual outputs |
| Verdict | Our system is worked correctly and successfully. |

|  |  |
| --- | --- |
| **Testing Scenario No:4** | |
| Scenario | Update student info testing scenario of our system |
| Input’s | Admin insert new information about existing student |
| Desired Output’s | Update report will show |
| Actual Output’s | Our desired output access to actual and practical output. So, this is successful. |
| Verdict | Our system is worked correctly and successfully. |

|  |  |
| --- | --- |
| **Testing Scenario No:5** | |
| Scenario | Student certificate application testing scenario of our system |
| Input’s | Student can apply for certificate |
| Desired Output’s | After click on certificate option |
| Actual Output’s | Our desired output access to actual and practical output. So, this is successful. |
| Verdict | Our system is worked correctly and successfully. |

|  |  |
| --- | --- |
| **Testing Scenario No:6** | |
| Scenario | Final report generation testing scenario of our system |
| Input’s | Admin will request to generate a report |
| Desired Output’s | After select category of report system will generate a report. |
| Actual Output’s | Our desired output access to actual and practical output. So, this is successful. |
| Verdict | Our system is worked correctly and successfully. |